IMPROVED PERFORMANCE OF HYBRID ERROR CONTROL TECHNIQUES FOR REAL-TIME DIGITAL COMMUNICATIONS OVER NOISY CHANNELS

by

Charlie Qing Yang

B.S.E.E., University of Science and Technology of China, 1982

M.A.Sc., University of Victoria, 1990

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Electrical and Computer Engineering

We accept this dissertation as conforming to the required standard

---

Dr. Vijay K. Bhargava, Supervisor (Department of ECE)

Dr. Qiang Wang, Departmental Member (Department of ECE)

Dr. Panajotis Agathoklis, Departmental Member (Department of ECE)

Dr. Hans A. Müller, Outside Member (Department of Computer Science)

Dr. Stephen B. Wicker, External Examiner (Georgia Institute of Technology)

Supervisor: Dr. Vijay K. Bhargava

ABSTRACT

Hybrid error control techniques to improve data communication performance for noisy channels have been extensively studied. However, a growing concern in communication system design is the impact of delay due to retransmissions and/or delay-prone technologies on system performance. Previous analyses have not considered various delay aspects of a hybrid error control system. Efficient error control techniques which are able to provide improved coding gain and throughput by promptly matching the error correction coding capability with the changing channel conditions have yet to be developed and investigated.

In this thesis, delay-related performance characteristics are investigated for asynchronous time division multiplexing links. Two different methods based on an imbedded Markov chain model are developed and applied to the system with a noisy feedback channel, yielding analytical expressions for the buffer occupancy and the block delay. A recursive expression for packet loss probability for systems with a finite transmitter buffer is obtained.

The concept of delay-limited error control coding is introduced for real-time communications. Performance improvement by truncation of a type-II hybrid-ARQ protocol with one retransmission is investigated in detail. It is shown that the truncated protocol has a bounded delay and bounded queue length under typical communication traffic conditions. The error performance of the truncated protocol is further analyzed for various mobile fading channels.

Matched-rate hybrid error control coding for both adaptive and non-adaptive cases is also studied. A new adaptive error control protocol using Reed-Solomon codes is proposed. The protocol uses novel feedback transmissions to achieve faster estimation of channel states. Numerical optimization is carried out by introduc-

ing overall throughput and modified throughput as efficiency criteria. Based on channel bit error rate measurement, optimum overall throughput is obtained with minimum implementation complexity.

Our general conclusions are: (1) Both delay and packet loss can be greatly reduced by incorporating a Reed-Solomon code into the data-link protocol for noisy channels. (2) The truncated hybrid error control protocol can provide coding gain improvement and reduced delay over the conventional (untruncated) protocol. (3) Throughput efficiency of a type-I or type-II hybrid-ARQ protocol can be significantly improved by using the proposed matched-rate error control techniques.

Examiners:

Dr. Vijay K. Bhargava, Supervisor (Department of ECE)

Dr. Qiang Wang, Departmental Member (Department of ECE)

Dr. Panajotis Agathoklis, Departmental Member (Department of ECE)

Dr. Hans A. Müller, Outside Member (Department of Computer Science)

Dr. Stephen B. Wicker, External Examiner (Georgia Institute of Technology)

Dr. Stephen B. Wicker, External Examiner (Georgia Institute of Technology)

Contents

List of Tables

List of Figures

Acknowledgments

I feel deeply indebted to my supervisor, Dr. Vijay K. Bhargava, for having offered me the opportunity to pursue my graduate studies in Victoria and introducing me to the subject of error control coding. Without his support and encouragement throughout my research, this work would never have been possible.

I would like to thank Drs. Q. Wang, P. Agathoklis, and H. A. Müller for their services as my supervisory committee members, and Dr. S. B. Wicker for his valuable comments on the work I have done.

My thanks are also extended to all the faculty members and staff in the ECE department. The help they offered me during my Ph.D. program is deeply appreciated.

I thank my colleagues at the Digital Communication Group, which is under the guidance of Dr. V. K. Bhargava and Dr. Q. Wang, for creating the friendly atmosphere in which I have had the pleasure to work. In particular, I wish to thank Mr. Gordon Webster and Mr. Dave Peterson for their helpful suggestions which have improved the quality of this thesis.

This research is supported by the Canadian Institute for Telecommunications Research, a Federal Network of Centers of Excellence, a postgraduate scholarship from the Natural Science and Engineering Research Council of Canada, a University of Victoria Fellowship, and a President's Research Scholarship.

Finally, I wish to dedicate this dissertation to my parents, Dr. Junhua Yang and Dr. Chong Zhao, my wife Julie Qing Zhu and my daughter Yifei Yang. Their love and care made all my work worthwhile.

Without delay constraint,
electrical communication makes no sense.
To achieve
real-time communication
over a noisy channel
remains
the most challenging problem
to researchers
in the area of digital communications.

Chapter 1

Introduction

1.1 Motivation for Research

The study of error control coding began in 1948 with the publication of Claude Shannon's famous paper [98]. Shannon demonstrated the existence of codes achieving reliable communication whenever the *data rate* is smaller than a threshold C called the *channel capacity*. For the additive white Gaussian noise (AWGN) channel, the channel capacity is given by

$$C = B\log_2\left[1+\frac{S}{N}\right], \tag{1.1}$$

where B is the channel bandwidth and S/N is the ratio of signal to noise power falling within the bandwidth. This remarkable result indicates that the ultimate performance limit caused by channel noise is not reliability, as generally believed before Shannon's work, but the rate at which data can be reliably transmitted.

The concept of channel capacity is fundamental to communication theory and is surprisingly powerful and general. It can be applied to a large class of channel models, whether memoryless or not, discrete or nondiscrete. However, Shannon's celebrated coding theorems are only existence theorems; they do not show how promising coding schemes can be constructed. Since the publication of Shannon's paper, a considerable amount of research has addressed the design and analysis of practical coding and decoding techniques permitting reliable communication at the data rates promised by the theory [9] [11] [17].

Today, error control coding is an area of increasing importance in digital communications. System planners and designers must use sophisticated techniques to cope with the ever-increasing demand for digital communication services. Perhaps the most important advantage digital communications has over its analog counterpart is the ease with which coding can be implemented to significantly improve communication performance.

The efficiency and reliability of a data communication system depend heavily on the chosen data-link protocol. The basic function of a data-link protocol is to provide an apparently error-free link between communication nodes in a network. Most existing error control protocols, such as high data link control (HDLC), employ cyclic redundancy check (CRC) bits to detect errors [97]. This type of error control, which is called Automatic Repeat reQuest (ARQ), involves segmenting the bit stream into blocks and adding error detection redundancy bits to each block. Whenever errors are detected, the blocks detected in error are retransmitted.

A major concern in mobile data communications is the control of transmission errors when channel conditions are poor due to multipath fading, Doppler effects and other interference. Under poor conditions in mobile data links, simple ARQ-based protocols often provide very low channel efficiency. In recent years, many hybrid error control protocols[1] have been proposed to improve the channel efficiency [1] – [123]. These more sophisticated protocols make use of both error detection and error correction coding in order to achieve high throughputs and low undetected error probabilities for two-way mobile data communications.

Most of us have a reasonably good idea of the role of feedback in control systems, where it is easier to make recursive corrections than open loop corrections. Feedback also plays an important role in communications by simplifying the design of error control coding. For example, HDLC with a suitably large buffer has been

[1] Also called *hybrid-ARQ schemes* in the literature.

widely used for data transmission [10]. Moreover, a communications system using a hybrid protocol may be more reliable than a system using only pure forward error correction, and have a higher throughput than a system with retransmission only [68]. Since the error statistics on most real channels vary with time, adaptive error control coding has recently become an active research topic [61] [62] [73] [100] [109] [110]. Obviously, adaptive error control must utilize some type of feedback protocol.

In the communications context, a feedback channel is often available, but both transmission time and system complexity may be increased if feedback is utilized. Proper use of feedback can greatly affect the trade-offs in a communication system design. However, to achieve real-time communication over a noisy channel remains the most challenging problem to researchers in the area of hybrid error control techniques. For example, the next-generation of cellular and microcellular systems are expected to provide efficient wireless transmission of speech, data, image, facsimile and video signals between portable terminals and the wireline network within real-time requirements. Here we use the word "real-time"[2] to mean that information is delivered to the end user within the desired time period. As will be shown in this work, hybrid error control techniques can offer promising solutions to various real-time communications challenges.

Most performance analyses for error control coding reported in the literature have been carried out by one of two research groups. One group has focused on various ARQ and hybrid-ARQ protocols. For performance evaluation of these feedback protocols, throughput and undetected error probability are normally used. The other group has only been concerned with forward error correction, using coding gain as a measure of the system performance. Common performance measures are lacking, probably due to the fact that in the early days the applications of FEC were mainly motivated by satellite communications [101], while ARQ was

[2]According to Webster's Ninth New Collegiate Dictionary, real-time is the actual time during which something takes place.

very popular in computer data communications [107]. Consequently, it has been difficult to compare the performance of various coding schemes with and without feedback.

We believe that much more research needs to be done in applying common standards to effectively evaluate and compare the performance of different coding schemes. More promising techniques can then be identified and developed to improve system performance in a cost-effective manner. To pursue this ultimate goal, we are going to study various performance aspects of hybrid error control coding techniques for digital communications in this work.

1.2 Contributions of the Dissertation

Hybrid error control techniques to improve data communications performance for noisy channels have been extensively studied. However, a growing concern in communication system design is the impact of delay due to retransmissions and/or delay-prone technologies on system performance. Previous analyses have not considered various delay aspects of a hybrid error control system. Efficient error control techniques which are able to provide improved coding gain and throughput by promptly matching the error correction coding capability with the changing channel conditions have yet to be developed and investigated.

In this work, the following three major contributions concerning improved performance of hybrid error control techniques are reported.

1. Delay-related performance characteristics are investigated for asynchronous time division multiplexing (ATDM) links with a hybrid-ARQ error control protocol. The protocol employs a Reed-Solomon code for both error detection and error correction. The system is modeled as an imbedded Markov chain with either finite or infinite transmitter buffer. Two different methods based on the imbedded Markov chain model are applied to the system with a noisy feedback channel, yielding analytical expressions for the buffer occupancy

and the block delay. A recursive expression for packet loss probability for systems with a finite transmitter buffer is also obtained.

2. The concept of delay-limited error control coding is introduced. In particular, performance improvement by truncation of a type-II hybrid-ARQ protocol with one retransmission is investigated in detail. It is shown that the truncated type-II hybrid protocol has a bounded delay and bounded queue length under typical communication traffic conditions. The error performance of the truncated protocol is further analyzed for mobile channels with *Rayleigh*, *Rice*, *Log-Normal*, and normalized *Nakagami-m* fading distributions. Meanwhile, an analytical approach is developed to compare the power savings of various error control protocols, including the truncated protocol and the conventional (untruncated) protocols.

3. A new adaptive error control protocol using Reed-Solomon codes is proposed and its throughput performance for a binary symmetric channel (BSC) with a time-varying channel error probability is analyzed. The novel protocol makes use of received blocks sent back to the transmitter to achieve faster estimation of channel states in a two-way data communication link. The throughput capacity of hybrid error control protocols is identified and an empirical adaptive algorithm is proposed. Matched-rate error control coding for non-adaptive cases is also studied. A novel throughput criterion is proposed to evaluate performance based on channel bit error rate measurement, and numerical optimization of the performance is undertaken.

The performance analysis results reported in this work have shown that the proposed hybrid error control techniques offer improved performance in the following three aspects.

1. Both delay and packet loss can be greatly reduced by incorporating a Reed-Solomon code into the data-link protocol, especially for mobile data links,

where severe fading may result in very poor performance when a pure ARQ protocol is used.

2. The truncated protocol provides coding gain improvement over the untruncated type-II hybrid-ARQ protocol. The truncated protocol is therefore best suited to power- and delay-limited mobile applications.

3. Throughput efficiency of a type-I or type-II hybrid-ARQ protocol can be significantly improved by using the matched-rate error control coding techniques.

1.3 Organization of the Dissertation

An outline of the remainder of this dissertation is as follows:

- *Chapter 2* describes the fundamental concepts in hybrid error control research and introduces important methods for performance evaluation in later chapters.

- *Chapter 3* investigates the impact of a noisy feedback channel on transmission delay and packet loss for ATDM links with a hybrid-ARQ error control protocol and a finite transmitter buffer.

- *Chapter 4* studies the efficient use of a small number of feedback signals (with or without retransmissions) to achieve real-time communications. In particular, time delay of a truncated type-II hybrid-ARQ protocol with one retransmission is analyzed by using the method developed in Chapter 3.

- *Chapter 5* employs a unified analytical approach to evaluate the coding gain of truncated protocols on various mobile fading channels. The results are compared with the performance of a conventional (untruncated) protocol.

- *Chapter 6* proposes a new adaptive error control protocol with feedback transmissions, and analyzes its throughput performance for a BSC with a time-varying channel error probability. The matched-rate throughput capacity of hybrid error control protocols is also discussed.

- *Chapter 7* develops a pragmatic method for numerical optimization of overall throughput. By numerical optimization, the optimum system parameters are found for type-I hybrid-ARQ with BCH codes and RS codes.

- *Chapter 8* concludes this dissertation and provides suggestions for future work which will extend the results of our investigation.

Chapter 2

Fundamentals of Hybrid Error Control

In this chapter we give a historic overview of error control for digital communications and examine the role of feedback in communication. The fundamental issues presented in this chapter form the basis of the whole thesis and most of them are used in later chapters.

2.1 The Classic Coding Problem

2.1.1 Forward Error Correction Coding

Error control coding is concerned with methods of delivering information from a source to a destination with a minimum number of errors. There are three basic techniques for error control coding which are used in communication systems: forward error correction (FEC), ARQ and a proper combination of FEC with ARQ (hybrid) [11] [68]. All three techniques add redundancy to data prior to transmission in order to reduce the effect of errors at the receiver. FEC does not need to have a feedback channel, while ARQ requires a feedback channel for necessary retransmissions.

The communication system depicted in Figure 2.1 employs FEC. The source generates data bits or messages that must be transmitted to a distant user over a noisy channel. Generally speaking, a specific signal is assigned to each of M

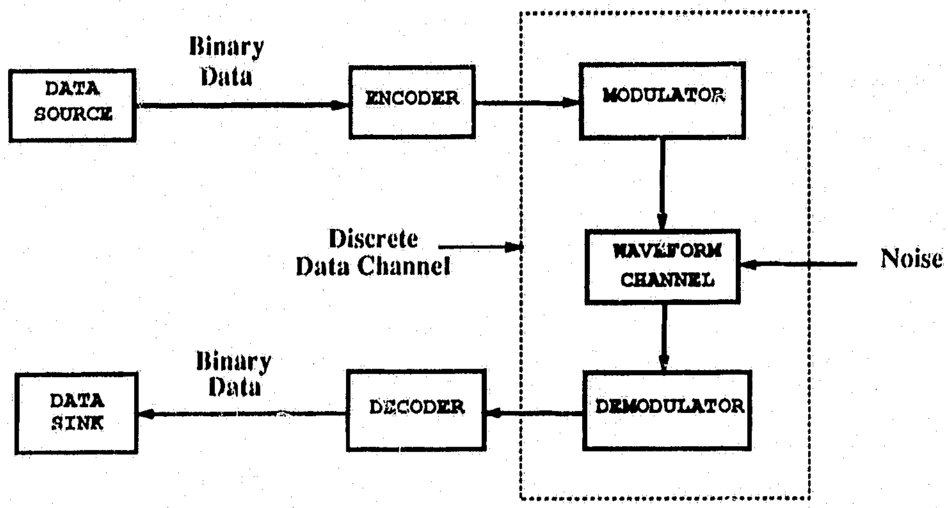

Figure 2.1: Digital communication using forward error correction coding

possible messages that can be emitted by the source. The selection rule that assigns a transmitted signal to each message is the *code*. The *encoder* implements the selection rule, while the *decoder* performs the corresponding inverse mapping. Because of channel noise, the transmitted signals may not arrive at the receiver exactly as transmitted, causing errors to occur at the decoder input. A natural design objective is to select a code that will permit most of the errors to be *corrected* by the decoder, thereby providing an acceptable level of reliability.

Coding is a design technique which can fundamentally change the trade-offs in a digital communication system. The most trivial example of coding is the repetition of the same message on the transmission channel. Here it is clear that redundancy, and therefore reliability, is obtained at the expense of transmission efficiency, or bandwidth utilization. In general, error control coding can increase signal quality from problematic to acceptable levels. If the attendant increases in complexity at the transmitter and receiver is economically justifiable, and bandwidth utilization is not unduly compromised, useful performance improvements may result. For example, with coding, less power may be required to communicate between a satellite and a mobile terminal. Furthermore, coding may result in an increase in the maximum number of mobile terminals per satellite.

2.1.2 Basic Coding Process

FEC coding systems have traditionally been divided into *block* and *convolutional* error-correction techniques.

In an (n, k) linear block code, a sequence of k information bits is used to obtain a set of $n - k$ parity bits, yielding an encoded block of n bits. Usually modulo-2 arithmetic is used to compute the parity bits. Modulo-2 arithmetic is particularly suited to digital logic; addition corresponds to the EXCLUSIVE-OR operation, while multiplication can be realized as an AND operation. The code rate r is defined as $r = k/n$ and n is called the block length. Linear codes form a linear vector space; two code words can be added (modulo 2) to produce a third code word.

The Hamming weight of a code word c is defined to be the number of nonzero components of c. For example, the code word c =(110101) has a Hamming weight of 4. The Hamming distance between two code words c_1 and c_2, denoted $d(c_1, c_2)$, is the number of positions in which they differ. For example if c_1 = (110101) and c_2 = (111000) then $d(c_1, c_2) = 3$. The *minimum distance* d of a linear block code is equal to the minimum weight of its nonzero code words. A code can correct all patterns of t or fewer random errors and detect all patterns having no more than s errors, provided that $s + 2t + 1 \leq d$. If the code is used for error correction alone, any pattern of t or fewer random errors can be corrected, provided that $2t + 1 \leq d$.

A convolutional code of rate $1/v$ may be generated by a K stage shift register and v modulo-2 adders. For each input information bit, the output of the modulo-2 adders provides v channel bits. The *constraint length* of the code is defined as the number of shifts over which a single information bit can influence the encoder output. For the simple binary convolutional code, the constraint length is equal to K, the length of the shift register.

Whether block coding or convolutional coding is used, the encoded sequence is mapped to suitable waveforms by the modulator and transmitted over the noisy channel. The physical (or waveform) channel consists of all the hardware (for example, filtering and amplification devices) and the physical media that the waveform passes through, from the output of the modulator to the input of the demodulator.

The demodulator estimates which of the possible symbols was transmitted based upon an observation of the received signal. Finally, the decoder estimates the transmitted information sequence from the demodulator output. The decoder makes use of the fact that the transmitted sequence is composed of codewords. Transmission errors are likely to result in the reception of a noncode sequence.

2.1.3 Coding Gain

It is often useful to express coding performance not in terms of the error rate reduction for a given signal-to-noise ratio (SNR), but as the SNR difference at a fixed bit error rate. Consider an AWGN channel with one-sided noise spectral density N_0 and no restriction on bandwidth. Let E_b denote the received energy per bit. It can be shown [11] that if the SNR E_b/N_0 exceeds $-1.6\ dB$, there exists a coding scheme which allows error-free communication, while reliable communication is not generally possible at lower signal-to-noise ratios. On the other hand, it is well known that uncoded phase shift keying (PSK) modulation over the same channel requires about $9.6\ dB$ to achieve a bit error rate of 10^{-5}. Thus, as shown in Figure 2.2, a potential coding gain of $11.2\ dB$ is theoretically possible.

Coding gain (in decibels, or dB) is defined as the difference in values of E_b/N_0 required to attain a particular error rate with and without coding [12]. Notice that coding gain is often obtained at the expense of transmission bandwidth[1]. The bandwidth expansion is the reciprocal of the code rate. Coding schemes delivering 2 to 8 dB coding gain are widely used in modern digital communication systems. This is because of the phenomenal decrease in the cost of digital hardware and the much less significant decrease in the cost of analog components such as power

[1]Except trellis-coded modulation.

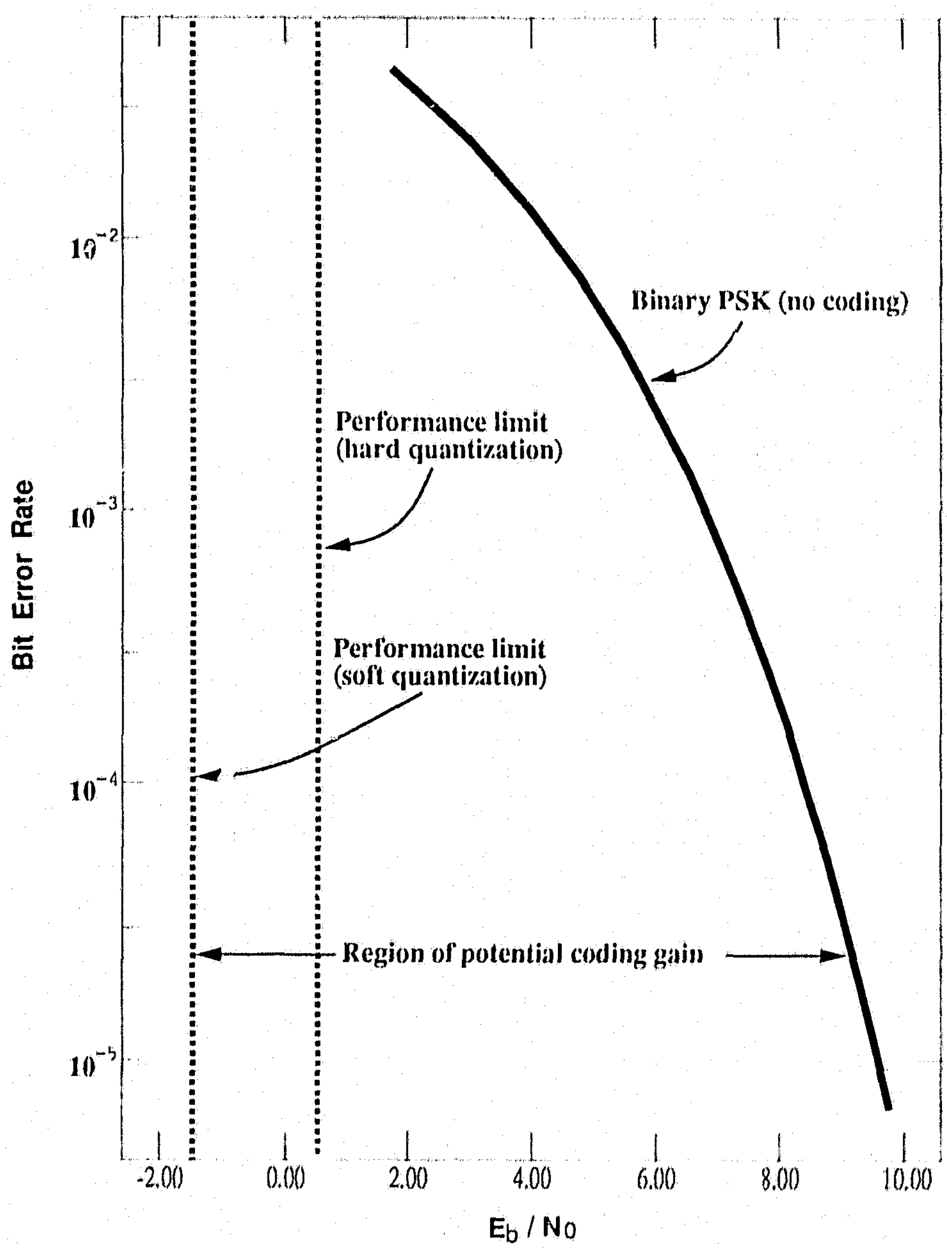

Figure 2.2: Performance of uncoded PSK over AWGN channel

amplifiers, antennae and so on.

Practical communication systems rarely provide the ability to make full use of the actual analog voltages of the received signal. The normal practice is to quantize these voltages. If binary quantization is used, we say that a *hard decision* is made at the receiver as to which level was actually sent. For example, in coherent PSK with equally likely transmitted symbols, the optimum threshold is zero [12]. The demodulator output is a one or a zero depending on whether the voltage is above or below the threshold. With coding, it is desirable to maintain an indication of the reliability of this decision. A *soft-decision* demodulator first decides whether the voltage is above or below the decision threshold, and then computes a "confidence" number which specifies how far from the decision threshold the demodulator output is. This number in theory could be an analog quantity, but in most practical applications a three-bit (eight-level) quantization is used. It is known that soft decision decoding is about 3 dB more efficient than hard decision decoding at very high E_b/N_0. A figure of 2 dB is more likely at realistic values of E_b/N_0.

2.2 From ARQ to Hybrid Error Control

2.2.1 The Idea of ARQ and Hybrid-ARQ

In an *automatic repeat request* (ARQ) system, whenever the receiver detects an error in the transmitted message, it sends a retransmission request to the transmitter over a feedback channel. These requests are repeated until the message is received correctly. Three basic types of ARQ protocols are commonly used: stop-and-wait, go-back-N, and selective-repeat [26] [68] [97].

Because of its simplicity, ARQ is used in many data communications systems. However, the technique has a major shortcoming: the throughput efficiency may be highly dependent on channel conditions. At low signal-to-noise ratios, the number of retransmissions required before a message is received correctly may be very large, and hence involve a long time delay. This delay is unacceptable in some

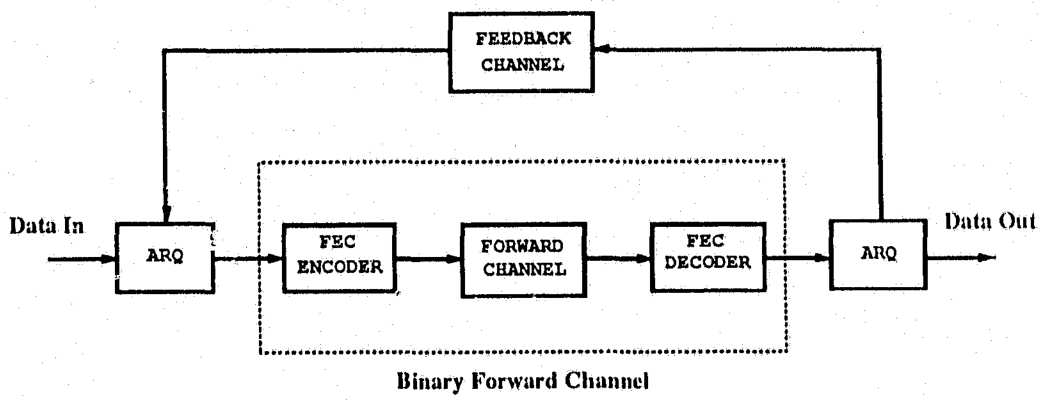

Figure 2.3: A hybrid-ARQ error control coding system

applications. One approach to reducing the time delay is the *hybrid-ARQ* protocol of Figure 2.3. Here an ARQ protocol is used to obtain a desired error rate. FEC coding is used to correct low-weight error patterns in each message, reducing the number of retransmission requests.

In a *type-I hybrid-ARQ* system, the message and error detecting parity bits generated by the ARQ protocol are further encoded with an FEC code [91]. At the receiver, the error correction parity bits are used to correct channel errors. The FEC decoder produces an estimate of the message and the error detection parity bits. The result is then tested by the error detection system to determine if the message should be accepted as error free, or rejected as containing errors. If the channel signal strength is poor (high bit error rate), or if the message is long, the probability of error-free transmission may approach zero. Under these conditions, the efficiency may be improved by using a type-I hybrid protocol rather than a simple ARQ protocol. However, if signal strength is adequate, the type-I hybrid protocol involves a waste of bandwidth, because the error correction parity bits are unnecessary.

In a *type-II hybrid-ARQ* system [69], the first transmission of a message is coded with error detection parity bits alone, as in a standard ARQ protocol. If the receiver detects errors in the received block, it saves the erroneous block in a buffer and requests a retransmission. The retransmitted information is not the original coded message, but a block of parity bits derived by applying an FEC code to the message. The receiver uses these parity bits (which may themselves be in error) to correct the block stored in the receiver buffer. If error correction is not successful, subsequent retransmissions are requested, which may consist of the original codeword or another block of error correction parity bits. The retransmission format depends on the strategy and on the error correction code used. The intention of type-II hybrid protocols is to provide the efficiency of standard ARQ under good channel conditions while achieving the performance improvement of type-I hybrid protocols under poor channel conditions.

2.2.2 Rate-Adaptive Hybrid Error Control

In order to cope with diverse system requirements and service options, *rate-adaptive hybrid error control* protocols, in which the code rate is adapted to suit prevailing transmission conditions [75] [115] [123], are very attractive. Such protocols are applicable to satellite or mobile communication systems, where adaptive signal compensation may provide adequate performance when channel conditions are poor (due to rainfall or fading, etc.) without a permanent bandwidth reduction caused by excessive coding overhead. In a rate-adaptive coding scheme, a high rate code is used under good channel conditions to maintain a high information rate, but as the channel condition worsens, the information rate is reduced by applying lower rate codes to maintain the desired reliability.

Traditional convolutional codes are not well suited to rate-adaptive coding techniques, because only low rate codes have been used. However, as data transmission rates increase, the need increases for good high rate convolutional codes with practical encoding and decoding techniques. *High rate punctured convolutional codes* are a significant recent development in which the difficulties usually associated with decoding high rate codes are significantly reduced [19] [44]. Viterbi or sequential decoding of rate b/v punctured convolutional codes is no more complex than the decoding of rate $1/v$ codes. In fact, the punctured code is obtained in a straightforward manner from a "parent" rate $1/v$ code, while a Viterbi or sequential decoder for the latter code can be easily modified to decode the punctured code. This property makes punctured convolutional codes attractive for rate-adaptive applications - a single decoder can accommodate the entire family of punctured codes arising from a single parent code.

Among the class of block codes, *maximum distance separable* (MDS) codes are particularly well suited to rate-adaptive coding techniques. An (n, k) code for which the minimum distance equals $n - k + 1$ is said to be maximum distance separable. MDS codes are optimal in the sense that no (n, k) code has minimum Hamming distance exceeding $n - k + 1$. The MDS codes most often encountered in practice are Reed-Solomon (RS) codes. RS codes are attractive not only because of the ease with which they can be decoded using Berlekamp's decoding algorithm, but also because they feature a wide variety of possible code rates and block lengths. RS codes are suitable for rate-adaptive coding schemes because a low rate parent RS code can be broken down into several high rate subcodes, in much the same way that a parent convolutional code yields high rate punctured codes. The subcodes are themselves linear block codes having the MDS property. With appropriate modifications, the decoder for the parent code can also be used to decode the subcodes. Thus a single decoder may be used in a rate adaptive system where the transmitter employs the subcode most suitable for the current channel state.

2.3 Capacity of Communication Channels with Feedback

2.3.1 Memoryless Channel Models

A *model* of a communication channel is a mathematical representation defined to approximately describe the channel characteristics. It is conventional to define a channel model to include the modulator, the demodulator, and all the intervening transmission equipment and media [81]. A *discrete memoryless channel* (DMC) is defined by an M-ary set of input symbols $\{x_i, (i = 0, 1, ..., M-1)\}$, a Q-ary set of output symbols $\{y_j, (j = 0, 1, ..., Q-1)\}$, and a set of conditional probabilities, called *transition probabilities*, which we can write as

$$P(y = y_j | x = x_i) = P(y_j | x_i). \tag{2.1}$$

The *binary symmetric channel* (BSC) is an important DMC model having a binary input, a binary output, and transition probabilities given by

$$\begin{aligned} P(y = 1 | x = 0) = P(y = 0 | x = 1) &= p_b \\ P(y = 0 | x = 0) = P(y = 1 | x = 1) &= 1 - p_b, \end{aligned} \tag{2.2}$$

where p_b is called the *bit error probability* and $0 \leq p_b \leq 1$.

Another important DMC model is the *additive white Gaussian noise* (AWGN) channel, which can be described by

$$y = x + n_G, \tag{2.3}$$

where n_G is a zero-mean Gaussian random variable with variance σ^2, and the input x can have any one of M discrete values. That is, the conditional probability density function of the output y, with an input x, is given by

$$p(y | x = x_i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(y - x_i)^2 / 2\sigma^2}. \tag{2.4}$$

For the AWGN channel model, the probability of a received bit being in error is not dependent on whether one or more preceding bits were in error. As the probability of error on each received bit is the same, the performance of this model can be completely described by a single number, which is denoted by p_b as in the BSC model.

2.3.2 Feedback for Memoryless Channels

The idea of hybrid error control brings up the fundamental question of the role of feedback in communication.

A measure of information for general sources was provided by Shannon [98] with the application of the concept of *entropy* to an information source. Consider a source which produces any one of M symbols $x_1, x_2, ..., x_M$, where the probabilities of occurrence are $P(x_1), P(x_2), ..., P(x_M)$; the entropy of the source is defined by

$$H = -\sum_{i=1}^{M} P(x_i) log P(x_i). \tag{2.5}$$

H represents the uncertainty about the information before it is received from the source. In the case of error-free transmission, the user receives uncorrupted source sy bols, and the uncertainty about the source information vanishes completely as symbols are received. In this ideal case, the channel transfers information from the source to the user at an average rate of H bits per symbol.

In all realistic cases, of course, transmission through the channel is not error-free, and thus after reception the user is left with some residual uncertainty. Minimizing this residual uncertainty while making efficient use of signal energy is the essence of the communication system design problem.

Consider a DMC model with a set of outputs $y_1, y_2, ..., y_Q$. The *mutual information* associated with the transmission of x_i and reception of y_j is defined by [40]

$$I(x_i; y_j) = log \frac{P(y_j|x_i)}{P(y_j)}. \tag{2.6}$$

The overall rate of transfer of information through the channel can then be calculated by simply averaging $I(x_i; y_j)$ over all possible values of x_i and y_j, giving us the *average mutual information*, defined by

$$I(X;Y) = \sum_{i=1}^{M}\sum_{j=1}^{Q} P(x_i, y_j) log \frac{P(y_j|x_i)}{P(y_j)} \tag{2.7}$$

which has units of bits per symbol when the logarithm base is 2. The *capacity* of a discrete memoryless channel is defined as the maximum value of $I(X;Y)$ with respect to all input distributions, that is,

$$C = \max_{P(x_i)} I(X;Y). \tag{2.8}$$

The most remarkable of Shannon's results is the *channel coding theorem* which

The most remarkable of Shannon's results is the *channel coding theorem* which

shows that it is possible (but not how!) to transmit data with an arbitrarily small error rate, as long as the data rate $R_d \leq C$. As an example, it can be easily shown that the capacity of the BSC is given by

$$C_{BSC} = 1 + p_b log p_b + (1 - p_b) log(1 - p_b) = 1 - H. \tag{2.9}$$

If the bit error probability $p = 0$, $C_{BSC} = 1$, i.e., we can transmit 1 bit per symbol. On the other hand, if $p_b = 0.5$, $C_{BSC} = 0$, and the channel fails to transmit any information at all. Therefore, in using the BSC model, we need only consider $0 \leq p_b \leq 0.5$.

A remarkable conclusion about feedback, which is also due to Shannon [99], is that *feedback does not improve the capacity of memoryless channels.* However, most real communication channels have been found to exhibit memory. Even when considering the pure memoryless case, there are numerous results that indicate that feedback may reduce the coding complexity. The simplest example of this might be the binary erasure channel, as shown in Figure 2.4, in which one sends $X \in \{0,1\}$ and receives $Y \in \{0,1,e\}$, where e denotes erasure. Suppose $Y = X$ with probability $1 - \delta$, and $Y = e$ with probability δ. It is easy to show

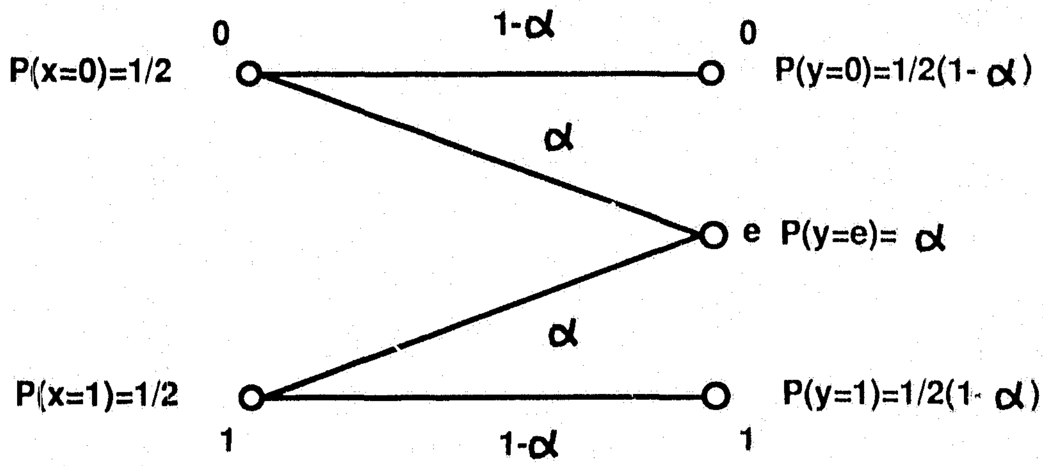

Figure 2.4: Capacity of the binary erasure channel

that the capacity of the channel is $C = 1 - \delta$, but is difficult to achieve this capacity without feedback. When feedback is used, however, the intended symbol can be retransmitted whenever an erasure is received. The expected number of transmissions required to reveal a given symbol X is $1/(1-\delta)$. Thus $(1-\delta)n$ transmitted bits require n transmissions and the resulting capacity is $C = 1 - \delta$.

2.3.3 Feedback for Channels with Memory

It has been recognized [47] that when *real systems* are measured *under real operating conditions* in the field, the errors are found to arrive mostly in bursts, a behavior which is not in accord with the AWGN model, and which cannot be well described by a bit error probability. Such channels are said to exhibit *memory*, i.e., statistical dependence in the occurrence of errors.

To control the errors effectively through some coding technique, it is necessary to study the statistical dependence of the errors. The fact that the errors tend to occur in bursts, i.e., somewhat "predictably," should prove to be advantageous. In information theory parlance, this means channel memory increases capacity. But how can memory be effectively exploited to realize the additional capacity? This basic question has provided motivation for the development of channel models which reflect the statistical dependence of error occurrences [52]. Often, channels with memory are modeled by using a Markov chain consisting of a finite number of states with defined transition probabilities. Such models attempt to simulate the transitions in real channel behavior from good to bad, and vice versa.

As was pointed out in [52], one underlying reason that led to the development of a variety of channel models is the quest to obtain a model that represents real channels "better," in the sense that a model should capture essential behavior of a real channel. With successive refinements, it is of course natural that these models tend to become increasingly complicated, sometimes becoming so unmanageable as to be of little practical use for the original intended purpose of being an intermediate step.

It seems doubtful that any detailed models which could be constructed would have general validity, even if enough data could be collected to construct them [47]. Our purpose in using a channel model is *not* to approximate a real error source as closely as possible, but rather , with minimal complexity, to obtain the major channel characteristics with the statistics necessary for evaluation of error control schemes. In this work, therefore, we will focus on performance evaluation by using only the BSC and some basic fading channel models. For real channels with memory, ideal interleaving will be assumed. Nevertheless, we would still like to have a look at the role of feedback for channels with memory.

Consider the capacity of time-varying additive Gaussian noise channels with feedback. It has been shown [25] [31] that the feedback capacity C_{FB} and the nonfeedback capacity C satisfy the inequalities

$$C_{FB} \leq 2C \tag{2.10}$$

and

$$C_{FB} \leq C + \frac{1}{2} \tag{2.11}$$

in bits per symbol. We can see that feedback can at most double the capacity, and at most one half bit per symbol can be added for an additive Gaussian noise channel. Therefore, *feedback could increa~: the capacity of channels with memory.*

It has recently been shown [63] that the channel capacity in a Rayleigh fading environment [57] is always lower than in a Gaussian noise environment. However, the calculation was carried out in an average sense which did not take into account the channel memory.

We wish to clarify the point that *memory increases channel capacity for binary discrete channels.* It has been shown [116] that the discrete memoryless process has maximum entropy among the class of binary discrete stochastic processes with error rate p and arbitrary memory length. In other words, the entropy for the binary channels with memory has an upper bound, i.e.,

$$H \leq H_0, \tag{2.12}$$

where H_0 is the entropy for a DMC with error rate p_b given by

$$H_0 = -p_b log p_b - (1 - p_b) log(1 - p_b). \tag{2.13}$$

A process with a given error rate p can be said to have memory if its H is less than H_0. The measure for memory, denoted as θ, can be defined by

$$\theta = \frac{H_0 - H}{H_0}. \tag{2.14}$$

Thus a memoryless ($\theta = 0$) process has maximum entropy H_0, while a process with $\theta = 1$ has zero entropy, i.e., there is no uncertainty about the relative error locations, such as in a cyclic process.

In the case of a binary communication channel with memory and arbitrary error rate p, the channel capacity is given by

$$C = 1 - H \quad \text{bit/symbol}. \tag{2.15}$$

Let C_0 denote the capacity for the BSC with error rate p, then

$$C_0 = 1 - H_0, \tag{2.16}$$

where H_0 is given by (2.13).

From (2.14), (2.15), and (2.16), we have

$$C = C_0 + \theta H_0. \tag{2.17}$$

The quantity θH_0, therefore, characterizes the additional capacity of the channel with memory compared to the BSC. An ARQ or hybrid-ARQ protocol is more efficient in a burst channel than a memoryless random channel. When a burst of very unreliable data is received, retransmissions could provide a more reliable group of symbols. Moreover, retransmissions can provide several copies of the message which could be used to obtain a soft-decision sequence for the decoder by using the code combining technique [21]. As a result, the additional capacity in a channel with memory could be realized by using feedback.

2.4 Methods for Channel Error Rate Estimation

Error detection with retransmissions is in fact *adaptive*, because transmission of redundant information is increased when errors occur. Again, feedback plays an important role during the coding adaptation. A plain ARQ protocol adapts slowly, and the quest for more rapid adaptation leads to the concept of using variable redundancy codes and forms the basis of type-II hybrid-ARQ protocols and other adaptive coding schemes.

Perhaps the most difficult problem in implementing an adaptive coding scheme is how to obtain a real-time adaptive algorithm. In other words, the channel error characteristics should be identified within a few block transmission times. Normally, channel noise is measured by using the "bit error rate" (BER), although this only makes sense if an AWGN channel model is assumed.

There exist many techniques for bit error rate measurement [103], but measuring the exact number of erroneous bits would require a lot of overhead and thus reduce the transmission efficiency. A very simple method for BER estimation is given as follows [77].

Let P_B denote the block error probability and p_b denote the bit error probability. For a received block with length n, we have

$$P_B = 1 - (1 - p_b)^n. \tag{2.18}$$

Assuming that $p_b \ll 1$ and that there is at most one erroneous bit per block, then

$$P_B = 1 - \{1 - \begin{pmatrix} n \\ 1 \end{pmatrix} p_b + \begin{pmatrix} n \\ 2 \end{pmatrix} p_b^2 - \ldots\} \leq np_b. \tag{2.19}$$

Often p_b is very small so that the above bound is very tight. Thus we have

$$p_b \approx \frac{P_B}{n}, \tag{2.20}$$

or

$$BER \approx \frac{TNBE}{TNB} \cdot \frac{1}{n}, \tag{2.21}$$

where $TNBE$ is the total number of blocks in error and TNB is the total number of blocks. By accumulating the values of $TNBE$ and TNB over a long period, the above formula does provide a real-time adaptive algorithm. However, as pointed out in [47], such a long term bit error rate is *completely* meaningless as a description of practical communication channels, because real channels often have errors occurring in bursts.

Other methods for rapid BER measurement include: (1) simply counting the errors in the known synchronization sequence inserted in each block; (2) comparing the bit-by-bit estimate obtained from the previous retransmitted blocks to the new retransmitted block; and (3) estimating the BER based on signal-to-noise estimations obtained from the demodulator.

Consider a general adaptive coding scheme using a code with several possible code rates which can be varied according to the channel condition. We believe that the most practical real-time adaptive algorithm for real channels would be to base the code rate on the previous transmission. For example, if the previous transmission is a retransmission, then the code rate should be decreased by one level to improve the error correction capability. If the previous transmission is a new transmission, then the code rate should be increased by one level. In Chapter 6 and Chapter 7, we propose more practical channel estimation methods which could achieve our goal of matching coding capability with the real channel conditions.

2.5 Performance Evaluation Methods

2.5.1 Reliability of Coding with Retransmissions

In the literature, the performance of various ARQ and hybrid-ARQ error control protocols is generally measured by reliability statistics and throughput efficiency. In an ARQ or hybrid-ARQ system, the receiver commits a decoding error whenever it accepts a received block with undetected errors. The *reliability* of a digital communication system is quantified by its decoding error probability, denoted as $Pr(E)$, which is defined as

$$Pr(E) = \frac{\text{probability of the occurrence of undetected errors}}{\text{probability that decoding succeeds}}. \tag{2.22}$$

Clearly, for an error control system to be reliable, the $Pr(E)$ should be made very small.

Consider an (n,k) linear block code, such as a CRC code, used for error detection in an ARQ or hybrid-ARQ protocol. Let us define the following probabilities:

- P_c — probability that a received block contains no errors.
- P_d — probability that a received block contains an uncorrectable but detectable error pattern.
- P_e — probability that a received block contains an undetectable error pattern.

The probability P_c depends on the channel error statistics, and the probabilities P_d and P_e depend on both the channel error statistics and the choice of the (n, k) error-detection code. P_e is normally called the undetected error probability of the code. Obviously, $P_c + P_d + P_e = 1$.

A decoded block is delivered to the end user only if it either contains no errors or contains an undetectable error pattern. Since an undetectable error pattern can occur on the initial transmission of a block or on any retransmission, $Pr(E)$ is given by

$$\begin{aligned} Pr(E) &= P_e + P_d P_e + P_d^2 P_e + \ldots \\ &= P_e \frac{1}{1 - P_d} = \frac{P_e}{P_c + P_e}. \end{aligned} \tag{2.23}$$

If the error-detection code is properly chosen, P_e can be made very small relative to P_c, and hence $Pr(E)$ can be made very small. Notice that P_d does not affect the reliability of an ARQ protocol. When a detectable error pattern occurs, the received block is not accepted by the receiver and a retransmission is requested.

For delay-sensitive applications, we may consider the same scheme with n retransmissions only. $Pr(E)$ is then given by

$$\begin{aligned} Pr(E) &= P_e + P_d P_e + P_d^2 P_e + \dots + P_d^n P_e \\ &= P_e \frac{1 - P_d^{n+1}}{1 - P_d} = \frac{P_e(1 - P_d^{n+1})}{P_c + P_e}. \end{aligned} \quad (2.24)$$

Consider now a basic ARQ protocol on a BSC channel with bit error rate p_b, we have

$$P_c = (1 - p_b)^n. \quad (2.25)$$

It has been proved that Hamming codes and the distance-5 primitive BCH codes have an undetected error probability P_e satisfying the upper bound [65] [66]

$$P_e \leq 2^{-(n-k)}. \quad (2.26)$$

More recently, Kasami *et al* proved that there exist codes with P_e satisfying the tighter bound [54]

$$P_e \leq 2^{-(n-k)}\{1 + (1 - 2p_b)^n - 2(1 - p_b)^n\}, \quad (2.27)$$

for $0 \leq p_b \leq 0.5$. The codes satisfying this bound include the distance-4 Hamming codes, the distance-6 primitive BCH codes of length $2^m - 1$ with $m \geq 5$, and the distance-8 primitive BCH codes of length $2^m - 1$ with m odd and $m \geq 5$. If a code satisfying the above bounds is used for error detection in an ARQ or hybrid-ARQ system, the $Pr(E)$ of the system can be made very small by using a moderate number of parity bits. For example, the (2047,2014) triple-error-correcting primitive BCH code satisfies the bound given by (2.27). Suppose that this code is used for error detection in a hybrid-ARQ system. Let $p = 10^{-3}$ (a typical high bit error rate). Then $P_c \approx 1.25 \times 10^{-1}$, and $P_e \leq 10^{-10}$. From (2.23), we have

$$Pr(E) = \frac{P_e}{P_c + P_e} \leq 8 \times 10^{-10}. \tag{2.28}$$

From this example we see that high system reliability can be achieved by a coding scheme with an ARQ protocol using very little parity overhead.

Consider next the performance of maximum distance separable (MDS) codes when they are used for error detection in various coding schemes with retransmissions [53]. The most important MDS codes are q-ary Reed-Solomon codes of length $n = q - 1$ [74] [90]. The RS codes make highly efficient use of redundancy, and block lengths and symbol sizes can be readily adjusted to accommodate a wide range of message sizes. RS codes also provide a wide range of code rates that can be chosen to optimize performance. In particular, any shortened RS code is also an MDS code. In addition, efficient decoding techniques are available for use with RS codes.

Another important property of an RS code is the fact that any k positions in the codeword may be used as an information set. A very useful consequence of this property is that it enables one to write down the exact weight distribution for any RS code. The weight distribution $\{A_i\}$ for an RS code or any MDS code defined on $GF(q)$ having block length n and minimum distance d is given by

$$A_i = \binom{n}{i}(q-1)\sum_{j=0}^{i-d}(-1)^j\binom{i-1}{j}q^{i-d-j}. \tag{2.29}$$

Derivation of this weight distribution formula can be found in [7] and [90].

A decoding procedure that corrects all error patterns of weight l or less and no others, where $l \leq t$ and t is the largest integer equal to or less than $(d-1)/2$, is called *bounded-distance decoding* [81]. If we assume the decoder to be a bounded distance decoder, then the weight distribution formula for the decodable words can be used to find the undetected error probability for MDS codes.

When a codeword c of a MDS code is transmitted over a communication channel, channel noise may corrupt the transmitted signals. As a result, the receiver receives the corrupted version of the transmitted codeword $c+e$, where e is an error pattern of some weight u. If $u \leq t$, then a bounded distance decoder at the receiver detects and corrects the error e and recovers c. If $u > t$, then the decoder fails and either

1. detects the presence of the error pattern but is unable to correct it, which corresponds to the probability P_d, or

2. decodes the received pattern $c+e$ incorrectly to some other codeword c' if the received pattern falls within the radius of c' with distance smaller than t, which corresponds to the probability P_e.

In coding schemes with ARQ protocols, case 2 is more serious than case 1. An undetected error can occur when an error pattern e is of weight $u \geq d-t$. Assume that all error patterns of weight u are equally probable. McEliece and Swanson offered an upper bound of 1/t! on $P_e(u)$, the conditional undetected error probability for RS codes (more generally, MDS codes), given that u symbol errors occur [78]. Later, Cheung obtained an exact formula for $P_e(u)$ [22]. Based on their results, we can calculate P_e, the probability of decoder undetected error, which is more useful when we apply RS codes to error control schemes with retransmissions.

For a q-ary symmetric channel, notice that

$$P_e = \sum_{u=d-t}^{n} P_e(u) \binom{n}{u} p_s^u (1-p_s)^{n-u}, \tag{2.30}$$

where p_s is the probability of channel symbol error. The undetected error probabilities of two typical RS codes showing the upper bound and the exact values

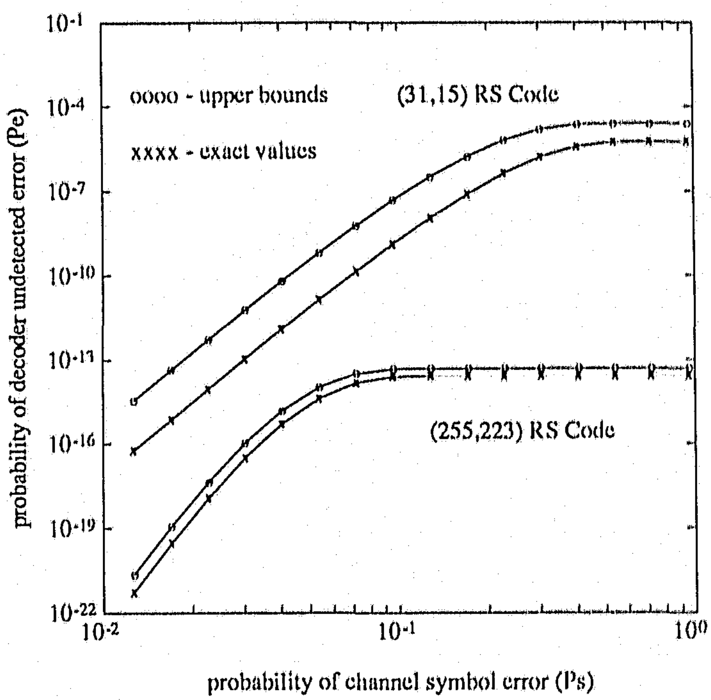

Figure 2.5: Decoder undetected error probability of RS codes

of $P_e(u)$ are plotted in Figure 2.5. Clearly, the upper bounds based on 1/t! are uniformly good but not very tight for small t. It can also be seen that RS codes can be very effective for error detection.

2.5.2 Throughput Efficiency

The *throughput*, denoted as η, of an ARQ or hybrid-ARQ system is defined as

$$\eta = \frac{\text{average number of information symbols accepted at the data sink per unit time}}{\text{total number of symbols that could be transmitted per unit time}}. \tag{2.31}$$

Assume that the forward channel in the system is a BSC with bit error rate p_b, and that the feedback channel is noiseless. Then the evaluation of throughput for an ARQ or hybrid-ARQ system is relatively easy when compared to the evaluation of other performance measures. For example, in a go-back-N (GBN) ARQ system, retransmission of an detected codeword in error always involves resending N codewords. Consequently, for a codeword to be successfully received, the average number of transmissions is

$$\begin{aligned} T_{GBN} &= (P_c + P_e) + (N+1)(P_c + P_e)(1 - P_c - P_e) + (2N+1)(P_c + P_e)(1 - P_c - P_e)^2 + \ldots \\ &= 1 + \frac{N(1 - P_c - P_e)}{P_c + P_e}. \end{aligned} \tag{2.32}$$

Therefore, the throughput of a go-back-N ARQ system is

$$\eta_{GBN} = \frac{1}{T_{GBN}}\left(\frac{k}{n}\right) = \left(\frac{k}{n}\right)\frac{P_c + P_e}{P_c + P_e + (1 - P_c - P_e)N}, \tag{2.33}$$

where P_c is given by (2.25). We see that the throughput of a go-back-N based error control protocol depends on both the channel error rate and the round-trip delay factor N. For a communication system with a high data rate and a long round-trip delay, N may become very large. As a result, $(1 - P_c - P_e)N$ becomes significantly larger than $(P_c + P_e)$ so that throughput quickly approaches to zero.

Now consider a selective-repeat (SR) ARQ system with an infinite buffer at the receiver to store the error-free codewords when a received block is detected in error. Such a system with an infinite buffer is called an *ideal selective-repeat ARQ* system. For a codeword to be accepted by the data sink, the average number of transmissions needed is

$$\begin{aligned} T_{SR} &= (P_c + P_e) + 2(P_c + P_e)(1 - P_c - P_e) + 3(P_c + P_e)(1 - P_c - P_e)^2 + \dots \\ &= \frac{1}{P_c + P_e}. \end{aligned} \tag{2.34}$$

Hence, the throughput of an ideal selective-repeat ARQ system is

$$\eta_{SR} = \frac{1}{T_{SR}}\left(\frac{k}{n}\right) = \left(\frac{k}{n}\right)(P_c + P_e). \tag{2.35}$$

We see that the throughput does not depend on the round-trip delay factor. As a result, selective-repeat ARQ offers significant benefits for satellite and long terrestrial channels. However, *the high throughput performance of ideal selective-repeat ARQ is achieved at the expense of extensive buffering, theoretically infinite buffering.* If a finite buffer is used at the receiver (as is the case in practical systems), *buffer overflow* may occur which reduces the throughput performance of the system.

The drawbacks of the ARQ and FEC coding schemes can be overcome if the two basic error control techniques are properly combined. Throughput performance of typical type-I and type-II hybrid-ARQ protocols have been derived in [91] and [69], respectively. Based on their results, we have plotted the throughput performance for hybrid-ARQ protocols as well as a plain ideal selective-repeat ARQ protocol in Figure 2.6. Also shown in Figure 2.6 is the throughput of forward error correction

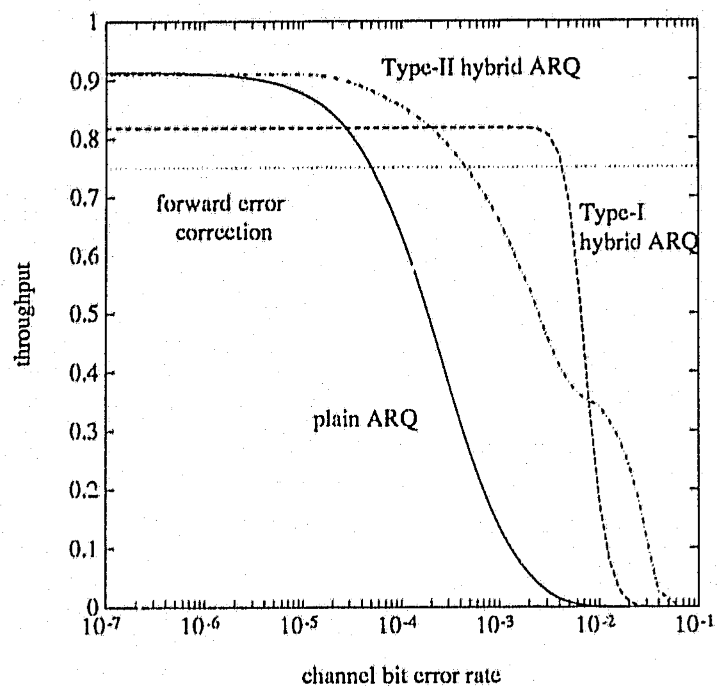

Figure 2.6: Throughput performance of various error control schemes

with code rate 3/4. Clearly, FEC provides the desired constant throughput at the expense of reduced reliability, while the throughput of various ARQ protocols approaches zero as the channel error rate increase.

In a hybrid-ARQ system, the following factors could affect the throughput efficiency of the system:

1. The length of the message blocks.

2. The error correction capability of the code.

3. The retransmission protocol.

4. The size of the buffers at both the transmitter and the receiver.

2.5.3 Analysis of Coding Gain

Although ARQ and FEC coding schemes share the same objectives: high reliability and high throughput of digital information, it has been difficult to quantitatively compare their relative performances. The reason is that the evaluation methods for ARQ and FEC coding schemes have traditionally been completely different.

In [87], the bit error probability of a BPSK modulated receiver on a BSC channel is generalized as

$$p_b = \frac{1}{2} erfc(\sqrt{\mu \frac{E_b}{N_0}}), \tag{2.36}$$

where $erfc(x)$ is the complementary error function defined by [12]

$$erfc(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} exp(-z^2)dz. \tag{2.37}$$

For FEC schemes, μ is the rate of the error correction code. For ARQ and hybrid-ARQ protocols, μ is the throughput.

By virtue of the above definition, the E_b/N_0 required to achieve the desired reliability can now be calculated for various error control schemes, including FEC, ARQ, hybrid-ARQ and any other hybrid error control schemes, on a common basis.

As an illustrative example, consider the hybrid-ARQ with BCH (127,106) code for $t = 1$, $t = 2$, and $t = 3$, and the corresponding FEC scheme as well as the selective-repeat ARQ protocol using the same code. The following formulas have been used to obtain the coding gain curves.

(1) For an FEC scheme, we have

$$\mu = \frac{k}{n}, \tag{2.38}$$

and the decoding error probability, $Pr(E)$, is upper bounded [67] as

$$Pr(E) \leq \sum_{i=t+1}^{n} \begin{pmatrix} n \\ i \end{pmatrix} p_b^i (1-p_b)^{n-i}, \tag{2.39}$$

where t is the error correcting capability of the BCH code.

(2) For the selective-repeat ARQ protocol, from (2.35) we have

$$\mu = \frac{k}{n}(P_c + P_e), \tag{2.40}$$

and $Pr(E)$ is given by equation (2.23) as

$$Pr(E) = \frac{P_e}{P_c + P_e}, \tag{2.41}$$

where P_c and P_e are given by equations (2.25) and (2.27), respectively.

(3) For the type-I hybrid-ARQ protocol, assume that the same BCH code is used for error correction and error detection. The decoder can correct up to t' errors and detect $t'+1$ or more errors, where t' is less than or equal to t. Based on the formulas derived in [91], μ can be found as

$$\mu = \frac{k}{n}(P_c + P_{CD} + P_{ID}), \tag{2.42}$$

and

$$Pr(E) = \frac{P_{ID}}{P_c + P_{CD} + P_{ID}}, \tag{2.43}$$

where P_c is given by equation (2.25), P_{CD}, the probability of correct decoding, is given by

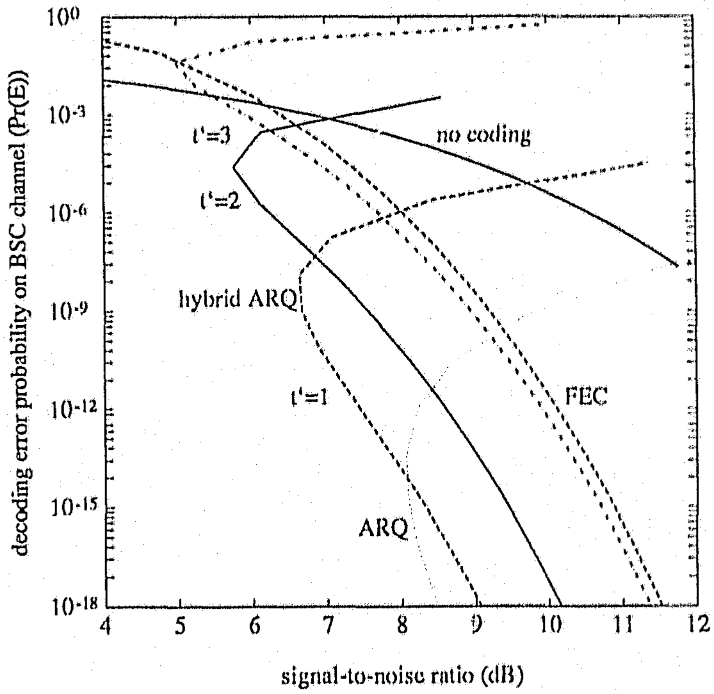

Figure 2.7: Coding gain analysis of various coding schemes

$$P_{CD} = \sum_{i=1}^{t'} \binom{n}{i} p_b^i (1-p_b)^{n-i}, \tag{2.44}$$

and P_{ICD}, the probability of incorrect decoding, is given by

$$P_{ICD} \approx \sum_{i=2t+1-t'}^{2t+1} \binom{n}{i} p_b^i (1-p_b)^{n-i} \frac{\sum_{l=2t+1-i}^{t'} \binom{n}{l}}{2^{n-k} - \sum_{j=0}^{2t-i} \binom{n}{j}}. \tag{2.45}$$

Based on the above formulas, we have plotted E_b/N_0 against $Pr(E)$ curves as shown in Figure 2.7. Clearly, the type-I hybrid-ARQ protocol, when t' is set to zero, is identical to the plain ARQ. On the other hand, if the code has no domain left for error detection, it is equivalent to FEC. Moreover, from the left envelope of all the curves, it is implied that the potential coding gain of an adaptive error control coding system could be significant.

2.6 Summary

In this chapter we have examined the role of feedback in communication systems. There are two fundamental conclusions: (1) Feedback can not improve the capacity of memoryless channels, but it can reduce the coding complexity necessary to achieve a particular standard of reliability. (2) Feedback could increase the capacity of channels with memory.

Our preliminary study on performance evaluation methods suggests that an adaptive coding scheme with limited feedback could achieve significant throughput and coding gain improvement over non-adaptive coding schemes.

Chapter 3

Imbedded Markov Chain Analysis of Delay and Packet Loss

3.1 Introduction

In this chapter we are concerned with mathematical methods for analyzing the delay and the packet loss probability of a hybrid error control system. Delay and packet loss are two important performance characteristics in practical mobile data system design. Delay determines the average time required to deliver a message from origin to destination. Delay considerations strongly influence the choice and performance of network algorithms, such as routing and flow control [97]. It is desirable to have relatively constant delay. Delay analysis also enables the protocol designer to estimate the necessary buffer capacity, in order to avoid packet loss due to buffer overflow.

Queueing theory plays a key role in the quantitative understanding of queueing aspects of ARQ and hybrid-ARQ protocols. In the literature Towsley and Wolf [106] first analyzed the delay and queue length of the stop-and-wait and the go-back-N protocols under the assumption that the feedback channel is noiseless. Later, Konheim [60] used generating functions to obtain an algorithmic solution for the queueing delay, and Saeki and Rubin [92] proposed upper and lower bounds on the expected block delay. More recently, delay and queue length of the selective-repeat ARQ protocol were analyzed [2] and a computationally efficient approximation method was proposed. However, in all of these publications the system is modeled as a discrete time queue with infinite buffer storage. Packet loss due to overflow of a practical finite buffer has not been investigated. Recently, a basic methodology similar to the analysis of the M/G/1 queue has been used to find the queueing delay for type-I hybrid-ARQ protocols [119]. However, delay analysis has not been fully explored for various hybrid-ARQ protocols with a noisy feedback channel.

In this chapter, we consider a hybrid-ARQ protocol using an RS code for both error detection and error correction. Such a protocol has recently been proposed for the land mobile radio channel [113]. Our emphasis is on the queueing analysis of the hybrid-ARQ protocol to obtain analytical solutions for average buffer occupancy, average block delay and probability of packet loss. We define *buffer occupancy* as the number of data blocks in a buffer, *block delay* as the interval between the time a data block enters a buffer and the time it is removed from the buffer, and *packet loss probability* as the probability of overflow due to a finite buffer. Two different queueing analysis methods are presented for infinite and finite transmitter buffers, respectively. In both cases, the system is modeled as an imbedded Markov chain.

In Section 3.2, the hybrid-ARQ protocol is described and the associated Markov chain model is introduced as the basis for our analysis. In Section 3.3, the average transmitter buffer occupancy and the block delay with an infinite buffer are derived. In Section 3.4, the average transmitter buffer occupancy and the probability of packet loss with a finite buffer are analyzed using a different method. Both methods are also found to yield the same analytical results for the infinite buffer case. In Section 3.5, numerical results are presented and compared with the results obtained with a pure ARQ protocol.

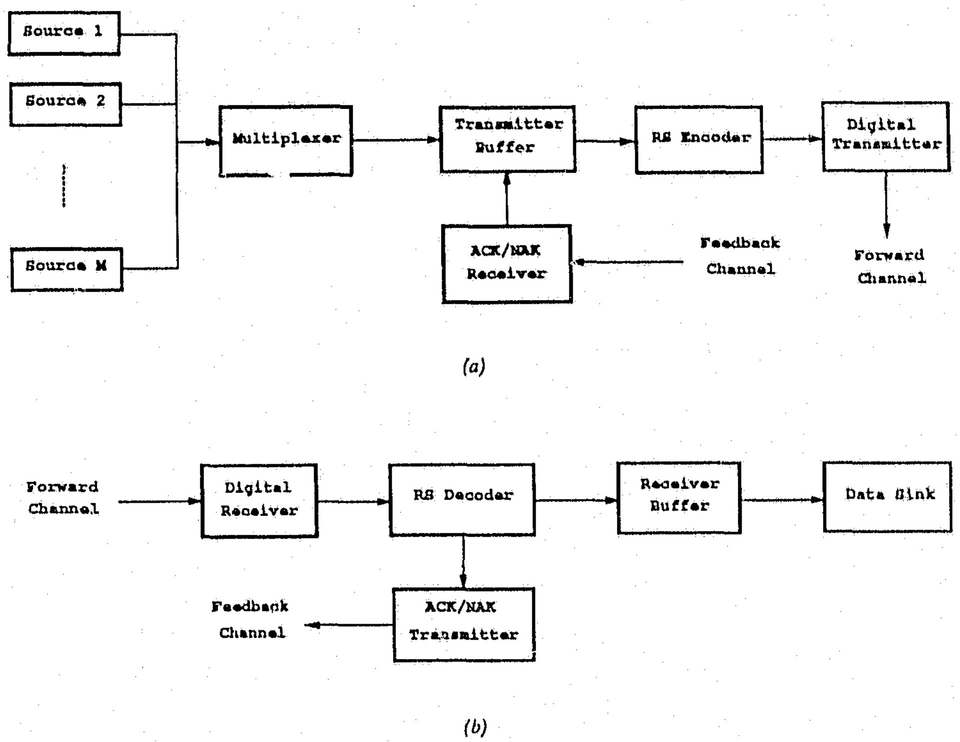

Figure 3.1: ATDM system with hybrid-ARQ error control using a Reed-Solomon code

3.2 The System and Its Model

Reduced to its most basic structure, a digital communications network consists of communication channels (links) and communication processors (nodes). Messages pass over the channels, controlled by the processors. The processors encode and decode messages, route messages over different channels, and otherwise control the traffic on the network. As the messages move from processor to processor, queues form at the different processors as messages arrive and wait for service. Various ARQ and hybrid-ARQ systems can be viewed as simple queueing networks formed by two-way links and data-link protocols. Improper design of coding schemes may lead to excessive queueing delay and packet loss.

The system under consideration is illustrated by the block diagram shown in Figure 3.1. Data blocks from different sources arrive at the transmitter buffer through an asynchronous time division multiplexer (ATDM). The transmitter buffer contains the newly arrived data blocks queued for transmission (*transmission queue*) and the transmitted data blocks waiting for acknowledgment messages (*acknowledgment queue*). The data block at the head of the transmission queue is encoded using an RS code and then transmitted. At the receiver, the received block is decoded using an RS decoder. If the received block contains no errors or if the decoder is unable to detect an error, the receiver sends an acknowledgment (ACK) message to the transmitter. If the RS decoder detects the presence of a block error but is unable to correct it, the receiver sends a negative-acknowledgment (NAK) message to the transmitter. Block retransmissions are based on an ARQ protocol such as stop-and-wait (SAW), go-back-N (GBN), or selective-repeat (SR).

Upon receiving an ACK message, the corresponding block in the acknowledgment queue is removed from the transmitter buffer. Upon receiving an NAK message, the transmitter buffer will function differently depending on the protocol used. For SAW ARQ, the NAKed block rejoins the transmission queue (at the queue head) to be retransmitted. For GBN ARQ, the NAKed block and all succeeding blocks in the acknowledgment queue rejoin the transmission queue (at the queue head) to be retransmitted. In SR ARQ, the most efficient of the three ARQ protocols, only the NAKed block reenters the transmission queue to be retransmitted. However, the receiver must contain storage to save post-NAK blocks until the error block is retransmitted successfully, and the logic for reinserting the block in the proper sequence. The transmitter for SR ARQ will also require more complex logic to be able to send blocks out of sequence. In this chapter, we consider the SAW ARQ and the GBN ARQ protocols. For these two protocols, an imbedded Markov chain model can be developed.

For the sake of illustration, suppose that there are 7 data blocks in the transmitter buffer. Suppose also that the first block is successfully transmitted and the second block is detected in error. Figure 3.2 and Figure 3.3 show the block

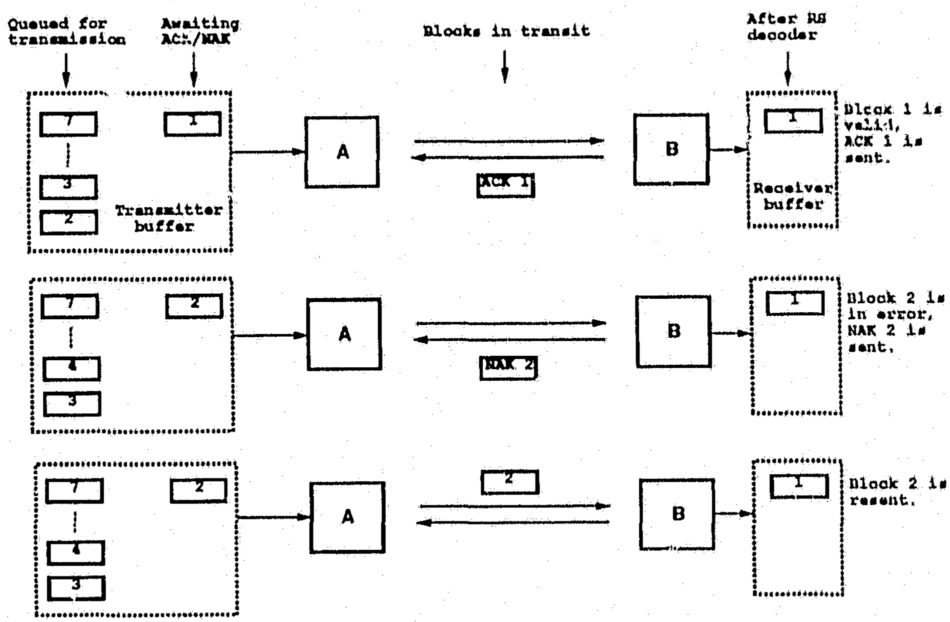

Figure 3.2: Data transmission based on SAW hybrid-ARQ error control

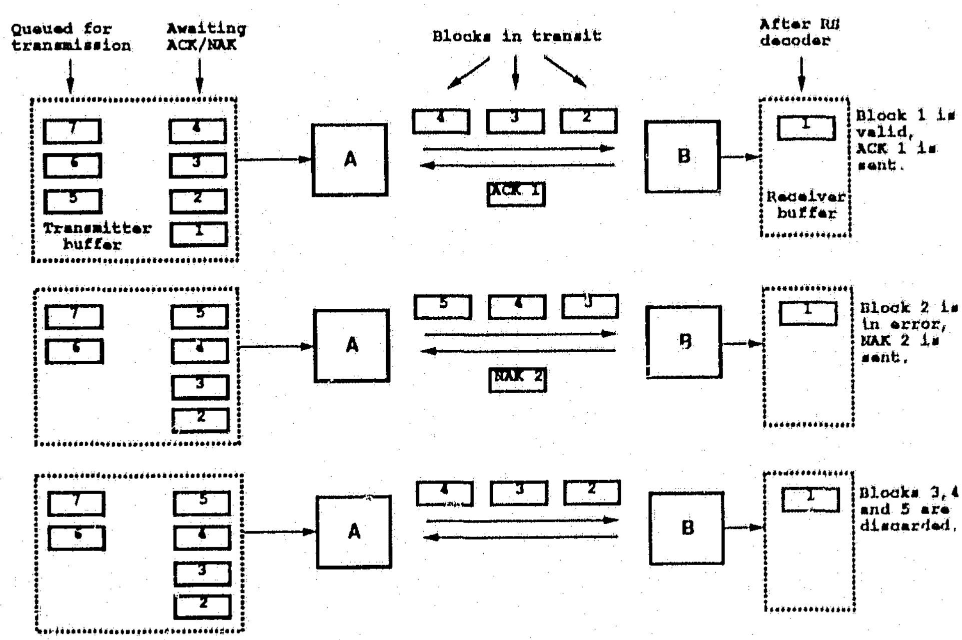

Figure 3.3: Data transmission based on GBN hybrid-ARQ error control

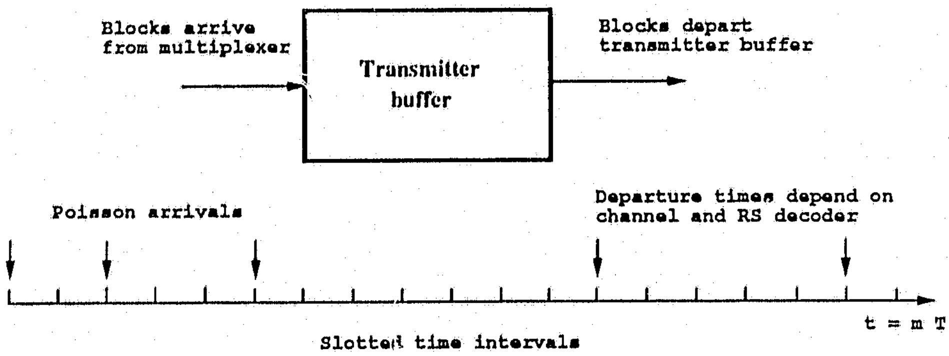

Figure 3.4: The M/G/1 Markov chain model

flows on a two way channel using the SAW hybrid-ARQ protocol and the GBN hybrid-ARQ protocol, respectively. The left column and the right column of the transmitter buffer correspond to the transmission queue and the acknowledgment queue, respectively. For SAW ARQ, only one transmitted block waits for acknowledgment at a time. In GBN ARQ, data blocks are transmitted continuously; the acknowledgment queue must hold many transmitted blocks pending acknowledgment.

In order to investigate the problem of delay and packet loss in the system with a finite transmission buffer, we will focus on the transmitter buffer. The analysis is based on the single server *M/G/1 queueing model* [59] [45] shown in Figure 3.4, where data blocks arrive according to a Poisson (Markov) process with rate λ and the time interval during which a block resides in the transmitter buffer has an arbitrary (General) distribution.

A *renewal point* is a time epoch at which the Markov property holds; that is, if t is a renewal point for a given process, then the future evolution of the process depends only on the state at time t. In a Markov process, every point t is a renewal point. A *Markov chain* is the discrete analogue of a Markov process, in which the Markov property holds only at discrete values of the parameter t, that is at a discrete set of renewal points.

With regard to the M/G/1 queueing model, the data block departure points, which are those epochs at which data blocks are successfully transmitted and leave the transmitter buffer, are renewal points. For whenever a data block leaves the buffer, the transition probabilities (and therefore the future evolution of the system) depend only on the number of data blocks in the buffer at the departure point of a data block. The system states at this discrete set of renewal points are termed an *imbedded Markov chain.*

The imbedded Markov chain, introduced by Kendall [55] [56], is a powerful method for the analysis of certain queueing models, including the M/G/1 model. The theory of Markov chains is extensive. Most existing queueing analyses of ARQ systems considered the transmission queue only and applied the formula directly from queueing theory. Since we consider the transmitter buffer as a whole, including both the transmission queue and the acknowledgment queue, we derive the results starting from basic assumptions. However, we take a direct approach to the analysis of the system, without extended excursions into the surrounding theoretical structure. Thus, we focus on the main ideas behind the theory of the imbedded Markov chain, and show how these ideas facilitate the analysis of delay and packet loss for ATDM with hybrid-ARQ error control. We restrict our attention mainly to equilibrium queues in which buffered data blocks wait until acknowledged, and in which transmission takes place in order of arrival.

In order to carry out a detailed analysis, we make the following two basic assumptions:

- transmissions are synchronized to the occurrence of time slots, where a slot (T seconds) carries exactly one message block.
- errors on successive transmissions are independent events.

3.3 Delay with an Infinite Buffer

The transmitter buffer input process is modeled as follows. Data blocks arrive at epochs T_1, T_2,...,T_k,.... The interarrival times $T_{k+1} - T_k$ ($k = 0, 1, ...; T_0 = 0$)

are identically distributed, mutually independent, positive random variables with distribution function

$$F(x) = P\{T_{k+1} - T_k \leq x\} = 1 - e^{-\lambda x} \tag{3.1}$$

independent of the index k. Consequently, the probability $P_j(t)$ that j arrivals occur in an interval $(T_k, T_k + t]$ has the Poisson distribution

$$P_j(t) = \frac{(\lambda t)^j}{j!} e^{-\lambda t} \quad (j = 0, 1, ...). \tag{3.2}$$

Let us assume first that the transmitter buffer has infinite capacity. We formulate the system equations from the viewpoint of a departing data block. Because of the probability of channel error, the RS decoder may detect errors in the received block and ask for retransmissions. Thus we consider that the imbedded points occur at the time points between data block interdeparture intervals.

Let N_i be the number of data blocks in the transmitter buffer (including any blocks waiting for acknowledgment) just after the departure epoch of the ith data block, and let a_i be the number of data blocks that arrive during the time period when the ith data block stays in the transmitter buffer. The random variables $a_1, a_2, ...$ are mutually independent and identically distributed, and a_i is independent of $N_1, N_2, ..., N_{i-1}$.

If the ith departing data block does not leave the transmitter buffer empty, then the $(i+1)$th departure will leave behind those same data blocks in the buffer left by the ith departure except for itself, plus all those data blocks which arrived during the staying interval of the $(i+1)$th data block. Thus

$$N_{i+1} = N_i - 1 + a_{i+1}, \quad \text{for } N_i > 0. \tag{3.3}$$

On the other hand, if the ith data block leaves the buffer empty, then the $(i+1)$th data block will leave behind just those data blocks which arrived during its staying time. Thus

$$N_{i+1} = a_{i+1}, \quad \text{for } N_i = 0. \tag{3.4}$$

Equations (3.3) and (3.4) can be combined into a single equation

$$N_{i+1} = N_i - U(N_i) + a_{i+1}, \tag{3.5}$$

where the random variable $U(N_i)$ is defined as

$$U(N_i) = \begin{cases} 0 & \text{if } N_i = 0 \\ 1 & \text{if } N_i > 0. \end{cases} \tag{3.6}$$

We now square both sides of (3.5):

$$N_{i+1}^2 = N_i^2 + U^2(N_i) + a_{i+1}^2 + 2N_i a_{i+1} - 2a_{i+1}U(N_i) - 2N_i U(N_i). \tag{3.7}$$

It follows from the definition (3.6) that

$$U^2(N_i) = U(N_i) \tag{3.8}$$

and

$$N_i U(N_i) = N_i, \tag{3.9}$$

so that equation (3.7) can be written as

$$N_{i+1}^2 = N_i^2 + U(N_i) + a_{i+1}^2 + 2N_i a_{i+1} - 2a_{i+1}U(N_i) - 2N_i. \tag{3.10}$$

We now take expected values of the terms in (3.10). Since N_i and a_{i+1} are independent, we have

$$\begin{aligned} E(N_{i+1}^2) &= E(N_i^2) + E(U(N_i)) + E(a_{i+1}^2) + 2E(N_i)E(a_{i+1}) \\ &\quad -2E(a_{i+1})E(U(N_i)) - 2E(N_i). \end{aligned} \tag{3.11}$$

Now let $i \to \infty$ in (3.11). Assuming that a statistical equilibrium distribution exists, then in equilibrium the distribution of the number of data blocks left behind by each departing data block is the same. We have

$$\lim_{i\to\infty} E(N_{i+1}^2) = \lim_{i\to\infty} E(N_i^2) = E(N^2). \tag{3.12}$$

Similarly, taking the expected values of the terms in (3.5) in the limit, we have

$$E(U(N)) = E(a). \tag{3.13}$$

Therefore, from (3.11) we obtain the average buffer occupancy

$$\begin{aligned} E(N) &= \frac{E(a)[1-2E(a)] + E(a^2)}{2[1-E(a)]} \\ &= E(a) + \frac{E(a^2) - E(a)}{2[1-E(a)]}. \end{aligned} \tag{3.14}$$

Note that $E(a)$ is called the *system load*, often denoted by ρ in queueing theory, which is simply the average number of arrivals occurring during an arbitrary interdeparture interval. From equation (3.13) we have

$$E(a) = E(U(N)) = P\{\text{the buffer is not empty}\} \leq 1. \tag{3.15}$$

Furthermore, from (3.14), for a steady-state solution to exist we must have $E(a) < 1$.

It remains to calculate $E(a)$ and $E(a^2)$. In a mobile fading environment, the feedback channel cannot be assumed error-free. When an ACK or NAK is missed the previous transmission must be repeated. For a SAW or GBN hybrid-ARQ protocol, a missed NAK has no detrimental effect because by setting a proper time-out, a default retransmission request can be invoked. The missed ACK, however, causes an extra transmission. If α is the ACK failure probability, the average number of transmissions increases by a factor of $\alpha/(1-\alpha)$ [26].

Consider first that a SAW hybrid-ARQ error control protocol is used. When a decoded data block is detected in error after the RS decoder, the erroneous block is retransmitted until the block is declared to be error free. The transmitted data block remains in the transmitter buffer until the ACK arrives. Let R denote the round-trip delay in slots. If there are k transmissions, the interdeparture interval will be $(k + \frac{\alpha}{1-\alpha})(R+1)T$. Since data blocks arrive at a Poisson rate of λ blocks per second, the probability of l blocks arriving in the interdeparture interval is

$$P(l,\ k) = \frac{[\lambda(k + \frac{\alpha}{1-\alpha})(R+1)T]^l}{l!}\, e^{-\lambda(k+\frac{\alpha}{1-\alpha})(R+1)T}, \tag{3.16}$$

where $l = 0, 1, \ldots$. The probability generating function for $P(l,\ k)$ is

$$G_k(z) \equiv \sum_{l=0}^{\infty} z^l P(l,\ k) = e^{-\lambda(k+\frac{\alpha}{1-\alpha})(R+1)T(1-z)}. \tag{3.17}$$

Note that the probability of the event that a retransmission occurs, denoted by P_E, is the probability of block retransmission. This probability is determined in Section 3.5. Now averaging (3.17) over k, the generating function for the number of data blocks which arrive in the interdeparture interval is given by

$$\begin{aligned} A(z) &= \sum_{k=1}^{\infty} G_k(z)(1-P_E)P_E^{k-1} \\ &= \frac{(1-P_E)e^{-\lambda\frac{R+1}{1-\alpha}T(1-z)}}{1-P_E e^{-\lambda(R+1)T(1-z)}}. \end{aligned} \tag{3.18}$$

By differentiating $A(z)$ with respect to z and then setting $z = 1$, we have

$$A'(1) = \lambda\frac{R+1}{1-\alpha}T + \frac{P_E}{1-P_E}\,\lambda(R+1)T, \tag{3.19}$$

$$A''(1) = \left(\lambda\frac{R+1}{1-\alpha}T\right)^2 + \frac{2P_E}{1-P_E}\cdot\frac{[\lambda(R+1)T]^2}{1-\alpha} + \frac{P_E(1+P_E)}{(1-P_E)^2}\,[\lambda(R+1)T]^2 \tag{3.20}$$

Note that $E(a) = A'(1)$ and $E(a^2) = A''(1) + A'(1)$. Thus, by equation (3.14), the average buffer occupancy for a SAW hybrid-ARQ protocol, denoted by N_{saw}, is

$$\begin{aligned} N_{saw} &= \lambda\frac{R+1}{1-\alpha}T + \frac{P_E}{1-P_E}\,\lambda(R+1)T \\ &+ \frac{(\lambda\frac{R+1}{1-\alpha}T)^2 + \frac{2P_E}{1-P_E}\frac{[\lambda(R+1)T]^2}{1-\alpha} + \frac{P_E(1+P_E)}{(1-P_E)^2}\,[\lambda(R+1)T]^2}{2[1-\lambda\frac{R+1}{1-\alpha}T - \frac{P_E}{1-P_E}\,\lambda(R+1)T]}. \end{aligned} \tag{3.21}$$

Consider next the GBN hybrid-ARQ error control protocol. Data blocks are transmitted in every slot. The transmitter is obliged to buffer previously transmitted blocks, until the corresponding ACKs arrive. When a decoded data block is detected to be in error by the RS decoder, the erroneous block as well as all succeeding data blocks waiting for ACK/NAK messages are retransmitted. The difference between GBN and SAW lies in the interdeparture interval that is seen by the transmitter buffer. In the absence of error the interdeparture interval is a single slot rather than a slot plus a round-trip delay as in the SAW case. If there are k transmissions, the interdeparture interval will be $T + [(k-1+\frac{\alpha}{1-\alpha})(R+1)]T$. Similar to the previous case, we find

$$P(l,\ k) = \frac{\{\lambda[1+(k-1+\frac{\alpha}{1-\alpha})(R+1)]T\}^l}{l!}\,e^{-\lambda[1+(k-1+\frac{\alpha}{1-\alpha})(R+1)]T}, \tag{3.22}$$

where $l = 0, 1, \ldots,$

$$G_k(z) = e^{-\lambda[1+(k-1+\frac{\alpha}{1-\alpha})(R+1)]T(1-z)}, \tag{3.23}$$

and

$$A(z) = \frac{(1-P_E)e^{-\lambda[1+\frac{\alpha}{1-\alpha}(R+1)]T(1-z)}}{1-P_E e^{-\lambda(R+1)T(1-z)}}. \tag{3.24}$$

Again, by differentiating $A(z)$ with respect to z and then setting $z = 1$, we have

$$A'(1) = \lambda[1 + \frac{\alpha}{1-\alpha}(R+1)]T + \frac{P_E}{1-P_E}\lambda(R+1)T, \tag{3.25}$$

$$\begin{aligned} A''(1) &= \left\{\lambda\left[1 + \frac{\alpha}{1-\alpha}(R+1)\right]T\right\}^2 \\ &+ \frac{2P_E}{1-P_E}(\lambda T)^2\left[1 + \frac{\alpha}{1-\alpha}(R+1)\right](R+1) + \frac{P_E(1+P_E)}{(1-P_E)^2}[\lambda(R+1)T]^2. \end{aligned} \tag{3.26}$$

Thus, again by equation (3.14), the average buffer occupancy for a GBN hybrid-ARQ protocol, denoted by N_{gbn}, is

$$\begin{aligned} N_{gbn} &= \lambda\left[1 + \frac{\alpha}{1-\alpha}(R+1)\right]T + \frac{P_E}{1-P_E}\lambda(R+1)T \\ &+ (\lambda T)^2 \frac{[1 + \frac{\alpha}{1-\alpha}(R+1)]^2 + \frac{2P_E}{1-P_E}[1 + \frac{\alpha}{1-\alpha}(R+1)](R+1) + \frac{P_E(1+P_E)}{(1-P_E)^2}(R+1)^2}{2\{1 - \lambda[1 + \frac{\alpha}{1-\alpha}(R+1)]T - \frac{P_E}{1-P_E}\lambda(R+1)T\}}. \end{aligned} \tag{3.27}$$

Based on these results, we can obtain the block delay for both SAW and GBN hybrid-ARQ protocols. Consider a long time interval $(0, mT)$ throughout which statistical equilibrium prevails. The average number of data blocks which enter the transmitter buffer during this interval is $\lambda(mT)$. Suppose that each arriving data block has associated with it a staying time. If the average staying time is W, then the average amount of staying time brought into the transmitter buffer during $(0, mT)$ is $\lambda(mT)W$. On the other hand, if L is the average number of data blocks present in the transmitter buffer throughout $(0, mT)$, then $L(mT)$ is the average amount of staying time used up in $(0, mT)$. If the system is in statistical equilibrium and the limiting values L and W exist, then as $m \to \infty$ the accumulation of staying time must equal the amount of staying time used up. Hence

$$\lim_{m\to\infty} \lambda(mT)W = \lim_{m\to\infty} L(mT), \tag{3.28}$$

from which we have

$$L = \lambda W. \tag{3.29}$$

This relationship is a basic result of queueing theory and is known as *Little's formula* [70]. It is stated in many ways, but its meaning is straightforward and intuitive in our context: $1/\lambda$ is the average time between the arrivals of two consecutive data blocks, L is the average number of data blocks in the system, and W is the average time spent by a data block in the system. Note that the definition of *system* in the original statement of Little's formula is somewhat flexible. In particular, we may denote by L the average length of the queue waiting for transmission and by W the average waiting time for transmission, or by L the average number of data blocks in the transmitter buffer (in either transmission queue or acknowledgment queue) and by W the average time spent in the transmitter buffer.

From Little's formula the average block delay for the SAW hybrid-ARQ protocol is given by

$$D_{saw} = \frac{N_{saw}}{\lambda}, \tag{3.30}$$

and the average block delay for the GBN hybrid-ARQ protocol is given by

$$D_{gbn} = \frac{N_{gbn}}{\lambda}. \tag{3.31}$$

3.4 Packet Loss with a Finite buffer

The analysis in the preceding section is based on the assumption of an infinite transmitter buffer. In many practical applications one would expect that the buffer size, although finite, would be large enough so that the probability of buffer overflow would be negligible. However, this may not be the case for mobile fading channels where poor channel conditions may cause frequent retransmissions, increasing the probability of buffer overflow. The buffer size is critical because data blocks arriving at a full transmitter buffer are lost. Consequently the reliability of data transmission not only depends on the undetected error probability of the RS decoder but also on the size of the transmitter buffer.

When the probability of buffer overflow is not negligible, the imbedded Markov chain analysis based on equation (3.5) is not applicable. Recall that we have assumed that the number of data blocks that arrive in an interdeparture interval is independent of the number already in the transmitter buffer. This assumption is violated when the buffer is full since there can be no arrivals. In the following, we analyze the system with a finite buffer by using another imbedded Markov chain technique, introduced also by Kendall [55] [56], based on state transition probabilities.

Again let N_k be the number of data blocks in the transmitter buffer (including any blocks waiting for ACK/NAK messages, but excluding the departing data block) at the instant the kth data block is acknowledged (successfully transmitted). Let M be the maximum number of data blocks the transmitter buffer can store. Then, by the law of total probability

$$\begin{aligned} P\{N_{k+1}=j\} &= \sum_{i=0}^{M} P\{N_{k+1}=j|N_k=i\}P\{N_k=i\}, \\ & \qquad j=0,1,...,M;\, k=1,2,.... \end{aligned} \tag{3.32}$$

Let A_j be the probability of j data blocks arriving during an interdeparture interval. Consider first the transition probability $P\{N_{k+1}=j|N_k=0\}$. If the kth data block leaves the buffer empty, then the $(k+1)$th data block necessarily arrives at an empty buffer. Hence data blocks remaining after the departure of this data block are all those which have arrived during its interdeparture interval. Since data blocks arriving at a full buffer are lost, we have

$$P\{N_{k+1} = j | N_k = 0\} = \begin{cases} A_j, & 0 \le j < M \\ \sum_{l=M}^{\infty} A_l, & j = M. \end{cases} \tag{3.33}$$

Similarly, when $N_k = i > 0$, the $(k+1)$th data block will leave behind the data blocks left behind by the kth data block, except for itself, plus all those data blocks which have arrived during the interdeparture interval of the $(k+1)$th data block. Therefore

$$P\{N_{k+1} = j | N_k = i\} = \begin{cases} A_{j-i+1}, & i > 0; \ \ i-1 \le j < M \\ \sum_{l=M}^{\infty} A_{l-i+1}, & i > 0; \ \ i-1 \le j = M. \end{cases} \tag{3.34}$$

During an interdeparture interval, since at most one data block leaves the buffer, we must have

$$P\{N_{k+1} = j | N_k = i\} = 0, \quad i > 0; \ \ j < i-1. \tag{3.35}$$

Note that the transition probabilities are independent of the value of the index k, that is, they are the same for every pair of successive data blocks. With the transition probabilities thus specified, we can calculate $P\{N_{k+1} = j\}$ using equation (3.32), which relates the distribution of the number of data blocks left behind by the $(k+1)$th departing data block to the distribution of the number left behind by the kth departure, for each $k = 1, 2, \ldots$. From a practical point of view, we are rarely interested in this distribution for any particular value of k; rather we wish to know the distribution of the number of data blocks left behind by an arbitrary departing data block.

Again assuming that a statistical equilibrium distribution exists, then in equilibrium (as $k \to \infty$) the dependency on k should vanish. Let L_j be the steady-state probability that there are j data blocks in the transmitter buffer at the beginning of an interdeparture interval, i.e., $L_j = \lim_{k\to\infty} P\{N_k = j\}$. From a mathematical point of view, it can be shown using the theory of Markov chains that this steady-state distribution, independent of the initial conditions, exists if and only if the system load $\rho < 1$ [56]. Note that if $\rho \geq 1$, then $L_j = 0$ for all finite j. The proof of this statement is the only place where the formal theory of Markov chains is relevant; we accept this intuitively plausible statement without proof. We then have

$$L_j = \lim_{k\to\infty} \sum_{i=0}^{M} P\{N_{k+1} = j | N_k = i\} L_i, \quad j = 0, 1, ..., M. \tag{3.36}$$

Substituting (3.33) to (3.35) into (3.36) yields

$$L_j = L_0 A_j + \sum_{i=1}^{j+1} L_i A_{j-i+1}, \quad 0 \leq j < M, \tag{3.37}$$

and

$$L_M = L_0 \sum_{i=M}^{\infty} A_i + \sum_{i=1}^{M} L_i \sum_{l=M}^{\infty} A_{l-i+1}. \tag{3.38}$$

Note that we can also use equation (3.37) to obtain the steady-state probabilities under the assumption of an infinite buffer ($M \to \infty$). This can be easily proved by defining the probability generating functions $P(z) = \sum_{j=0}^{\infty} L_j z^j$ and $A(z) = \sum_{j=0}^{\infty} A_j z^j$. Using equation (3.37) we then have

$$\begin{aligned} P(z) &= \sum_{j=0}^{\infty}(L_0 A_j + \sum_{i=0}^{j} L_{i+1} A_{j-i}) z^j \\ &= L_0 A(z) + (L_1 + L_2 z + L_3 z^2 + \cdots) A(z) \\ &= \left(L_0 + \frac{P(z) - L_0}{z}\right) A(z). \end{aligned} \tag{3.39}$$

Therefore,

$$P(z) = \frac{A(z)(z-1)}{z - A(z)} L_0. \tag{3.40}$$

We know from equation (3.15) that $L_0 = 1 - \rho$. By differentiating $P(z)$ with respect to z and then setting $z = 1$, we obtain equation (3.14) of the previous section.

We now turn our attention to finding the probability of packet loss for the hybrid-ARQ system with a finite transmitter buffer. Recall that data blocks arriving at a full buffer are lost. The steady-state probability L_M that the buffer contains M data blocks includes cases of overflow. In order to find the probability of packet loss, denoted by P_{loss}, we rewrite equation (3.38) as

$$L_M = L_0 A_M + \sum_{i=1}^{M} L_i A_{M-i+1} + L_0 \sum_{i=M+1}^{\infty} A_i + \sum_{i=1}^{M} L_i \sum_{l=M+1}^{\infty} A_{l-i+1}. \tag{3.41}$$

Clearly, the third and the fourth terms on the right side of equation (3.40) constitute the probability of packet loss. Therefore,

$$\begin{aligned} P_{loss} &= L_0 \sum_{i=M+1}^{\infty} A_i + \sum_{i=1}^{M} L_i \sum_{i=M+1}^{\infty} A_{l-i+1} \\ &= L_M - L_0 A_M - \sum_{i=1}^{M} L_i A_{M-i+1}. \end{aligned} \tag{3.42}$$

It remains to calculate L_j, $j = 0, 1, ..., M$ and A_l, $l = 0, 1, ..., M$. From equation (3.37) we have

$$L_{j+1} = \frac{L_j - \sum_{i=1}^{j} L_i A_{j-i+1} - L_0 A_j}{A_0}, \quad j = 0, 1, ..., M-1. \tag{3.43}$$

By using the normalization equation

$$\sum_{j=1}^{M} L_j = 1, \tag{3.44}$$

we can then obtain $L_0, L_1, ..., L_M$ through a simple iteration procedure. That is, we initialize $L'_0 = 1$ and find $L'_1, ..., L'_M$ from equation (3.42). Normalizing, we then have

$$L_j = \frac{L'_j}{\sum_{i=0}^{M} L'_i}, \quad j = 0, 1, ..., M. \tag{3.45}$$

The average number of data blocks in the transmitter buffer, denoted by N_a, can then be found from

$$N_a = \sum_{j=0}^{M} j\, L_j. \tag{3.46}$$

Note that for the system with a finite buffer, Little's formula cannot be applied to find the block delay. Nevertheless, the block delay under the infinite buffer assumption provides an upper bound on block delay when the assumption is not satisfied. Clearly, if the probability of packet loss is very small, this upper bound represents a good approximation of the block delay for the finite buffer case.

In order to find A_l, the probability of l data blocks arriving in an interdeparture interval, consider first the SAW hybrid-ARQ protocol. As already discussed (see equation (3.16)), if data blocks arrive at a Poisson rate of λ blocks per second, the probability of l blocks arriving in an interdeparture interval is

$$P(l,\ k) = \frac{[\lambda(k + \frac{\alpha}{1-\alpha})(R+1)T]^l}{l!}\, e^{-\lambda(k+\frac{\alpha}{1-\alpha})(R+1)T}. \tag{3.47}$$

Averaging over k, the probability of l data blocks arriving in an interdeparture interval

$$
\begin{aligned}
A_l &= \sum_{k=1}^{\infty} P(l,\ k)(1-P_E)P_E^{k-1} \\
&= \sum_{k=1}^{\infty} \frac{[\lambda(k+\frac{\alpha}{1-\alpha})(R+1)T]^l}{l!}\, e^{-\lambda(k+\frac{\alpha}{1-\alpha})(R+1)T}(1-P_E)P_E^{k-1}. \quad (3.48)
\end{aligned}
$$

It is easy to verify that the sum of all A_l $(l = 0, 1, 2, ...\infty)$ equals one. In order to calculate A_l $(l = 0, 1, ..., M)$, let

$$\beta = e^{-\lambda(R+1)T} P_E \qquad (3.49)$$

and

$$f_l(\beta) = \sum_{k=1}^{\infty} (k + \frac{\alpha}{1-\alpha})^l \beta^k. \qquad (3.50)$$

Then we have

$$
\begin{aligned}
A_l &= \frac{[\lambda(R+1)T]^l}{l!} \cdot \frac{1-P_E}{P_E} e^{-\lambda(\frac{\alpha}{1-\alpha})(R+1)T} \sum_{k=1}^{\infty} (k+\frac{\alpha}{1-\alpha})^l [e^{-\lambda(R+1)T}P_E]^k \\
&= \frac{[\lambda(R+1)T]^l}{l!} \cdot \frac{1-P_E}{P_E} e^{-\lambda(\frac{\alpha}{1-\alpha})(R+1)T} \sum_{k=1}^{\infty} (k+\frac{\alpha}{1-\alpha})^l \beta^k \\
&= \frac{[\lambda(R+1)T]^l}{l!} \cdot \frac{1-P_E}{P_E} e^{-\lambda(\frac{\alpha}{1-\alpha})(R+1)T} f_l(\beta), \quad (3.51)
\end{aligned}
$$

where

$$f_0(\beta) = \sum_{k=1}^{\infty} (k + \frac{\alpha}{1-\alpha})^0 \beta^k = \frac{\beta}{1-\beta}, \qquad (3.52)$$

and

$$f_1(\beta) = \sum_{k=1}^{\infty}(k + \frac{\alpha}{1-\alpha})^1 \beta^k = \frac{\beta}{(1-\beta)^2} + \frac{\alpha}{1-\alpha} \cdot \frac{\beta}{1-\beta}. \tag{3.53}$$

For $l = 2, 3, ..., M$, note that

$$\begin{aligned} \frac{df_{l-1}(\beta)}{d\beta} &= \frac{d}{d\beta}[\sum_{k=1}^{\infty}(k + \frac{\alpha}{1-\alpha})^{l-1}\beta^k] \\ &= \sum_{k=1}^{\infty}(k + \frac{\alpha}{1-\alpha})^{l-1} k \beta^{k-1} \\ &= \frac{1}{\beta}\left[f_l(\beta) - \frac{\alpha}{1-\alpha} f_{l-1}(\beta)\right]. \end{aligned} \tag{3.54}$$

Therefore the following simple recursive expression can be used to calculate $f_l(\beta)$:

$$f_l(\beta) = \beta\frac{df_{l-1}(\beta)}{d\beta} + \frac{\alpha}{1-\alpha} f_{l-1}(\beta), \tag{3.55}$$

and thus A_l can then be calculated by using equation (3.51).

Similarly, the above procedure can be used to find the probability of packet loss for the GBN hybrid-ARQ system. The only difference is the interdeparture intervals and thus the values of A_l ($l = 0, 1, ...$). Specifically, for GBN hybrid-ARQ, we have

$$P(l,\ k) = \frac{\{\lambda[1 + (k - 1 + \frac{\alpha}{1-\alpha})(R+1)]T\}^l}{l!} e^{-\lambda[1+(k-1+\frac{\alpha}{1-\alpha})(R+1)]T}. \tag{3.56}$$

Again averaging over k we have for the probability of i data blocks arriving in an interdeparture interval

$$
\begin{aligned}
A_l &= \sum_{k=1}^{\infty} \frac{\{\lambda[1+(k-1+\frac{\alpha}{1-\alpha})(R+1)]T\}^l}{l!} e^{-\lambda[1+(k-1+\frac{\alpha}{1-\alpha})(R+1)]T}(1-P_E)P_E^{k-1} \\
&= \frac{[\lambda(R+1)T]^l}{l!} \cdot \frac{1-P_E}{P_E} e^{-\lambda[1+(\frac{\alpha}{1-\alpha}-1)(R+1)]T} f_l(\beta), \qquad (3.57)
\end{aligned}
$$

where

$$
f_l(\beta) = \sum_{k=1}^{\infty} (k + \frac{1}{R+1} - 1 + \frac{\alpha}{1-\alpha})^l \beta^k, \qquad (3.58)
$$

and β is given by equation (3.48). The recursive expression (3.54) can be easily modified to find A_l for the GBN case. Specifically, we have

$$
f_0(\beta) = \sum_{k=1}^{\infty} (k + \frac{1}{R+1} - 1 + \frac{\alpha}{1-\alpha})^0 \beta^k = \frac{\beta}{1-\beta}, \qquad (3.59)
$$

$$
\begin{aligned}
f_1(\beta) &= \sum_{k=1}^{\infty} (k + \frac{1}{R+1} - 1 + \frac{\alpha}{1-\alpha})^1 \beta^k \\
&= \frac{\beta}{(1-\beta)^2} + \left(\frac{1}{R+1} - 1 + \frac{\alpha}{1-\alpha}\right) \frac{\beta}{1-\beta}, \qquad (3.60)
\end{aligned}
$$

and

$$
f_l(\beta) = \beta \frac{df_{l-1}(\beta)}{d\beta} + \left(\frac{1}{R+1} - 1 + \frac{\alpha}{1-\alpha}\right) f_{l-1}(\beta). \qquad (3.61)
$$

3.5 Results

In this section some numerical examples concerning the block delay and the probability of packet loss for ATDM with hybrid-ARQ error control are presented. In particular, the (31,15) 8-error correcting RS code and the (255,223) 16-error correcting RS code are used in both SAW and GBN hybrid-ARQ protocols. The results will be compared with the results of the SAW and GBN ARQ protocols using the popular CRC-16 and/or CRC-32 error detecting codes.

The RS codes considered here are defined over $GF(2^m)$, where m is the number of bits per symbol. We assume that the m-bit symbols are transmitted sequentially over a binary memoryless channel. The receiver performs symbol regeneration prior to the RS decoding. The memoryless channel assumption can be satisfied for mobile radio channels by bit interleaving over a sufficiently long period. The delay introduced by interleaving will be taken into account as part of processing delay in the calculation.

Suppose that in each transmission, a block of N RS codewords is transmitted. The analytical results derived in the previous sections require the probability of block retransmission P_E to be determined. Let P_{ED} be the probability of error detection by an RS code; the probability of block retransmission is then given by

$$P_E = 1 - (1 - P_{ED})^N. \tag{3.62}$$

A bounded distance decoder based on the Berlekamp-Messey algorithm is normally used to implement the RS decoder [81]. The decoder selects as the transmitted codeword any codeword that differs from the received word in t or fewer symbols, where t is the error correction capability of an RS code. If no such codeword exists, a codeword error is detected. If the received word is within distance t of an incorrect codeword, a decoding error is committed. Let P_{CD} and P_{ICD} be the probabilities of correctly decoding and incorrectly decoding into another valid codeword, respectively. We have

$$P_{ED} = 1 - P_{CD} - P_{ICD}. \tag{3.63}$$

For the bounded distance RS decoder which corrects up to t symbol errors, the probability of correct decoding can be easily calculated by

$$
\begin{aligned}
P_{CD} &= \sum_{k=0}^{t} \binom{n}{k} p_s^k (1-p_s)^{n-k} \\
&= \sum_{k=0}^{t} \binom{n}{k} [1-(1-p_b)^m]^k [(1-p_b)^m]^{n-k}, \qquad (3.64)
\end{aligned}
$$

where p_s and p_b are the channel symbol error probability and the channel bit error probability, respectively.

For the probability of incorrect decoding P_{ICD}, the computation is much more involved, although an exact formula for P_{ICD} exists [22]. Assume that all error patterns of weight k are equally probable. Let $P_e(k)$ denote the probability of incorrect decoding given that an error pattern of weight k occurs. It has been shown that P_{ICD} is given by [53]

$$
\begin{aligned}
P_{ICD} &= \sum_{k=d-t}^{n} P_e(k) \binom{n}{k} P_s^k (1-P_s)^{n-k} \\
&= \sum_{k=d-t}^{n} P_e(k) \binom{n}{k} [1-(1-p_b)^m]^k [(1-p_b)^m]^{n-k}. \qquad (3.65)
\end{aligned}
$$

As shown in [22], $P_e(k)$ may be expressed as follows:

$$
P_e(k) = \frac{D_k}{\binom{n}{k}(2^m-1)^k}, \quad \text{for } d-t \le k \le n, \qquad (3.66)
$$

where

$$
D_k = \binom{n}{k} \sum_{j=0}^{k-d+t} (-1)^j N_j, \quad \text{for } d-t \le k \le n, \qquad (3.67)
$$

$$N_j = \binom{n}{j}\left[2^{m(k-d+1-j)}\sum_{i=0}^{t}\binom{n}{i}(2^m-1)^i - \sum_{i=0}^{t}\binom{k-j}{i}(2^m-1)^i\right], \quad \text{for } 0 \le j \le k-d, \tag{3.68}$$

and

$$N_j = \binom{n}{j}\left[\sum_{w=d-k+j}^{t}\binom{n-k+j}{w}\sum_{i=0}^{w-d+k-j}(-1)^i\binom{w}{i}\left(2^{m(w-d+k-j-i+1)}-1\right) \cdot \sum_{s=w}^{t}\binom{k-j}{s-w}(2^m-1)^{s-w}\right], \quad \text{for } k-d+1 \le j \le k-d+t. \tag{3.69}$$

For pure ARQ protocols, the probability of block retransmission is

$$P_E = 1-(P_c+P_e), \tag{3.70}$$

where P_c is the probability that a received n-bit block contains no error and P_e is the probability that a received n-bit block contains an undetectable error pattern. For CRC codes, we have

$$P_c = (1-p_b)^n, \tag{3.71}$$

and P_e can be estimated by using the following bound [68],

$$P_e \le 2^{-(n-k)}[1+(1-2p_b)^n - 2(1-p_b)^n], \tag{3.72}$$

for $0 \le p_b \le 0.5$, where $n-k$ is the number of parity bits in a CRC code.

It is known that RS codes are very effective for simultaneous error correction and detection [53]. In particular, we may compare P_{ICD}, the probability of incorrect decoding (undetected error) of RS codes, with P_e, the probability of undetected

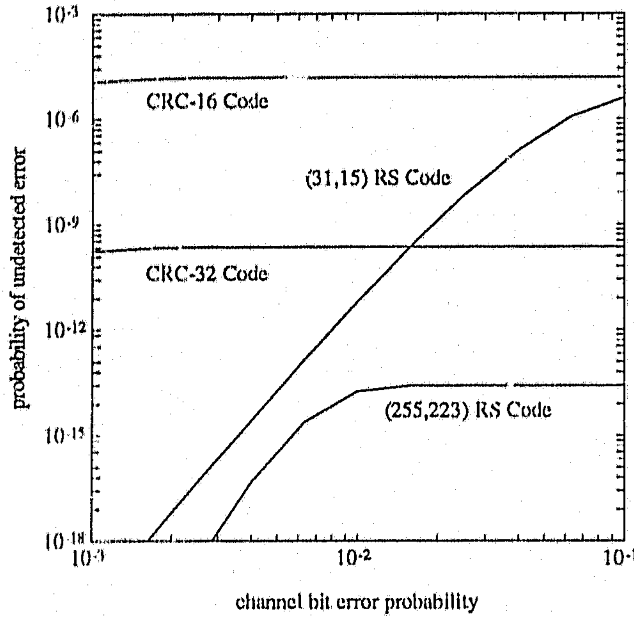

Figure 3.5: Probability of undetected error vs. channel bit error probability, where a block length of 2000 bits is assumed for CRC-16 and CRC-32 codes

error of CRC codes. The comparison results are shown in Figure 3.5 in terms of channel bit error probability. It clearly seen that the (31,15) and (255,223) RS codes can be effectively used for error detection even when their error correction capability has been fully utilized.

In most network applications, particularly those in which bandwidth is at a premium, ACK and NAK messages are piggybacked onto blocks traveling in the reverse direction (feedback channel). For simplicity therefore, the ACK failure probability α is assumed to be the same as the probability of block retransmission P_E. Under this assumption, equation (3.21) can be written as

$$N_{saw} = \frac{1+P_E}{1-P_E}\lambda(R+1)T + \frac{(1+3P_E+P_E^2)[\lambda(R+1)T]^2}{2(1-P_E)^2[1-\frac{1+P_E}{1-P_E}\lambda(R+1)T]}. \qquad (3.73)$$

Similarly, equation (3.27) can be written as

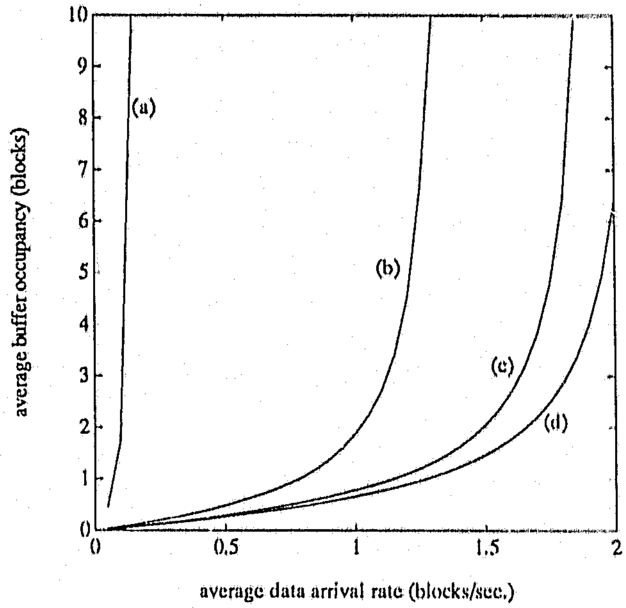

Figure 3.6: Average buffer occupancy for SAW protocol with: (a) CRC-16 code given $p_b = 10^{-3}$; (b) CRC-16 code given $p_b = 10^{-4}$; (c) (255,223) RS code given $p_b = 10^{-3}$; (d) (31,15) RS code given $p_b = 10^{-3}$

$$N_{gbn} = \lambda \left[1 + \frac{2P_E}{1-P_E}(R+1)\right] T + \frac{(\lambda T)^2 \left[1 + \frac{4P_E}{1-P_E}(R+1) + \frac{P_E(1+4P_E)}{(1-P_E)^2}(R+1)^2\right]}{2\{1 - \lambda[1 + \frac{2P_E}{1-P_E}(R+1)]T\}} . \quad (3.74)$$

In obtaining the following numerical results, we have assumed a data transmission speed of 9600 bps, total transmitter/receiver processing delay of 100 ms, and channel propagation delay of 100 ms. In order to compare the performance on the basis of similar block lengths, we also assume that each block carries ten (31,15) RS codewords, or one (255,233) RS codeword, or one CRC codeword of 2000 bits.

The average buffer occupancy of the SAW and GBN hybrid-ARQ error control protocols using (31,15) and (255,233) RS codes is shown in Figure 3.6 and Figure 3.7, respectively. The block delay of these protocols is shown in Figure 3.8 and Figure 3.9. Also shown in these figures for comparison are the results for the plain SAW and GBN ARQ protocols using CRC-16 error detecting code. It is clearly

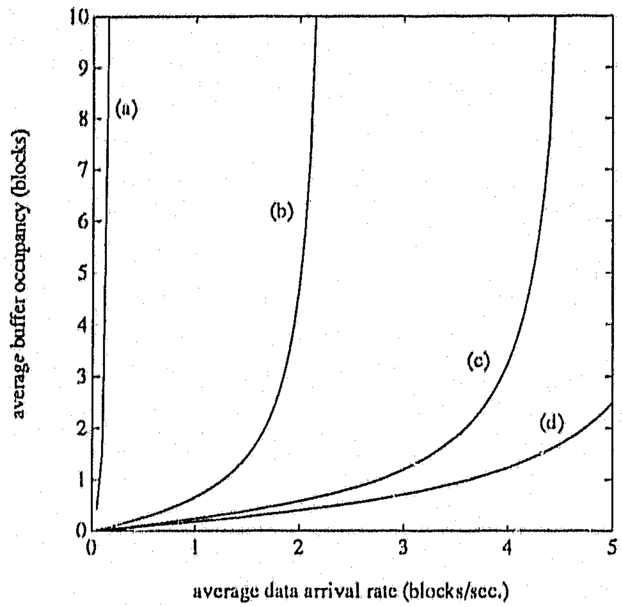

Figure 3.7: Average buffer occupancy for GBN protocol with: (a) CRC-16 code given $p_b = 10^{-3}$; (b) CRC-16 code given $p_b = 10^{-4}$; (c) (255,223) RS code given $p_b = 10^{-3}$; (d) (31,15) RS code given $p_b = 10^{-3}$

shown that the average buffer occupancy, and thus the average block delay, is greatly reduced when the hybrid-ARQ error control protocol is applied, especially when the data arrival rate is high (heavy traffic) or the channel condition is poor (severe fading).

When the transmitter buffer has a finite capacity, the probability of packet loss will be non-zero due to buffer overflow. The probability of packet loss for the SAW and GBN hybrid-ARQ error control protocols using (31,15) and (255,233) RS codes are shown in Figure 3.10 and Figure 3.11. Again, the results for CRC-16 code are also shown for comparison. In Figure 3.10 and Figure 3.11, the buffer is assumed to be able to store 5 blocks. The recursive equations (3.54) and (3.60) have been used to find the values of A_l. The probability of packet loss versus the buffer size are shown in Figure 3.12 and Figure 3.13 for the SAW hybrid-ARQ protocol and the GBN hybrid-ARQ protocol, respectively. For a data arrival rate

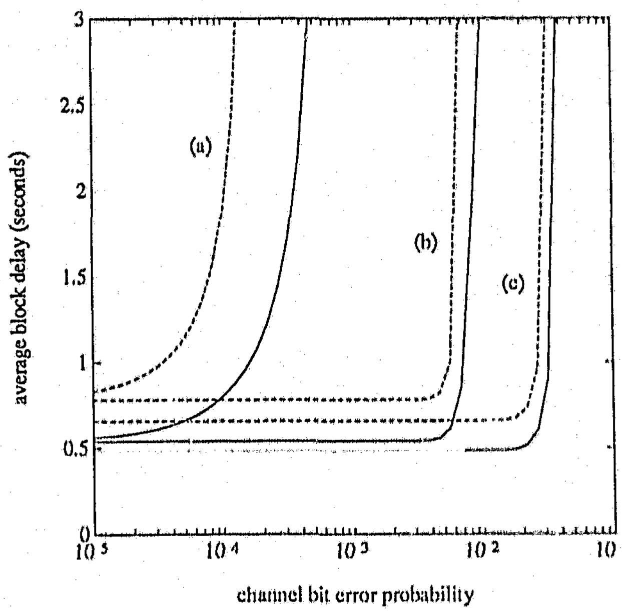

Figure 3.8: Average block delay for SAW protocol with: (a) CRC-16 code; (b) (255,223) RS code; (c) (31,15) RS code, given $\lambda = 0.2$ blocks/sec (solid lines) and $\lambda = 1.0$ blocks/sec (dashed lines)

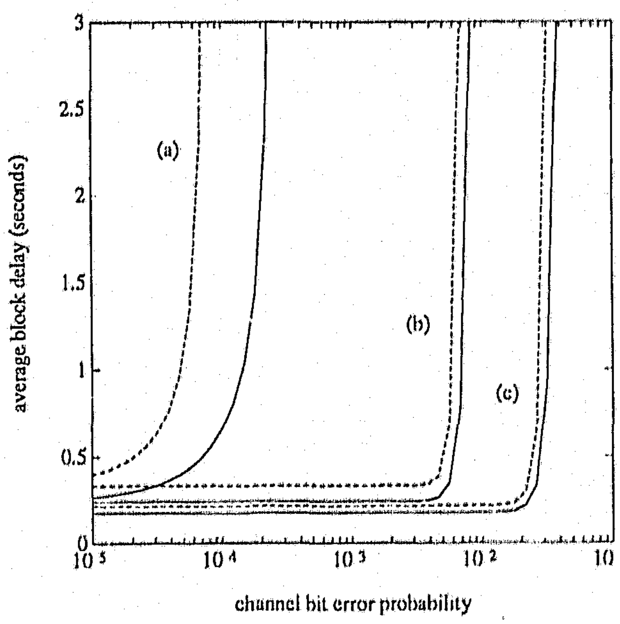

Figure 3.9: Average block delay for GBN protocol with: (a) CRC-16 code; (b) (255,223) RS code; (c) (31,15) RS code, given $\lambda = 1.0$ blocks/sec (solid lines) and $\lambda = 2.5$ blocks/sec (dashed lines)

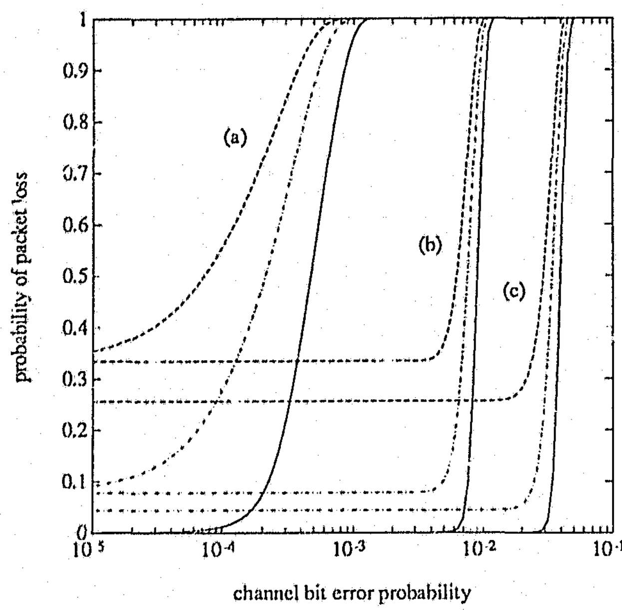

Figure 3.10: Probability of packet loss for SAW protocol with: (a) CRC-16 code; (b) (255,223) RS code; (c) (31,15) RS code, given $\lambda = 1.0$ blocks/sec (solid lines); $\lambda = 2.0$ blocks/sec (dash-dot lines); $\lambda = 3.0$ blocks/sec (dashed lines)

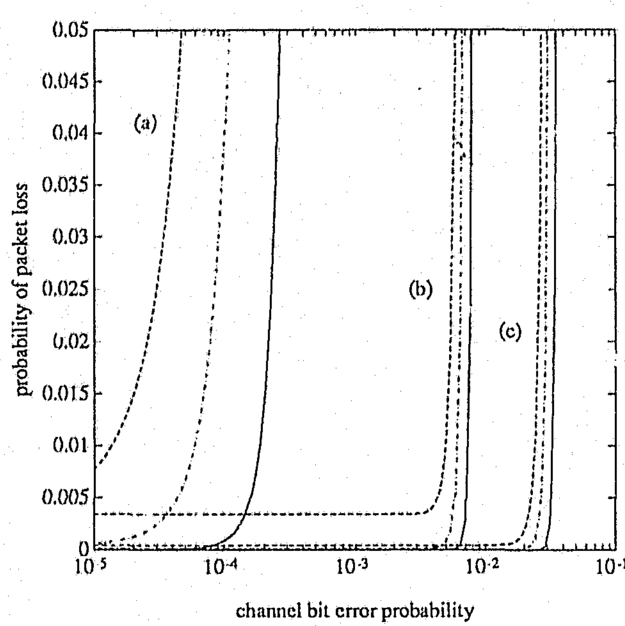

Figure 3.11: Probability of packet loss for GBN protocol with: (a) CRC-16 code; (b) (255,223) RS code; (c) (31,15) RS code, given $\lambda = 1.0$ blocks/sec (solid lines); $\lambda = 2.0$ blocks/sec (dash-dot lines); $\lambda = 3.0$ blocks/sec (dashed lines)

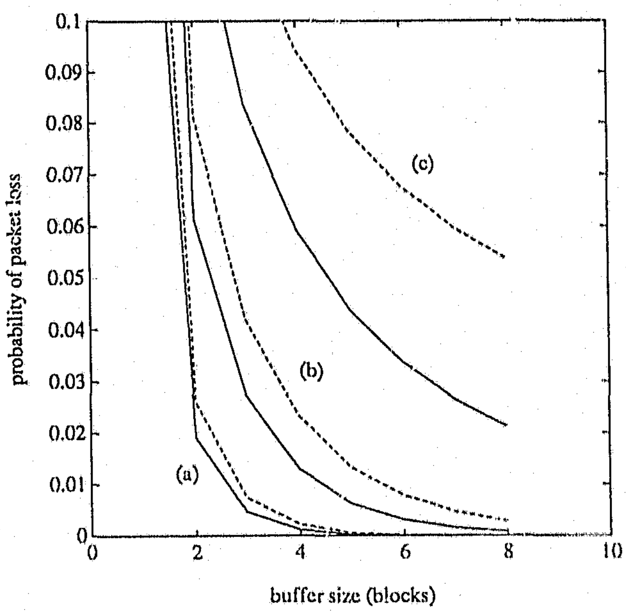

Figure 3.12: Probability of packet loss for SAW protocol with (255,223) RS code (dashed lines) and (31,15) RS code (solid lines), given (a) $\lambda = 1.0$ blocks/sec; (b) $\lambda = 1.5$ blocks/sec; (c) $\lambda = 2.0$ blocks/sec

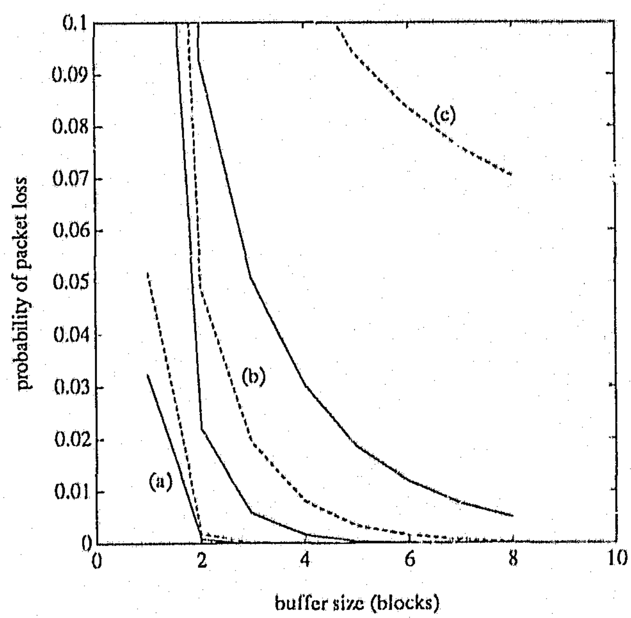

Figure 3.13: Probability of packet loss for GBN protocol with (255,223) RS code (dashed lines) and (31,15) RS code (solid lines), given (a) $\lambda = 1.0$ blocks/sec; (b) $\lambda = 3.0$ blocks/sec; (c) $\lambda = 5.0$ blocks/sec

up to 1.5 blocks/sec (with SAW) and up to 3.0 blocks/sec (with GBN), the results show that a buffer size of 5 to 6 blocks can provide a packet loss probability of less than 1%. Further buffer size increases will not significantly reduce the packet loss probability.

3.6 Summary

The analytical results derived from the imbedded Markov chain system model suggest that the block delay and the probability of packet loss of an ARQ-based protocol in digital communication networks can be greatly reduced by incorporating a Reed-Solomon code into the protocol. Although we have focused on the hybrid-ARQ protocol using Reed-Solomon codes, the procedure shown can also be used to analyze other hybr' ' ARQ error control protocols. Based on the performance analysis of buffer occupancy, block delay, and packet loss, practical design trade-offs can then be made between the buffer size and error correction parameters to achieve maximum performance and minimum complexity.

Chapter 4

Reducing Time Delay by Protocol Truncation

4.1 Introduction

It is reasonable to expect that the delay of a hybrid error control system can be made small enough to provide real-time communications. However, as shown in Chapter 3, system queueing delay could still be unbounded. In this chapter we investigate if and how the delay of a hybrid error control system can be made bounded while retaining the required system reliability.

A major concern in digital mobile communications is the control of transmission errors under channel conditions which fluctuate due to multipath fading, Doppler effects and other interference. Three crucial constraints for real-time mobile communications applications are limited time delay, limited signal power, and limited signal bandwidth. The required reliability is normally not as high as that needed by other systems, such as store-and-forward packet data communications where practically error-free transmission is required. For example, an average BER of less than 10^{-6} for more than 90% of one minute periods is considered satisfactory for voice communications on 64 kbits/sec international connections [43].

For applications involving real-time voice communications and interactive messaging (such as inquiry-based services), ARQ-based protocols impose an unacceptably long transmission delay. On the other hand, a pure FEC strategy must employ a code suitable for worst case channel conditions. As a consequence, the code parameters have non-optimum values during time intervals with lower error probability.

For a mobile fading channel with a nonstationary noise level, it may be desirable to implement an adaptive error control coding protocol. When the channel is quiet, only error detection parity bits are included in each transmission. When the channel becomes noisy, extra error correction parity bits are added. Ideally, the code rate is selected according to the actual channel state. The idea of adaptive incremental redundancy was originated by Mandelbaum [75] [76], and has been considered elsewhere [30] [73]. This concept of adaptivity forms the basis of type-II hybrid-ARQ protocols. The original motivation for using hybrid-ARQ protocols was to provide error-free data transmission. Unfortunately, type-II hybrid-ARQ protocols, like pure ARQ protocols, may still cause unacceptable delay under certain channel conditions.

As discussed in Chapter 3, delay analysis enables the protocol designer to estimate the necessary buffer capacity, in order to avoid block losses. Delay also affects communication cost and impairs real-time dialogues. Clearly, many applications require limited delay. The concept of error control coding with a limited number of retransmissions was first proposed in 1980 by Fujiwara *et al* [39]. The idea of limited retransmissions is also mentioned in a recent paper [1], but the emphasis there is not on reducing the time delay. The topic of type-II hybrid-ARQ protocols with limited retransmissions has been addressed in [73] but the important issues of delay analysis have not been investigated.

In this chapter, we first introduce the concept of delay-limited adaptive error control. A truncated type-II hybrid-ARQ protocol is then investigated in detail. The queueing delay and transmission delay are analyzed based on the imbedded Markov chain method developed in the previous chapter. In particular, queue length, queueing delay and transmission delay of the truncated selective-repeat protocol are analyzed. Numerical results are also obtained to permit comparison of the delay performance of the truncated protocol and other coding methods under various channel conditions. We assume that the feedback channel is error-free and that ideal interleaving is used.

4.2 Delay-limited Adaptive Coding

Delay is the amount of time between when the information is coded and placed in the channel and when it is delivered to the data sink. Obviously, without delay constraint, electrical communication makes no sense. In order to achieve communication reliability over a noisy channel, various ARQ and hybrid-ARQ protocols may require an infinite time delay. On the other hand, FEC schemes use codes of reasonable length to obtain limited delay at the expense of reduced reliability. Delay-limited adaptive coding schemes are special types of hybrid ARQ schemes with a limited number of retransmissions.

Consider adaptive error control coding in which only m retransmissions are allowed, where m is usually a small number, say, two or three. We call this an *mR adaptive coding scheme*. For example, in a 2R adaptive coding scheme, the adaptation is only based on the information provided by at most two retransmissions. Suppose that we use both a CRC code for error detection and a maximum distance separable code for error correction. Assuming that the error correction code has M adaptable code rates, the total number of adaptable states is $M+1$, because no error correction is needed when channel noise does not affect the reliability. A simple adaptation strategy may be that each retransmission request causes a reduction in code rate or a down-shift of the code state, and each ACK signal causes an increase in code rate or an up-shift of the code state. Notice that the number of retransmissions is limited to two, but each retransmission does provide us with useful channel information for the adaptation.

In comparison with FEC, ARQ and various hybrid-ARQ schemes, an mR adaptive coding scheme offers the following advantages:

1. *Increased efficiency* Since most real communication channels are of good quality most of time, an mR adaptive coding scheme stays at the highest code rate most of time so that it is more efficient than a FEC scheme which has a fixed code rate based on a worst channel condition.

2. *Decreased complexity* An mR adaptive coding scheme completely eliminates the buffer overflow problem which might occur in most ARQ or hybrid-ARQ protocols. Since only n retransmissions are allowed, the buffer size can be precisely determined during the design.

3. *Limited delay* Most ARQ and hybrid-ARQ protocols result in a long time delay in a noisy environment, while an mR adaptive coding scheme always has a limited time delay.

4. *Good reliability* By proper design, the error correction code in an mR adaptive coding scheme can be adapted to a very low code rate within m retransmissions to provide high reliability.

Clearly, the performance improvement of an mR adaptive coding scheme is achieved at the expense of a possible increase in complexity compared with ARQ protocols and a possible increase in time delay compared with FEC schemes. The design trade-off between minimizing the coding complexity and maximizing the performance improvement under delay constraints is thus an interesting topic to be explored in this chapter.

In general, a digital communication system using a delay-limited adaptive error control coding scheme can be illustrated by the block diagram shown in Figure 4.1, where m is the maximum number of retransmissions allowed in the protocol.

In particular, retransmissions in a delay-limited protocol can be avoided when real-time voice communication is required. This special case may be called *rate-adaptive forward error correction.* The throughput performance of rate-adaptive FEC, FEC, ARQ, type-I and type-II hybrid-ARQ is compared in Figure 4.2. It is

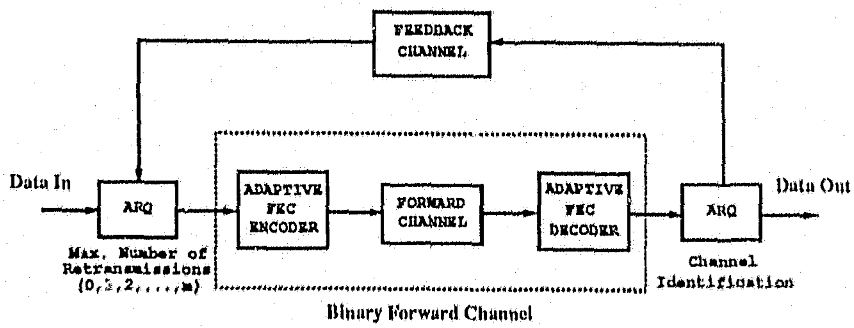

Figure 4.1: Digital communication using delay-limited adaptive error control coding

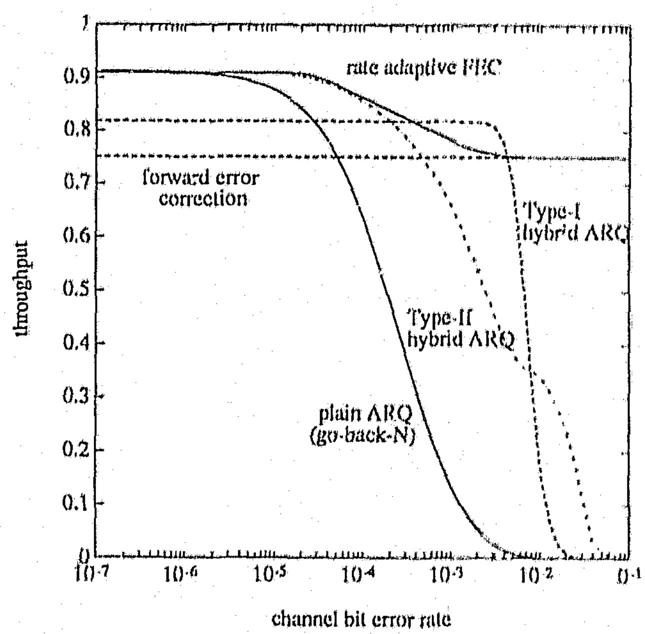

Figure 4.2: Throughput of various error control techniques

shown that rate-adaptive FEC provides the highest *overall throughput* [118] among the various error control techniques.

By using an adaptive FEC encoder and decoder in the system, a high rate code is used under good channel conditions to maintain a high information rate. As the channel condition worsens, the information rate is reduced by applying lower rate codes to maintain the desired BER performance. Such a technique is very suitable to mobile communication systems, where adaptive signal compensation may provide adequate performance under channel conditions which are temporarily poor (due to multipath fading, shadowing, and rainfall, etc.) without a permanent bandwidth reduction caused by excessive coding overhead.

As we pointed out in Chapter 2, high rate punctured convolutional codes [44] and maximum distance separable block codes [74] such as RS codes are particularly well suited to implement the adaptive FEC encoder and decoder. In addition, rate 1/2 invertible BCH codes [111] can also be used in a delay-limited adaptive coding protocol. The advantage of using these codes is that a single decoder is employed to implement the adaptive capability of coding.

Notice that there is a fundamental trade-off involving error probability. We may define *protocol bit error probability* as the probability that the receiver commits a decoding error when an error control protocol is used. This probability is a measure of the communication reliability. In a conventional ARQ-based protocol for data communications, the protocol bit error probability must be made very small (typically 10^{-9}) because error-free transmission is a basic requirement. However, the cost of error-free transmission is that both the transmission delay and the error detection overhead are increased. For applications where transmitted power and transmission delay are critical, we may reduce the reliability or increase the protocol bit error probability, subject to a grade of service requirement (typically 10^{-6}). The delay-limited adaptive coding techniques do represent such an approach.

4.3 Description of the Truncated Protocol

The particular delay-limited adaptive coding scheme to be further analyzed is a truncated type-II hybrid-ARQ protocol. This protocol is a truncated version of the protocol originated by Wang and Lin [111]. The original protocol will henceforth be referred to as the untruncated protocol. The truncation limits the number of retransmissions to a finite number. For delay-limited applications, the smaller the delay the better. For this reason, and for the sake of simplicity, we consider the truncated type-II hybrid-ARQ protocol with a single retransmission.

In the truncated protocol, the first transmission of a message involves error detection coding only. When the receiver detects the presence of errors in a received word, it saves the erroneous word in a buffer and requests a retransmission. The retransmission is a block of parity bits which is obtained by applying a rate 1/2 invertible error correction code to the original message [69]. Either block or convolutional codes can be used, but in this chapter we consider only block codes. If no errors are detected on the first retransmission, the parity bits are inverted to recover the original information. If errors are detected, the parity bits are combined with the erroneous word stored in the receiver buffer to form a lower rate error correction code. Error correction is then performed and the decoded word is delivered to the receiver.

Like the Wang-Lin protocol, the truncated protocol employs two block codes, C_0 and C_1. C_0 is an (n, k) high-rate code used only for error detection; C_1 is a $(2n, n)$ invertible code which is designed for error correction. For mobile communications, where the error probability requirement is around 10^{-6} instead of "error-free" (say, 10^{-10}), the truncated protocol allows reduced signal power and provides a reduced transmission delay compared to the untruncated protocol.

In terms of design and implementation, the truncated protocol differs from the untruncated protocol in the following respects:

1. The reliability of the truncated protocol is determined by both C_0 and C_1

instead of by the error detection capability of C_0 only. Since only one retransmission is allowed, the error correction capability of C_1 should be selected to satisfy the required reliability level. The error detection capability of C_0 can be minimized subject to the reliability requirement in order to obtain additional coding gain.

2. In the untruncated protocol, each erroneously received message block is stored in a receiver buffer for error correction at a later time. Because the receiver buffer may be full, an extra mechanism must be introduced to resolve the possibility of buffer overflow. By contrast, receiver buffer design for the truncated protocol is much simpler because the required buffer size can be precisely predetermined. If the receiver buffer can accommodate the number of blocks sent during a round-trip transmission time period, there is no possibility of buffer overflow.

4.4 Delay Analysis

4.4.1 Time Delay and Queueing Models

As presented in Chapter 3, various ARQ and hybrid-ARQ systems can be viewed as simple queueing networks formed by two-way links and retransmission protocols. Improper design of coding schemes may lead to excessive time delay. Therefore, it is important to understand the nature and mechanism of delay.

In practice, delay in hybrid-ARQ systems consists of two components, namely queueing delay and transmission delay. *Queueing delay* is the delay between the time the message is assigned to a transmission queue (buffer) and the time it starts being transmitted. *Transmission delay* is the delay between the time the message starts being transmitted and the time it is successfully delivered to the user. In the following analysis, the queue discipline is again assumed to be *first in, first out.* The analysis is based on the single server M/G/1 queueing model, where messages arrive according to a Poisson process with rate λ and the message service times have a general distribution.

The use of queueing models often requires simplifying assumptions since, unfortunately, more realistic assumptions make meaningful analysis extremely difficult. For this reason, it is sometimes impossible to obtain accurate quantitative delay predictions on the basis of queueing models. In Chapter 3, we analyzed delay and packet loss probability for SAW- and GBN-based hybrid error control protocols. For hybrid error control schemes with a selective-repeat protocol, the analytical queueing solutions are still unknown. Fortunately, the truncated hybrid scheme with a selective-repeat protocol can be analyzed by using the method developed in the previous chapter. The imbedded Markov chain model provides a basis for delay approximations, as well as valuable insights.

4.4.2 Queueing Analysis

Suppose the truncated type-II hybrid-ARQ protocol is used for error control in an asynchronous time-division multiplexing system, where a large number of users are multiplexed onto a common channel. In order to carry out queueing delay analysis, we make the following basic assumptions:

- the transmitter buffer has infinite capacity.
- message blocks arrive at a Poisson rate of λ blocks per second.
- transmissions are synchronized to the occurrence of time slots, where a slot (T_s seconds) carries exactly one message block.
- errors on successive transmissions are independent events.

The system is modeled as an imbedded Markov chain. Because of the probability of error in the channel, two consecutive imbedded points are not separated by single time slot but by the time period of message interdepartures. Let a_i denote the number of message blocks that arrive during the ith interdeparture interval. Let n_i be the number of message blocks in the transmitter buffer at the end of the ith interdeparture interval. Since at most one message block leaves the buffer during this interval we have

$$n_{i+1} = n_i - U(n_i) + a_{i+1}, \tag{4.1}$$

where $U(*)$ is the unit step function. By evaluating the generating functions $P(z) = \sum_k z^k P[n_i = k]$, and $A(z) = \sum_k z^k P[a_i = k]$ on both sides of (4.1), we obtain the probability generating function of the system state:

$$P(z) = \frac{(1-\lambda T_s)(1-z)A(z)}{A(z)-z}. \tag{4.2}$$

Here $A(z)$ is the probability generating function for the number of message blocks arriving in an interdeparture interval. The procedures leading to equation (4.2) and the conditions for a steady-state solution are basically the same as those for the M/G/1 queue [45]. Let T_{max} be the maximum time period a message block could stay in the buffer. We call λT_{max} the *system load*. For a steady-state solution to exist we must have $\lambda T_{max} < 1$. By clearing fractions in equation (4.2) and differentiating successively, we obtain the average number of message blocks in the transmitter buffer

$$n_a = \frac{(1-\lambda T_s)A'(1)}{1-A'(1)} + \frac{A''(1)}{2[1-A'(1)]}. \tag{4.3}$$

For the truncated selective-repeat type-II hybrid-ARQ protocol with one retransmission, message blocks are transmitted in every slot. When a decoded message block is detected to be in error, the erroneous block is transmitted one more time. The transmitter is obliged to buffer previously transmitted blocks, until an ACK or NAK comes. If an ACK is received at the transmitter, the copy of the transmitted block is discarded. The block is retransmitted if an NAK is received, and then discarded after the retransmission. Therefore, if no error is detected after decoding, a message block will stay in the buffer for $R+1$ slots, where R denotes the round-trip delay in slots. If a retransmission occurs, a message block will stay in the buffer for $R+2$ slots. Thus the condition for a steady-state solution to exist is $\lambda T_s(R+2) < 1$.

If message blocks arrive at a Poisson rate of λ blocks per second, the probability of l blocks arriving in the interdeparture interval is

$$P(l/m\ retransmissions) = \frac{[\lambda(R+1+m)T_s]^l}{l!} e^{-\lambda(R+1+m)T_s}, \tag{4.4}$$

where $l = 0, 1, 2, \ldots$, and, for the truncated protocol, $m = 0, 1$. The probability generating function for $P(l/m\ retransmissions)$ is

$$G(z) = e^{-\lambda(R+1+m)T_s(1-z)}. \tag{4.5}$$

Let P_{B1} denote the probability of the event that the retransmission occurs. We have

$$P_{B1} = P_d = 1 - P_c - P_e. \tag{4.6}$$

Averaging (4.5) over m, the generating function for the number of message blocks in the interdeparture interval is given by

$$\begin{aligned} A(z) &= G(z|m=0)(1-P_{B1}) + G(z|m=1)P_{B1} \\ &= e^{-\lambda(R+1)T_s(1-z)}(1-P_{B1}) + e^{-\lambda(R+2)T_s(1-z)}P_{B1}. \end{aligned} \tag{4.7}$$

By differentiating $A(z)$ with respect to z, we have

$$A'(1) = \lambda T_s[(R+1) + P_{B1}], \tag{4.8}$$

and

$$A''(1) = (\lambda T_s)^2[(R+1)^2 + 2(R+1)P_{B1} + P_{B1}]. \tag{4.9}$$

Thus, by equation (4.3), the average queue length n_a is

$$n_a = \frac{(1-\lambda T_s)\lambda T_s[(R+1)+P_{B1}]}{1-\lambda T_s[(R+1)+P_{B1}]} + \frac{(\lambda T_s)^2[(R+1)^2+2(R+1)P_{B1}+P_{B1}]}{2-2\lambda T_s[(R+1)+P_{B1}]}. \tag{4.10}$$

From Little's formula (3.29) the queueing delay is then given by

$$D_q = \frac{n_a}{\lambda}. \tag{4.11}$$

Notice that the consideration of the truncated protocol with a single retransmission has greatly simplified the derivation of the generating function for the number of message blocks in the interdeparture interval. The analytical results for the untruncated type-II hybrid-ARQ protocol are much more involved and are not considered in this chapter. Nevertheless, it is clear that with a large number of retransmissions, the type-II hybrid-ARQ protocol will experience a greater queueing delay under poor channel conditions. In particular, the untruncated type-II hybrid-ARQ protocol has unbounded queueing delay. The delay will approach infinity when the channel bit error rate is beyond the error correction capability of the designed code.

4.4.3 Transmission Delay

When a message block is ready for transmission and the system queue is not empty, the block must wait in the queue until all previous blocks are transmitted. This waiting time is the queueing delay given by equation (4.11). If the queue is empty when a message block arrives, the block transmission will take place immediately in the next time slot. For the truncated protocol, consider a message block at the

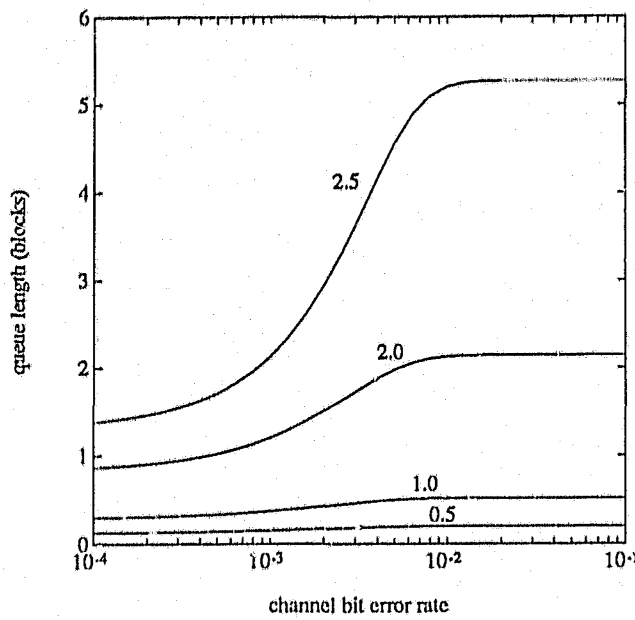

Figure 4.3: Queue length of the truncated protocol for arrival rates 0.5, 1.0, 2.0, 2.5

head of the transmission queue. We can easily calculate the transmission delay for this block as follows.

We denote propagation delay by T_p, channel data rate by R_d, and acknowledgement time by T_a. The first transmission delay, or the time taken to transmit a block if it is received correctly the first time, is equal to $T_p + n/R_d$. Round-trip delay, or the time between the reception of an incorrect block and the reception of the associated retransmission, is equal to $2T_p + T_a + n/R_d$. Thus, for the truncated protocol with one retransmission, the average transmission delay is

$$D_t = T_p + \frac{n}{R_d} + (2T_p + T_a + \frac{n}{R_d})P_{B1}. \tag{4.12}$$

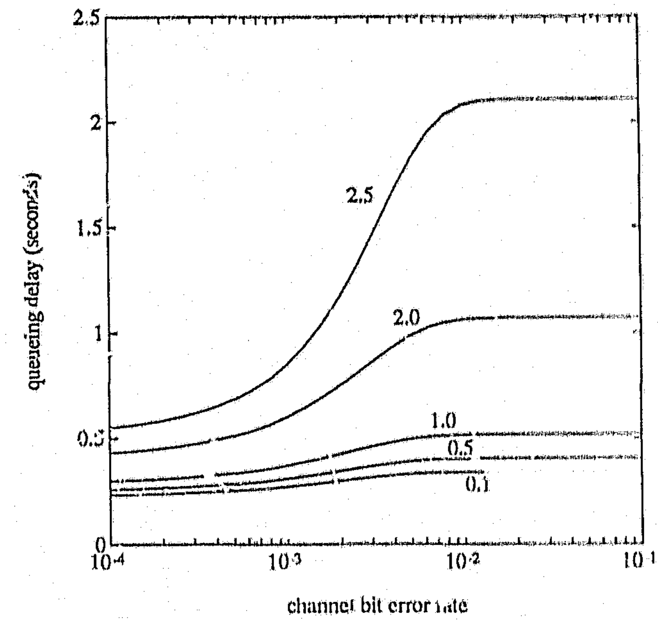

Figure 4.4: Queueing delay of the truncated protocol for arrival rates 0.1, 0.5, 1.0, 2.0, 2.5

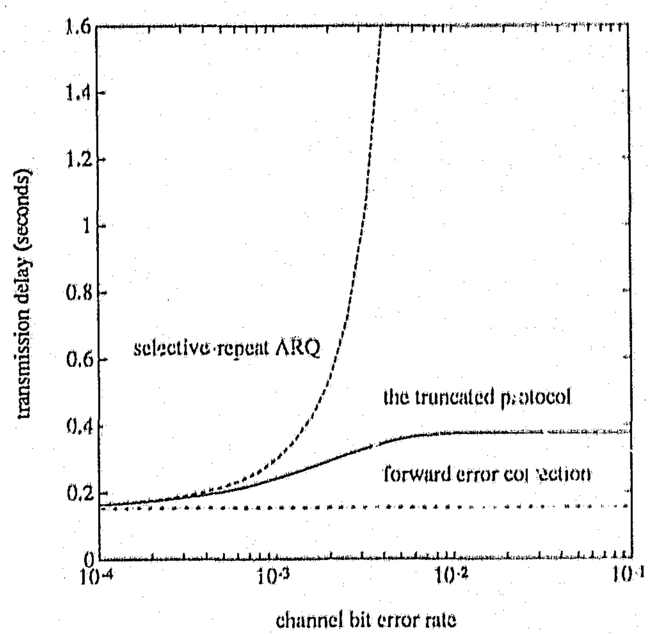

Figure 4.5: Transmission delay comparison of various coding protocols

4.5 Numerical Results and Discussion

Based on formulas (4.10), (4.11) and (4.12), numerical results for the queue length, queueing delay, and transmission delay can be obtained. Assume a channel data rate of $R_d = 4800$ bps, a block length $n = 500$ bits, a propagation delay $T_p = 50$ ms, and an acknowledgement time $T_a' = 20$ ms. Then we obtain the queue length in blocks versus channel bit error rate (p_b) as shown in Figure 4.3. The queueing delay in seconds versus channel bit error rate is shown in Figure 4.4. We can see that when the data arrival rate increases from 0.1 block/s (light traffic) to 2.5 block/s (heavy traffic), the queue length and queueing delay also increase. However, for any given data arrival rate, both queue length and queueing delay are bounded due to limited retransmissions.

The transmission delay in seconds versus average channel bit error rate is shown in Figure 4.5. We have also plotted the transmission delay for forward error correction and for the ideal selective-repeat ARQ protocol as references. The results clearly show that transmission delay for the truncated protocol is bounded by the maximum delay due to the limited number of retransmissions. By contrast, the transmission delay of the selective-repeat ARQ protocol can be very long (unbounded) when the channel bit error rate is beyond a certain limit, which may occasionally occur on a time-varying mobile fading channel.

4.6 Summary

In order to achieve high communication reliability, various ARQ and hybrid-ARQ protocols may impose a very long time delay. On the other hand, FEC coding provides limited delay at the expense of reliability or bandwidth. The delay constraints for a two-way transmission link often allow a small number of retransmissions, say one or two. Therefore, the hybrid-ARQ protocol with a finite number of retransmissions may offer a good mix of ARQ-based protocol and forward error correction coding.

The queueing models and the analytical results discussed in this chapter suggest that the time delay of an untruncated ARQ-based protocol in digital communication networks can be greatly reduced by protocol truncation. A trade off can then be made between the buffer size and error correction parameters to achieve maximum performance and minimum complexity. Specifically, the proposed truncated type-II hybrid-ARQ protocol has bounded time delay, which is determined by the delay constraint imposed on the coding design. Therefore, the truncated protocol can offer better performance than both the untruncated protocol and pure FEC in delay-limited applications.

Chapter 5

Coding Gain Improvement by Protocol Truncation

5.1 Introduction

In this chapter we continue our investigation of the truncated protocol. We turn our attention to the aspect of power saving for power- and delay-limited applications.

Error control coding and protocols can change signal quality from problematic to acceptable. Less power may thus be used to communicate between satellite and mobile terminals while error control techniques can be used to overcome the resulting loss in performance. In mobile data applications, hybrid-ARQ protocols have been proven to provide better reliability than a pure forward error correction and a higher throughput than the system with retransmission only [26] [118].

In future personal communication applications, signal power will be a critical resource and should be minimized subject to the user's desired grade of service. Much research has been done to study different approaches to improving power saving for point-to-point transmission in the face of unreliable channels [117]. However, in power limited mobile systems, there is little known about how to evaluate coding gain for a hybrid error control protocol on a mobile fading channel.

In this chapter, we propose the use of a generalized coding gain measure for performance evaluation of hybrid error control protocols on mobile fading channels. The truncated type-II hybrid-ARQ protocol proposed in the previous chapter is further discussed. In particular, we investigate the possible coding gain improvement obtained by using the truncated type-II hybrid-ARQ protocol on mobile fading channels, including a normalized Nakagami-m fading channel model as well as the well known Rayleigh and Rician channel models. The coding gain, transmission efficiency, and output error probability of the truncated protocol are discussed in detail. Numerical results are obtained for coding gain performance under various channel conditions for the truncated protocol and comparable coding methods.

The performance of error control protocols on mobile channels is often calculated in an average sense. The reason is that the signal-to-noise ratio varies in time due to fading. Our performance evaluation will be based on analytical channel models. In order to carry out the analysis, we assume that the feedback channel is error-free and that ideal interleaving is used.

5.2 Description of Channel Models

5.2.1 The Nakagami-m Fading Model

There are essentially four types of statistical channel models used to characterize fading radio signals, namely *Rayleigh*, *Rice*, *Nakagami*, and *lognormal* distributions [104]. Among these models, the Nakagami model is most versatile and best fits the bulk of field test data [3]. Thorough studies of Nakagami fading channels by Barrow [4] and others [88] [33] also show that communication performance problems are analytically tractable. Therefore, the Nakagami model provides an analytical basis for evaluating the coding gain performance of the truncated hybrid-ARQ protocol over time varying mobile fading channels, which include the additive white Gaussian noise channel and the Rayleigh fading channel as special cases.

We assume that the signal is modulated by binary differential phase shift keying (DPSK). Since DPSK requires phase stability over only two consecutive signaling intervals, this modulation technique is quite robust in the presence of signal fading. For DPSK on a Nakagami fading channel, the received signal in the interval $(0,T_b)$

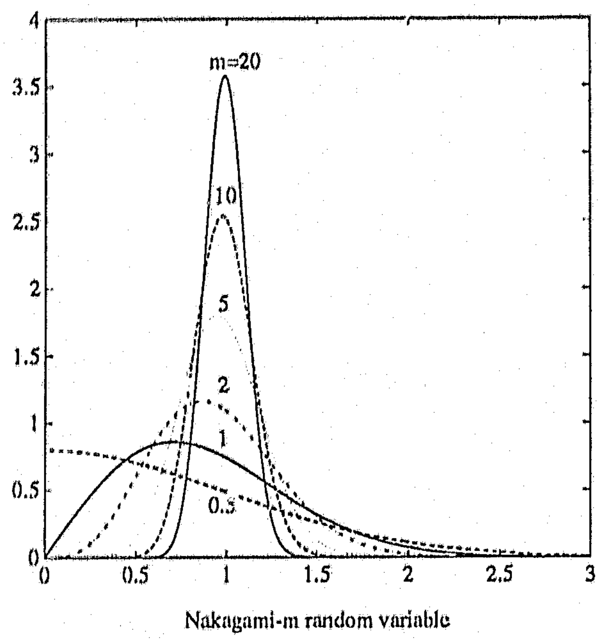

Figure 5.1: Probability density function of the Nakagami-m distribution

has the form

$$r(t) = \gamma g(t) + n(t), \tag{5.1}$$

where T_b is the bit duration, $g(t)$ is the transmitted DPSK signal with *signal bit energy* E_b, and $n(t)$ is AWGN with an one-sided noise spectral density N_0. The term γ, first proposed by Nakagami [86], and known as the nonselective fading envelope, is a random variable with an m-variate distribution. γ is a *Nakagami-m* random variable whose probability density function is given by

$$f(\gamma) = 2\frac{m^m \gamma^{2m-1}}{\sigma^m \Gamma(m)} exp(-\frac{m\gamma^2}{\sigma^2}); \quad \gamma \geq 0. \tag{5.2}$$

Here σ^2 is the average power of the fading envelope γ, m is a parameter representing the inverse of the normalized variance of γ^2, and $\Gamma(m)$ is the Gamma function of

m. It can easily be verified that the Nakagami-m distribution has the following properties:

1. The case $m = 1/2$ corresponds to the one-sided Gaussian distribution.
2. The case $m = 1$ corresponds to a Rayleigh distribution
3. As $m \to \infty$, the density tends to an impulse function. γ therefore becomes a constant (nonfading) so that the received signal is affected by AWGN only.

Without loss of generality, we adopt the convenient normalization $\sigma^2 = 1$ [27], so that the received signal in the fading channel has an average power equal to E_b. Thus we obtain a one-parameter Nakagami-m distribution with the density

$$f(\gamma) = \frac{2m^m \gamma^{2m-1}}{\Gamma(m)} exp(-m\gamma^2); \quad \gamma \geq 0. \tag{5.3}$$

This probability density function is plotted in Figure 5.1 for various values of m.

5.2.2 Mobile Satellite Channel Models

The important mobile satellite channel models are first briefly reviewed in this section. The characterization of a mobile satellite channel is very complex. There are essentially three types of statistical channel models used to characterize mobile satellite radio signals, namely *Rayleigh*, *Rice*, and *Log-Normal* distributions [71]. The signal is again assumed to be modulated by binary differential phase shift keying (DPSK).

A satellite signal received by a mobile terminal can be approximated by a direct component (unshadowed or shadowed) and a diffuse component (Rayleigh distributed). The direct component is the line-of-sight signal between the mobile terminal and the satellite. It is affected by the ionosphere and the troposphere and other natural or human-made obstacles on the earth's surface. The diffuse component of the received signal represents the multipath reflection from the surrounding terrain.

For the diffuse component, the probability density of the signal amplitude (envelope) γ is the Rayleigh density function given by

$$f(\gamma) = \frac{\gamma}{b_0} exp \left[-\frac{\gamma^2}{2b_0} \right], \tag{5.4}$$

where b_0 is the mean-square value or power of the diffuse component γ.

When the direct component is unshadowed, the received satellite signal is Rice distributed with the following density function:

$$f(\gamma) = \frac{\gamma}{b_0} exp \left[-\frac{\gamma^2 + A_c^2}{2b_0} \right] I_0(\gamma A_c / b_0), \tag{5.5}$$

where b_0 is again the power of the diffuse component. A_c is the line-of-sight signal amplitude, and $I_0(\cdot)$ is the modified Bessel function of zeroth order.

When the direct component is shadowed, the received signal amplitude γ can be characterized by a Log-Normal density function given by

$$f(\gamma) = \frac{1}{\sqrt{2\pi d_0}\,\gamma}\, exp \left[-\frac{(ln\,\gamma - \mu_0)^2}{2d_0} \right], \tag{5.6}$$

where μ_0 and d_0 are the mean and variance of $ln\,\gamma$.

For a land mobile satellite channel, the radio path may be modeled as the combination of a shadowed direct component and a diffuse component. In this case, the received signal can be statistically described by the following density function:

$$f(\gamma) \;=\; \frac{\gamma}{b_0\sqrt{2\pi d_0}} \int_0^\infty \frac{1}{Z} exp \left[-\frac{(ln\,Z - \mu_0)^2}{2d_0} - \frac{\gamma^2 + Z^2}{2b_0} \right] I_0(\gamma Z / b_0)\, dZ$$

$$\approx \begin{cases} \frac{1}{\sqrt{2\pi d_0}\,\gamma}\, exp\left[-\frac{(\ln\gamma-\mu_0)^2}{2d_0}\right] & \text{(Log-Normal)} \quad \gamma \gg \sqrt{b_0} \\ \frac{\gamma}{b_0} exp\left[-\frac{\gamma^2}{2b_0}\right] & \text{(Rayleigh)} \quad \gamma \ll \sqrt{b_0} \end{cases}, \tag{5.7}$$

5.3 Coding Gain Analysis

5.3.1 Generalized Coding Gain

In deriving the performance of binary DPSK for various fading channels, we begin with the bit error probability for a nonfading channel, which is

$$P_b(\gamma) = \frac{1}{2} exp(-\frac{\gamma^2 E_b}{N_0}). \tag{5.8}$$

We view (5.8) as a conditional error probability, where the condition is that γ is fixed. When we average $P_b(\gamma)$ over the Nakagami distributed fading parameter γ, the resulting bit error probability is [4]

$$\begin{aligned} P_b &= \int_0^\infty P_b(\gamma) f(\gamma) d\gamma \\ &= \frac{1}{2}(\frac{m}{m+E_b/N_0})^m, \end{aligned} \tag{5.9}$$

where E_b/N_0 is the average *signal-to-noise ratio* in power units. When the transmitted digital signal is encoded, the bit error probability becomes

$$P_b = \frac{1}{2}(\frac{m}{m+E_{bc}/N_0})^m, \tag{5.10}$$

where E_{bc} represents the signal bit energy after encoding.

In many mobile communication applications, signal power is a critical resource and should be minimized subject to the user's desired grade of service. The power saving obtained through FEC coding is described as coding gain [11]. As discussed

CHAPTER 5. CODING GAIN IMPROVEMENT 92

in Chapter 2, the coding gain is normally defined as the difference between E_b/N_0 ratios needed to achieve a given bit error probability with coding and without coding. For our truncated protocol, however, this definition does not account for the additional signal power due to possible retransmissions. Therefore, the average signal-to-noise ratio has to be generalized in order to evaluate the performance of various protocols on a common basis. A straightforward generalization can be obtained by simply defining E_{bc}/N_0 as [87] [120]

$$\frac{E_{bc}}{N_0} = \frac{1}{\mu} \cdot \frac{E_b}{N_0}, \tag{5.11}$$

where μ is defined as the average number of channel bits required to transmit an information bit. Here μ is introduced to account for the average signal bit energy increase due to retransmissions. We call μ the *transmission efficiency.* For FEC coding, μ is the inverse of the error correction coding rate. For any scheme involving retransmissions, including ARQ-based protocols and the truncated protocol, μ is the inverse of the throughput multiplied by the error detection coding rate. This definition is easily justified by noting that a retransmission of the signal bit will double the signal bit energy.

When a coding protocol is used, the average bit error probability of DPSK over the Nakagami-m fading channel is given by

$$P_b = \frac{1}{2}\left(\frac{m}{m + \frac{1}{\mu}\frac{E_b}{N_0}}\right)^m. \tag{5.12}$$

Similarly, for Rayleigh fading channels, it can be easily found that

$$P_b = \frac{1}{2(1 + b_0 \frac{E_b}{N_0})}. \tag{5.13}$$

For Rician fading channels, we find

$$P_b = \frac{1}{2(1 + b_0\frac{E_b}{N_0})}\, exp\left[-\frac{\frac{E_b}{N_0}}{1 + b_0\frac{E_b}{N_0}}\right]. \tag{5.14}$$

However, for Log-Normal fading channels and the more general cases described by (5.7), numerical computation is normally required.

The above expressions will be used later to calculate the coding gain of the truncated protocol and other coding protocols.

5.3.2 Transmission Efficiency

We now analyze the transmission efficiency of the truncated selective-repeat type-II hybrid-ARQ protocol. The transmission efficiency μ will be used to determine error performance on fading channels.

Under the noiseless feedback assumption, the analysis of transmission efficiency is similar to the throughput analysis of the untruncated protocol. However, the constraint of one retransmission allows us to obtain the transmission efficiency directly. The analysis in [11] is only capable of finding a bound on the throughput by considering a system inferior to the untruncated protocol.

Suppose that C_0 is an (n, k) error detection code and C_1 is a $(2n, n)$ invertible code with the capability of correcting up to t errors. In the following analysis we assume that messages are transmitted in the form of blocks consisting of either one message block I in C_0 or one parity block P. The receiver buffer can accommodate the maximum number of message blocks transmitted during a round trip transmission time.

For a transmitted message block I to be successfully delivered to the sink using the truncated selective-repeat type-II hybrid-ARQ protocol, the average number of channel bits required per information bit is the average number of blocks required to transmit the message block divided by the error detection coding rate. Through a straightforward application of the results in [111], the transmission efficiency can be found as

$$\mu = \frac{n}{k}[P_c + 2(1 - P_c - P_e)P_t], \tag{5.15}$$

where k/n is the code rate, P_c is the probability that the received message block contains no error, P_e is the undetected error probability for the code C_0, and P_t is the conditional probability that the receiver recovers a message block after receiving a parity block. We have

$$P_c = (1 - P_b)^n. \tag{5.16}$$

If C_0 is properly chosen, P_e is upper bounded as follows [68]:

$$P_e \leq [1 - (1 - P_b)^k]2^{-(n-k)}. \tag{5.17}$$

As shown in [111], P_t can be expressed as

$$P_t = P_c + (1 - P_c - P_e)\frac{q_0 - y}{1 - y}, \tag{5.18}$$

where

$$q_0 = \sum_{j=0}^{t} \binom{2n}{j} P_b^j (1 - P_b)^{2n-j}, \tag{5.19}$$

and

$$y = (1 - P_b)^n \left[2 \sum_{l=0}^{t} \binom{n}{l} P_b^l (1 - P_b)^{n-l} - (1 - P_b)^n\right]. \tag{5.20}$$

Substituting (5.19) into (5.16), we obtain the following result for the transmission efficiency of the truncated protocol:

$$\begin{aligned} \mu &= \frac{n}{k}\left[P_c + 2(1-P_c-P_e)(P_c + (1-P_c-P_e)\frac{q_0-y}{1-y})\right] \\ &\leq \frac{n}{k}\left[P_c + 2(1-P_c)(P_c + (1-P_c)\frac{q_0-y}{1-y})\right]. \end{aligned} \tag{5.21}$$

5.3.3 Protocol Error Probability

The *protocol error probability*, denoted by $Pr(E)$, is the probability that the receiver commits a decoding error when the error control protocol is used. This probability is a measure of the reliability of the communication system. There is a fundamental trade-off involving this error probability. In a conventional (untruncated) ARQ-based protocol for data communications, $Pr(E)$ must be minimized because error-free transmission is a basic requirement. However, the cost of error-free transmission is that both the transmission delay and the error detection overhead will be increased. For applications where transmitted power and transmission delay are critical, we may reduce the reliability subject to a grade of service requirement. The truncated protocol represents such an approach.

For the truncated protocol with one retransmission, the protocol error probability is

$$\begin{aligned} Pr(E) &= Pr(A_0^e) + Pr(A_0^d E_1^E) \\ &= Pr(A_0^e) + Pr(A_0^d)Pr(E_1^E|A_0^d), \end{aligned} \tag{5.22}$$

where A_0^e and A_0^d are the events that a message block I in C_0 contains, respectively, undetectable errors and detectable errors, and E_1^E is the event that, after receiving the parity bits of the transmitted block (retransmission), the receiver commits a decoding error by accepting a decoded message block I^* with uncorrectable errors. Reliability considerations for the truncated protocol differ from those for the untruncated protocol and FEC coding. In an untruncated protocol, reliability is mainly determined by the undetected error probability for C_0 [68]. In forward error correction coding, only uncorrectable errors are considered. In the truncated protocol, both undetectable and uncorrectable errors are taken into account. Based on the definitions of error events, we have

$$Pr(A_0^e) = P_e, \tag{5.23}$$

where P_e is given by (5.18), and

$$Pr(A_0^d) \equiv P_d = 1 - P_c - P_e, \tag{5.24}$$

where P_c is given by (5.17).

Let q_1 be the conditional probability of correctly decoding the received message-parity pair $(\hat{I}, \hat{P})$ given that both $\hat{I}$ and $\hat{P}$ are detected in error. q_1 is given by [111]

$$q_1 = \frac{q_0 - y}{1 - y}, \tag{5.25}$$

where q_0 and y are given by (5.20) and (5.21), respectively.

After receiving the parity bits, the receiver commits a decoding error by accepting a received message block with undetected errors or accepting a decoded message block I^* with uncorrected errors. Therefore

$$Pr(E_1^E|A_0^d) = P_e + P_d(1 - q_1). \tag{5.26}$$

Substituting (5.24) to (5.27) into (5.23), we obtain the following result:

$$Pr(E) = P_e + P_d(P_e + P_d(1 - \frac{q_0 - y}{1 - y})). \tag{5.27}$$

From this result we see that the reliability of the truncated protocol is determined by both the error detection capability of code C_0 and the error correction capability of the code C_1.

5.4 Numerical Results and Discussion

5.4.1 Coding Gain Comparison

Some examples are now given to provide a comparison of coding gain for the truncated protocol and other comparable coding strategies. The following four coding protocols as well as the uncoded case are used in our comparison:

1. A pure FEC scheme employing a (500,470,3) BCH code (high rate);
2. A pure FEC scheme employing a (1000,470,59) BCH code (low rate);
3. A Wang-Lin type-II hybrid-ARQ protocol employing a (1000,500,55) error correction code and a (500,470) error detection code;
4. A truncated type-II hybrid-ARQ protocol which also employs a (1000,500,55) error correction code and a (500,470) error detection code.

Notice that for each transmission, the coding strategies 1, 3, and 4 have the same bandwidth efficiency so that they can be compared on a common bandwidth basis. On the other hand, the coding strategies 2, 3, and 4 have the same designed code rate so that they can be compared on the same coding capability basis.

For the truncated protocol, we have obtained the protocol error probability given by (5.28). For the untruncated protocol, the protocol error probability is approximately given by [111]

$$Pr(E) \approx (1 + P_d^2)\frac{P_e}{P_c + P_e}. \tag{5.28}$$

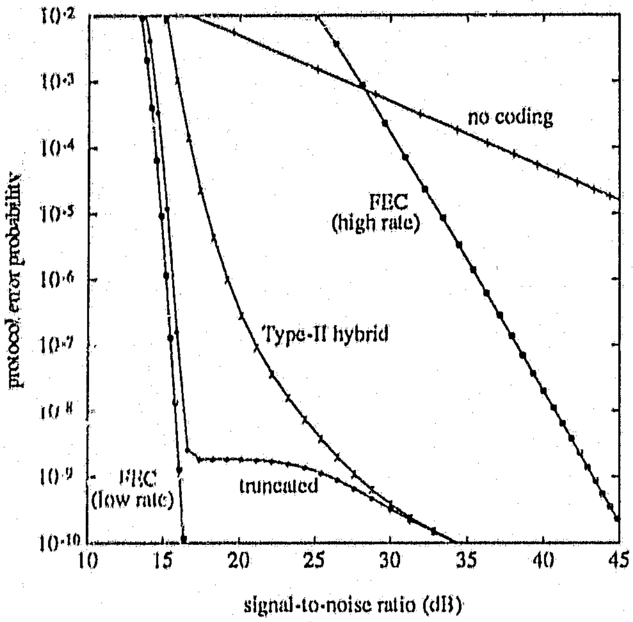

Figure 5.2: Coding gain analysis ($m = 1$, Rayleigh fading channel)

The protocol error probability in FEC is simply the probability of incorrect decoding, denoted by P_{ICD}. We assume that *bounded-distance decoding* is used in FEC. The bounded-distance decoding procedure corrects all error patterns of weight t or less and no others, where t is the largest integer equal to or less than $(d-1)/2$ [81]. Since there is no error detection involved, for an (n,k,t) linear block code with bounded-distance decoding, the probability of incorrect decoding on a memoryless binary symmetric channel is given by [81]

$$P_{ICD} = \sum_{i=t+1}^{n} \binom{n}{i} P_b^i (1 - P_b)^{n-i}. \tag{5.29}$$

For DPSK modulation on a Nakagami-m fading channel, the channel bit error probability P_b is given by (5.13).

Figure 5.2 shows the average signal-to-noise ratio versus the protocol error probability for the above four coding strategies as well as the uncoded case. In regard to the coding gain improvement, we have the following comments:

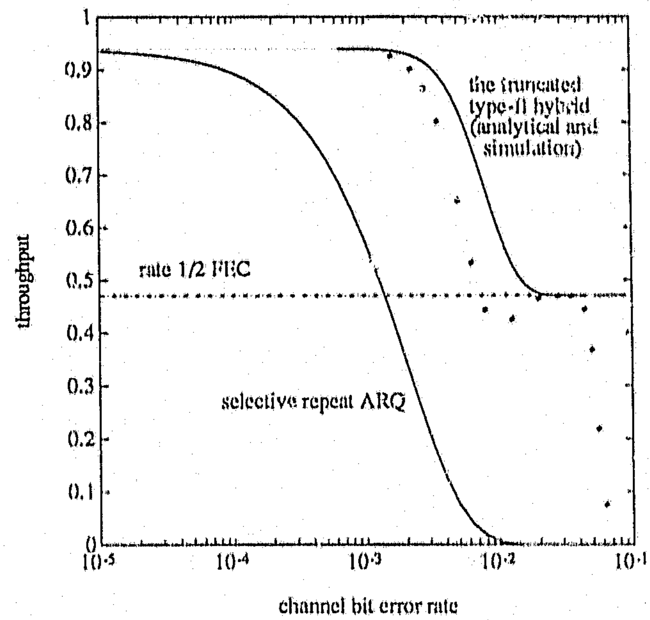

Figure 5.3: Throughput comparison of various coding protocols, where simulation results shown count only error-free blocks

- When the signal-to-noise ratio is high, i.e., the channel is in good condition, the performance of the truncated protocol approaches the performance of the untruncated type-II hybrid-ARQ protocol. This is because in most cases no retransmissions are required when the channel is good.

- When the signal-to-noise ratio is low, the performance of the truncated protocol approaches the performance of low-rate FEC code, providing improved power saving and reduced time delay in comparison to the untruncated protocol. This is because the extra retransmissions in the untruncated protocol are simply wasted if the goal of the system design is to provide reasonable reliability (say, 10^{-7}) instead of "error-free" performance (say, $< 10^{-10}$).

- When the signal-to-noise ratio is low, the performance of the truncated protocol is slightly inferior to the performance of low-rate FEC. This is because the truncated protocol must reserve some parity bits for error detection instead of making full use of the code's error correction capability.

- Both the truncated and the untruncated protocol outperform the high-rate FEC strategy at the expense of increased complexity.

From Figure 5.2 one may conclude that the low-rate FEC is the best in terms of coding gain improvement. However, since most practical channels are in good condition most of the time, the low-rate FEC does not make efficient use of the channel. This can be clearly seen when we compare the throughput of a low-rate FEC strategy and the truncated protocol as shown in Figure 5.3. Simulation results are also shown for the truncated protocol. It can be seen that the truncated protocol provides twice the throughput of the low-rate FEC when the channel bit error rate is below 10^{-3}.

Figures 5.4 to 5.8 compare the coding gain of the truncated protocol and the untruncated protocol for other values of m. Different values of the parameter m represent different channel fading conditions.

Figures 5.9, 5.10 and 5.11 further show the average signal-to-noise ratio (E_b/N_0) versus the protocol error probability on the Rayleigh, Rician, and AWGN channel, respectively. Notice that on the Rayleigh fading channel, it is shown that the truncated protocol is 3 dB, 5 dB, and 8 dB better than the type-II hybrid-ARQ at BER levels of 10^{-6}, 10^{-7}, and 10^{-8}, respectively.

Suppose that we are given design requirements for reliability, bandwidth efficiency, and transmission delay. From the numerical results shown, we see that by proper selection of both the error detection and the error correction codes, the truncated protocol provides significant coding gain improvement over the untruncated type-II hybrid-ARQ protocol. Moreover, the implementation complexity for the truncated protocol is less than that for the untruncated protocol because both the transmitter buffer and the receiver buffer are easier to design for the truncated protocol than the untruncated protocol.

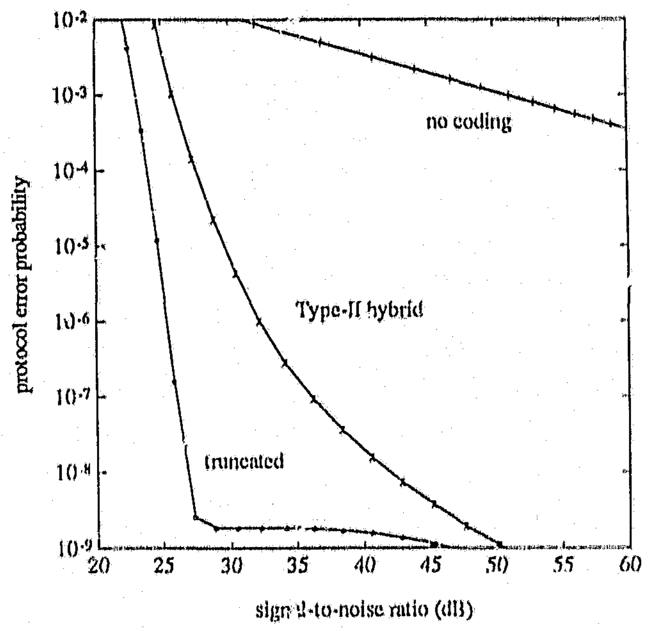

Figure 5.4: Coding gain analysis ($m = 0.5$)

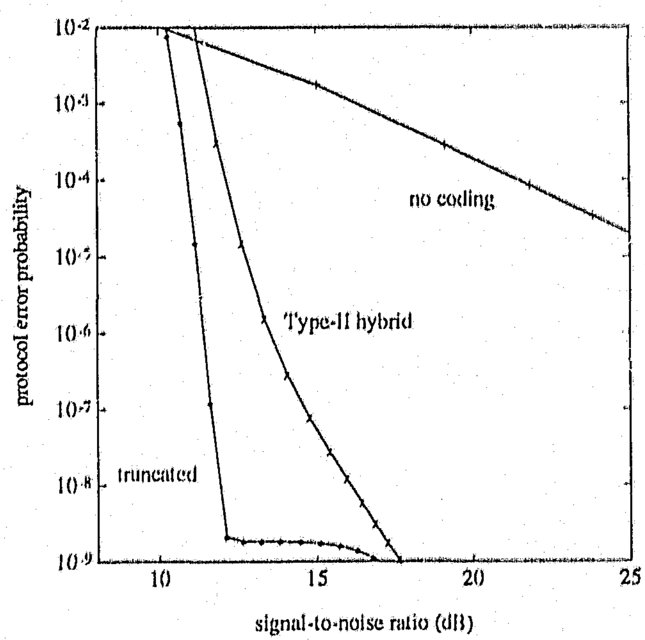

Figure 5.5: Coding gain analysis ($m = 2$)

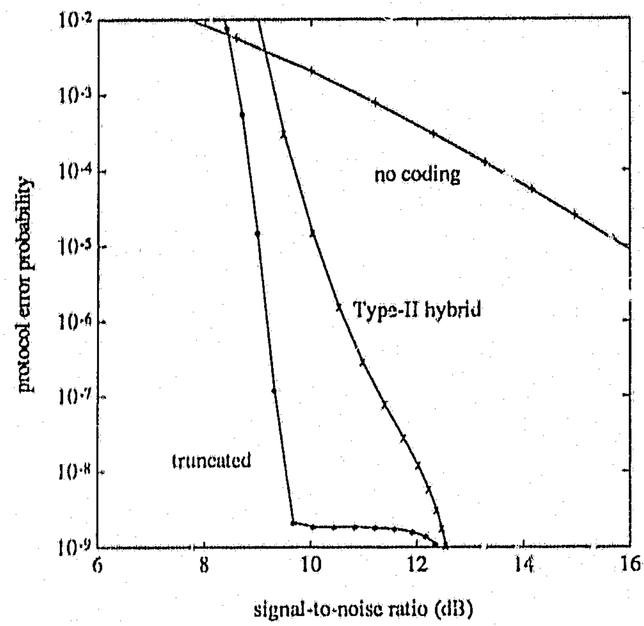

Figure 5.6: Coding gain analysis ($m = 5$)

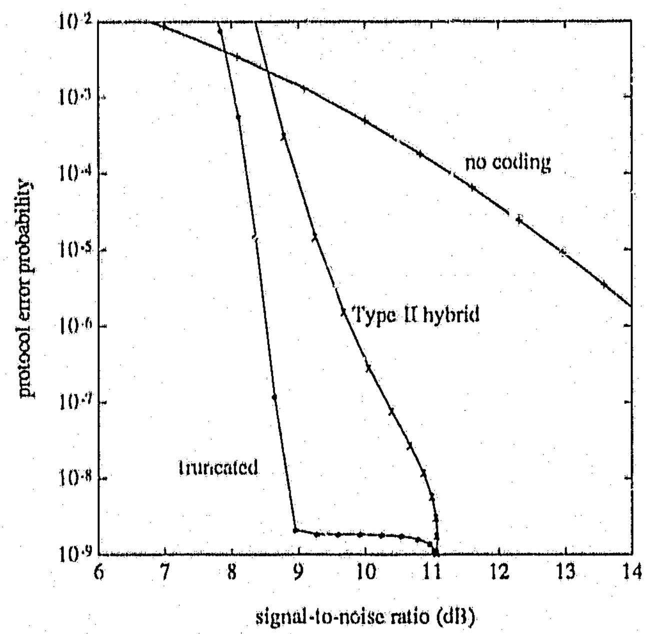

Figure 5.7: Coding gain analysis ($m = 10$)

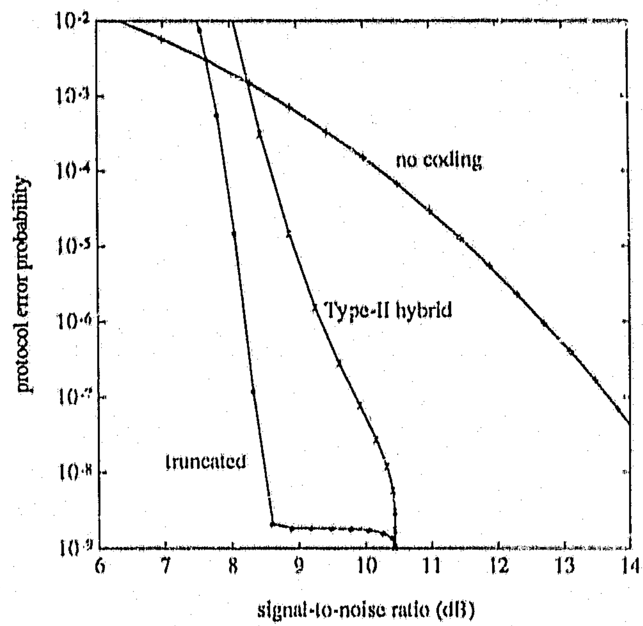

Figure 5.8: Coding gain analysis ($m = 20$)

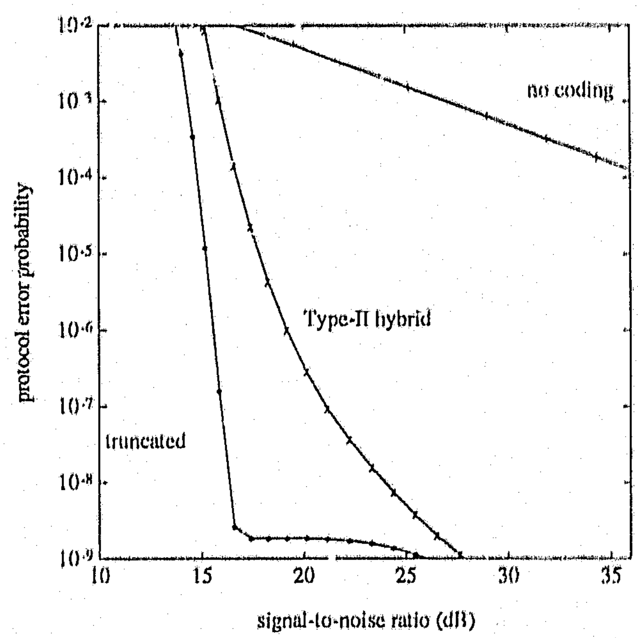

Figure 5.9: Coding gain analysis on a Rayleigh fading channel

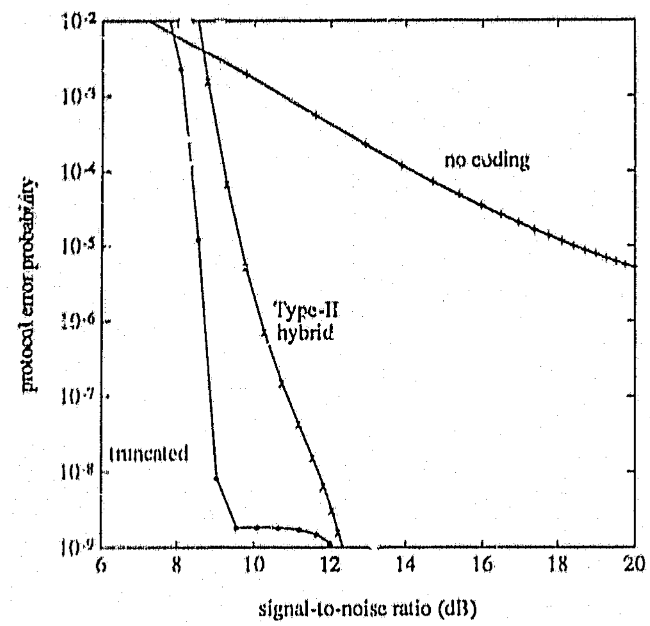

Figure 5.10: Coding gain analysis on a Rician fading channel

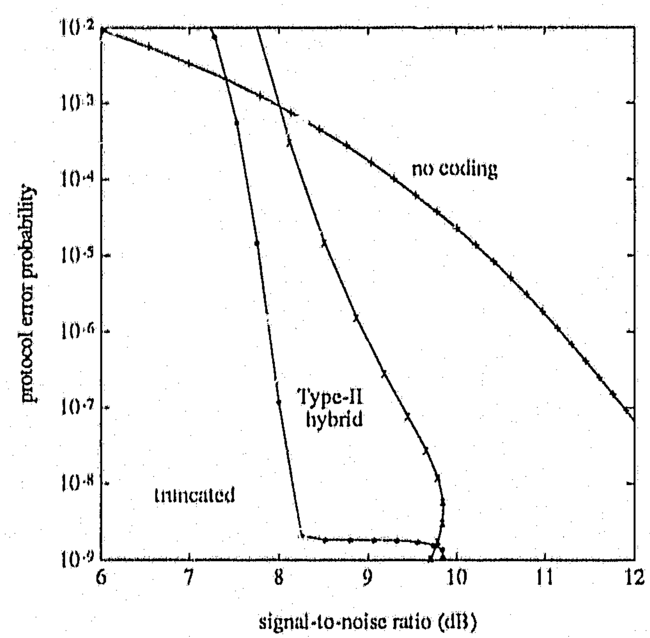

Figure 5.11: Coding gain analysis on an AWGN channel

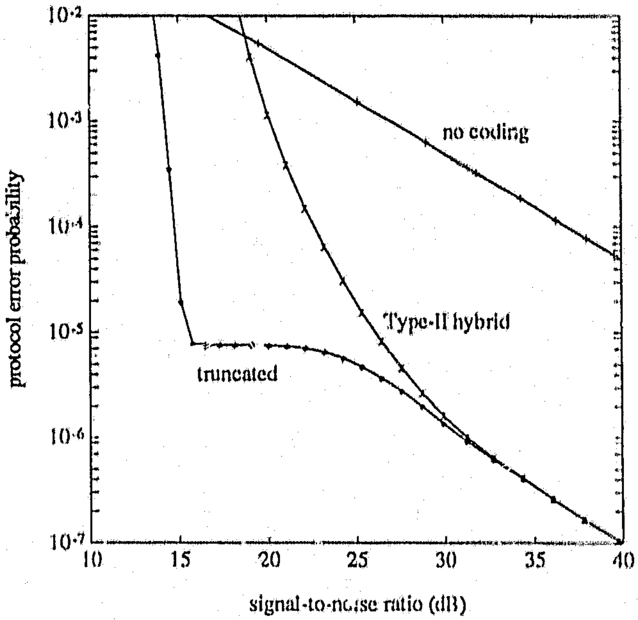

Figure 5.12: Impact of error detection capability on performance ($m = 1, n - k = 18$)

5.4.2 Optimum Error Detection

In the Wang-Lin protocol, the capability of the error detection code is determined by the reliability requirement. Power saving in terms of signal-to-noise ratio is not a major concern. However, in the truncated protocol, the selection of error detection capability is crucial. The error detection code should be chosen such that the signal-to-noise ratio is minimized subject to the given reliability requirement. Figures 5.12 and 5.13 exhibit the behavior of two different error detection codes with the same code length. There is a bit error probability floor between 15 dB and 25 dB in both cases. These floors are determined by the number of parity bits in the error detection code C_0.

In practice, if many parity bits are allocated for the error detection code, providing a very low protocol error probability (say, 10^{-10}), the extra parity bits are wasted for a typical error probability requirement of 10^{-6}. Unnecessary parity bits also reduce the coding gain when the truncated protocol is used. Therefore, for a

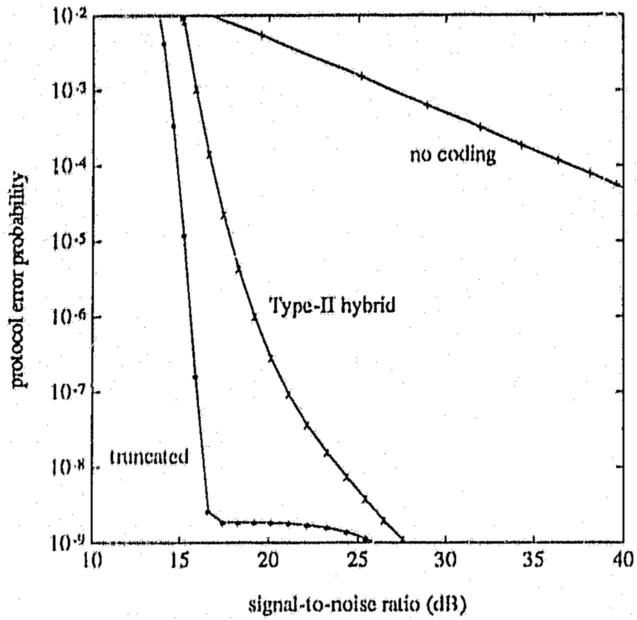

Figure 5.13: Impact of error detection capability on performance ($m = 1, n - k = 27$)

given reliability requirement, we can find an optimum error detection code for the protocol.

Suppose that a (1000,500) invertible BCH code [67] is specified for the code C_1. We wish to determine the error detection code C_0 that will achieve the maximum power savings. Assuming a fading channel parameter of $m = 1$, the minimum required signal-to-noise ratio corresponding to various numbers of parity bits is given in Table 5.1. Reliability requirements of 10^{-5}, 10^{-6}, and 10^{-7} are assumed. Table 5.1 shows that the required signal-to-noise ratio achieves a minimum with $n - k = 27$ parity bits. Notice that this improved coding gain is achieved at the expense of a slightly reduced throughput.

5.4.3 Queueing Results on Nakagami Fading Channel

Based on queueing results derived in Chapter 4, further numerical results for the queue length and queueing delay on mobile fading channels can be obtained.

Table 5.1: Optimum error detection

	Required signal-to-noise ratio(dB)										
$n-k$	0	9	18	27	36	45	54	63	72	81	90
$Pr(E) = 10^{-5}$	47.0	46.9	15.4	15.2	15.3	15.4	15.5	15.6	15.6	15.7	15.8
$Pr(E) = 10^{-6}$	57.0	56.9	30.4	15.6	15.7	15.8	15.8	15.9	16.0	16.1	16.2
$Pr(E) = 10^{-7}$	67.0	66.9	40.0	15.9	16.0	16.1	16.2	16.2	16.3	16.4	16.5

Consider the Nakagami fading channel and again assume a channel data rate of $R_d = 4800$ bps, a block length $n = 500$ bits, a propagation delay $T_p = 50$ ms, and an acknowledgement time $T_a = 20$ ms. The queue length in blocks versus average channel bit error rate (P_b) is shown in Figure 5.14 and Figure 5.15. The queueing delay in seconds versus average channel bit error rate (P_b) is shown in Figure 5.16 and Figure 5.17. Again we can see that when the data arrival rate increases from 0.1 (light traffic) to 2.5 (heavy traffic), the queue length and queueing delay also increase. However, for any given data arrival rate, both queue length and queueing delay are bounded due to limited retransmissions.

5.5 Summary

An analytical approach has been employed to evaluate the coding gain of the truncated hybrid error control protocol on mobile fading channels. By using this approach, the power savings offered by various error control protocols can be compared on a common basis. The channel models which we have considered include the *Nakagami-m*, *Rayleigh*, *Rice*, and *Log-Normal* fading distributions. In particular, the truncated protocol is shown to provide significant coding gain improvement (5 - 8 dB) over the comparable FEC and the conventional type-I or type-II hybrid-ARQ protocols for delay-limited mobile applications.

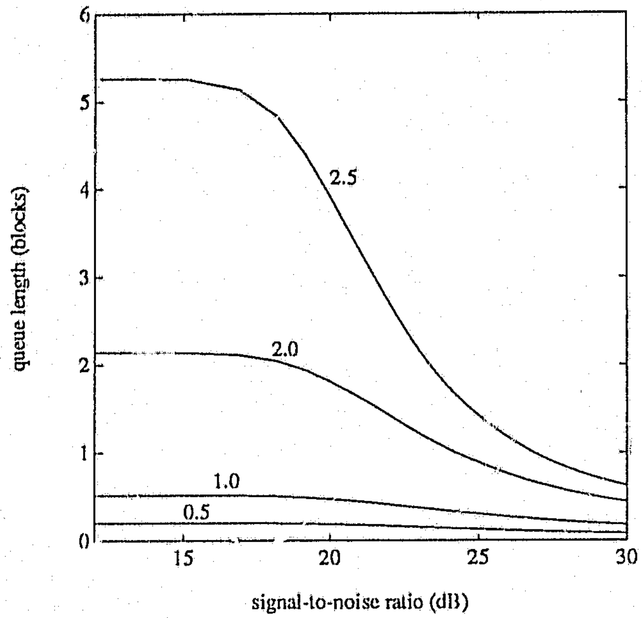

Figure 5.14: Queue length of the truncated protocol over Nakagami fading channel (m=1) for arrival rates 0.5, 1.0, 2.0, 2.5

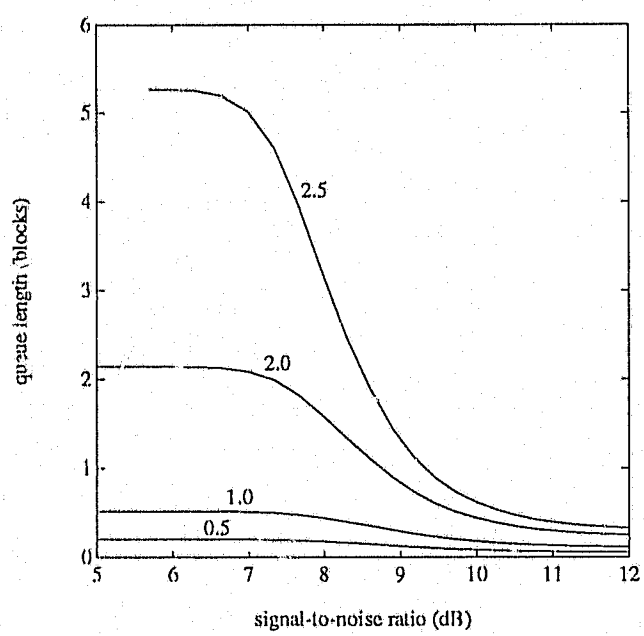

Figure 5.15: Queue length of the truncated protocol over Nakagami fading channel (m=10) for arrival rates 0.5, 1.0, 2.0, 2.5

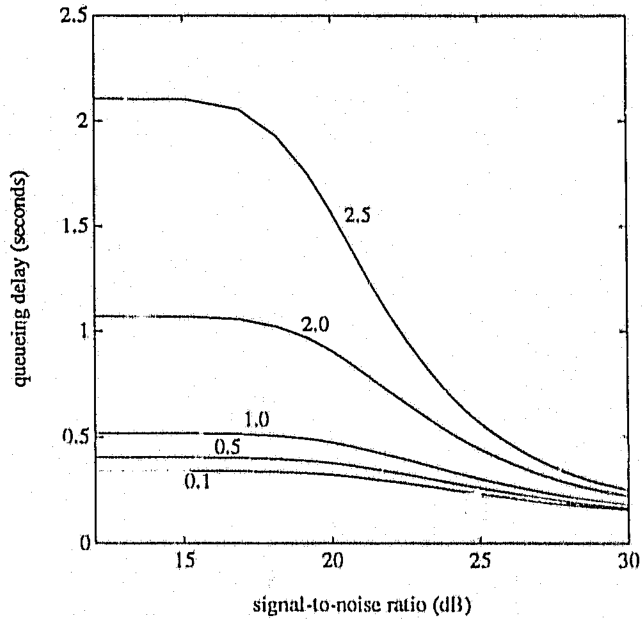

Figure 5.16: Queueing delay of the truncated protocol over Nakagami fading channel (m=1) for arrival rates 0.1, 0.5, 1.0, 2.0, 2.5

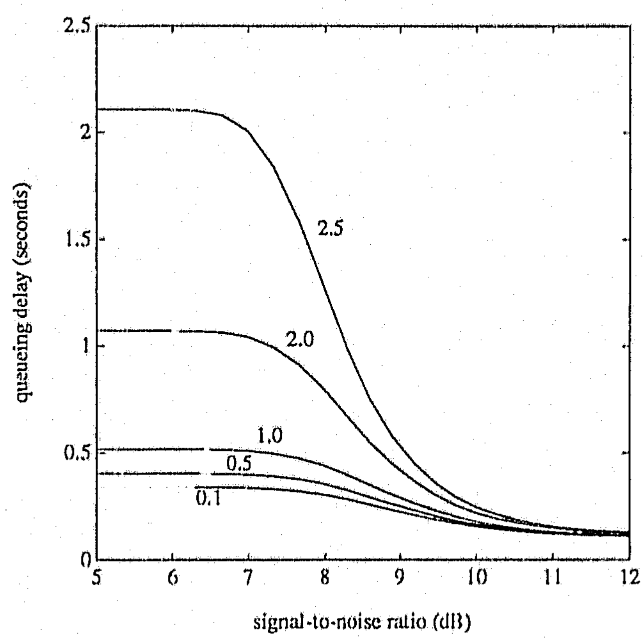

Figure 5.17: Queueing delay of the truncated protocol over Nakagami fading channel (m=10) for arrival rates 0.1, 0.5, 1.0, 2.0, 2.5

Chapter 6

Matched-rate Adaptive Coding with Feedback Transmissions

6.1 Introduction

Up to now, code parameters have been predetermined in every hybrid error control protocol we have analyzed. Whether or not the predetermined code parameters are matched with the real channel conditions has not been considered. In order to further improve the performance, we would like to investigate the problem of matching code parameters with changing channel conditions.

In mobile communications, the control of transmission errors is a major challenge due to fading and other time-varying interference. Satellite communications with land-mobile terminals also suffer from large variations in the received signal power due to multipath fading, signal shadowing and rain attenuation [20] [58]. Simple ARQ protocols often provide very low channel efficiency or throughput in such noisy links. For real channels with a time-varying statistic, one would like to design an adaptive error control coding system. When the channel is good, the system would have only error detection parity bits included in each transmission. When the channel becomes noisy, extra error correction parity bits should be added to match the channel condition, resulting in efficient channel use.

In the previous chapters, we have discussed various types of hybrid error control techniques. These more sophisticated protocols make use of both error detection and error correction coding to achieve high throughputs and low undetected error probabilities. However, often the error correction capability in a hybrid-ARQ protocol is not matched with the actual channel conditions. In other words, the code rate is not optimum in terms of efficient use of the channel. For example, the well-known type-II hybrid-ARQ protocol using a half-rate invertible code [111] has only two adaptable code rates. As is shown in this chapter, the throughput provided by this half-rate type-II hybrid-ARQ protocol is far less than the maximum achievable throughput.

More recently, several adaptive ARQ-based protocols have been proposed to further improve throughput [109] [100] [77]. In these adaptive protocols, the optimum code rate or optimum block size is selected according to the actual channel state. However, the channel state estimation in these protocols is based on counting the number of erroneous blocks. Since an erroneous block is defined as a block containing one or more erroneous bits, a good channel state estimation requires the transmission of many blocks, and hence is slow. In [44] and [51], generalized type-II hybrid-ARQ protocols with incremental redundancy have been proposed and analyzed. However, as in a conventional type-II hybrid-ARQ protocol, the generalized protocol starts with the highest code rate so that the code rate may not be quickly matched with the channel condition if a poor channel state lasts for a long time period, as it does in some mobile data links. Efficient adaptive protocols which are able to promptly match the error correction coding capability with the changing channel conditions have yet to be developed and investigated.

In this chapter, we propose a new adaptive error control protocol using Reed-Solomon codes. The protocol uses feedback transmissions of received blocks to achieve faster channel state estimation. Such a protocol can be applied to many applications where a feedback channel is available and not fully utilized, such as file transfer in a two-way communication link. In section 6.2, we describe the proposed hybrid-ARQ protocol, which uses an RS code with feedback transmissions. Section 6.3 analyzes throughput of the proposed protocol as well as the type-II

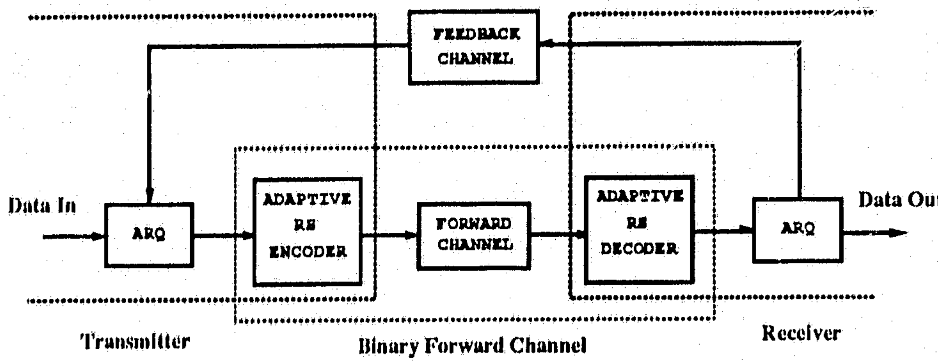

Figure 6.1: Hybrid-ARQ error control using a rate-adaptive Reed-Solomon code

hybrid-ARQ protocol using a half-rate RS code. In section 6.4, a matched-rate adaptive algorithm is derived and the throughput capacity of hybrid-ARQ protocols is investigated. The impact of typical time-varying channel conditions is discussed in section 6.5, followed by some concluding remarks in section 6.6.

6.2 Description of the Proposed Protocol

The system under consideration is illustrated by the block diagram shown in Figure 6.1. Adaptive forward error correction is concatenated with an ARQ protocol. The CCITT CRC-32 [97] is used for error detection and an RS code is used for error correction. RS codes are chosen because they can provide a wide range of code rates and so they are good candidates to implement an adaptive error control protocol. It is also known that RS codes make highly efficient use of redundancy and that efficient decoding techniques are available for use with RS codes. In addition, the weight distribution of any RS code is known, making performance evaluation analytically tractable [53].

Consider a CRC code that has a codeword length of n' symbols and an RS code that has a codeword length of n symbols. Let k_i be the number of information symbols after the ith update of the error correction code rate. Assume that the symbol error probability in a feedback transmission is approximately the same as in the preceding forward transmission and denote it by p_s. The proposed protocol can then be described by the following procedure:

Step 1: CRC encoding - k_i information symbols are encoded to an (n', k_i) CRC codeword.

Step 2: RS encoding - the n' CRC codeword symbols are further encoded to an (n, n') RS codeword which is denoted by N.

Step 3: Forward transmission - on average $p_s n$ symbols in an RS codeword of length n will be in error after forward transmission. Let N^* denote the block corresponding to the RS codeword received at the receiver after the forward transmission.

Step 4: RS error correction - n symbols in N^* are first decoded by the RS decoder.

Step 5: CRC error detection - the decoded n' symbols are further checked by the CRC.

Step 6: Feedback transmission (case 1) - if no error is detected and the CRC codeword does not contain any RS parity symbols, an ACK message is sent back to the transmitter and the procedure is repeated from Step 1.

Step 6′: Feedback transmission (case 2) - if no error is detected (acknowledged) and the CRC codeword does contain RS parity symbols, the undecoded block N^* is sent back to the transmitter and the procedure is continued from Step 7.

Step 6″: Feedback transmission (case 3) - if the CRC detects an error, the undecoded block N^* is sent back to the transmitter and the procedure is continued from Step 7.

Step 7: Channel estimation - let N^{**} denote the block received at the transmitter after the feedback transmission. At the transmitter, the Hamming distance in symbols between N^{**} and N is counted. This number is denoted by $dis(N, N^{**})$ and is used to calculate the estimated channel symbol error probability, denoted by p_s^*.

Step 8: Code rate adaptation - the RS code rate is changed based on a function of p_s^*, denoted by $F(p_s^*)$. The function F is derived in section 6.4. It is designed to let the error correction capability of the RS code match the estimated channel condition.

Step 9: If the preceding codeword is not acknowledged, the procedure returns to step 1 and retransmissions are invoked - the number of information symbols is changed to obtain k_{i+1} based on the new code rate. The number of information symbols k_{i+1} is unchanged until the codeword is successfully acknowledged by the receiver.

Step 9′: If the preceding codeword is acknowledged, the procedure returns to step 1 and new transmissions are started - the number of information symbols is changed to obtain k_{i+1} based on the new code rate.

In the proposed protocol, unless the code rate is at the highest rate (CRC only) and the received block is acknowledged, the receiver sends back the whole undecoded block to the transmitter. The novel aspect of the proposed protocol is the estimation of the channel state using the information provided by feedback transmissions of whole blocks instead of ACK or NAK messages.

Block retransmissions in the proposed protocol can be based on any basic ARQ protocol such as stop-and-wait, go-back-N, or selective-repeat. For simplicity, in the following we will consider only the SR protocol. Ideal interleaving is also assumed so that the channel may be modeled by a binary symmetric channel with a time-varying bit error probability. As in an ideal SR protocol, the receiver is assumed to contain sufficient storage to save any post-NAK blocks until the block received in error is retransmitted successfully, and the logic for reinserting the block in the proper sequence. The transmitter is also assumed to have the required logic to send blocks out of sequence.

It is possible that every retransmission of an NAKed codeword in Step 9 may have a different new code rate. This would require more complex buffering at the transmitter. Since the new code rate is designed to be matched with the estimated channel condition, the probability of more than one retransmission is very low. Therefore, as long as the channel remains approximately unchanged during two round-trip transmission periods for most of the time, we can adopt the strategy of keeping the adapted code rate until a successful transmission occurs.

Also notice that in every first transmission, the number of information symbols in the first step may be different depending on the estimated channel condition during the transmission of the preceding codeword. Therefore, the initial number of information symbols (k_0) does not matter over a long term and may be simply set to have error detection only. In the following sections, this dynamic adaptive strategy is shown to provide a much improved throughput relative to comparable type-I and type-II hybrid-ARQ protocols.

6.3 Throughput Analysis

In throughput analysis, an ACK failure probability can be easily introduced to consider the possible loss of ACK messages under poor channel conditions [26] [120] [121]. Since the proposed protocol is to be used in applications where the feedback channel is not fully utilized, we may assume that ACK messages can be heavily coded. The loss of ACK messages will therefore not be considered in the following analysis. Modulation is also assumed to provide the required block synchronization.

Consider that the RS code employed in the proposed protocol has a codeword length of $n = 256$ symbols. The RS code is defined over $GF(2^m)$. Each symbol contains m bits ($m = 8$). CRC-32 has 32 redundancy bits so that the CRC codeword length is $n' = k + 4$ symbols or $n' = 8k + 32$ bits, where k is the number of information symbols in a codeword. Notice that the $(256, n')$ RS code is in fact a lengthened RS code obtained by adding an information symbol to the $(255, n')$ RS code. After lengthening, the minimum distance (d) remains unchanged and $d = n - k + 1$.

The m bit symbols are transmitted sequentially over a BSC with time-varying channel bit error probability p_b. The receiver performs symbol regeneration prior to the RS decoder. By bit interleaving over a sufficiently long period, the memoryless channel assumption can be fulfilled for mobile radio channels. For clarity, we have used the equivalent channel symbol error probability, denoted by p_s, which is given by

$$p_s = 1 - (1 - p_b)^m. \tag{6.1}$$

For a forward transmission immediately followed by a feedback transmi[illegible] due to poor channel conditions, the real channel will probably have the same poor condition for transmissions in both directions, because of the short time period involved. We therefore consider a channel with the following time-varying statistics:

- With probability ν, the channel condition at the beginning of a codeword transmission is unchanged relative to the preceding codeword;

- With probability $1 - \nu$, the channel condition at the beginning of a codeword transmission is changed relative to the preceding codeword;

- The channel condition during the short period of a codeword transmissions, including both forward transmissions and feedback transmissions, is assumed to be constant.

Suppose that each block contains exactly one RS codeword of length n symbols. A codeword is then simply an encoded data block. To derive the analytical result for throughput, P_E, the probability of block retransmission, needs to be determined first.

As defined in Chapter 3, P_{ED} is the probability of decoding failure of an RS code. A bounded distance decoder based on the Berlekamp-Messey algorithm is used to implement the RS decoder. The decoder selects as the transmitted codeword any codeword that differs from the received word in t or fewer symbols, where t is the error correction capability of an RS code. If no such codeword exists, a codeword error is detected. If the received block is within distance t of an incorrect codeword, a decoding error is committed.

Again let P_{CD} and P_{ICD} be the probabilities of correctly decoding and incorrectly decoding into another valid codeword, respectively. Equations (3.63) to (3.69), the analytical results of RS codes, can then be readily applied. For a Reed-Solomon code of length 256 symbols with the capability of correcting up to t symbol errors, the typical values of P_{CD} and P_{ICD} are shown in Figure 6.2. Notice that the effect of P_{ICD} can not be neglected when channel conditions are worse than 10^{-2}.

Let P_c be the probability that a received block contains no symbol errors. We have

$$P_c = (1 - p_s)^n. \tag{6.2}$$

Let P_e be the probability that a CRC codeword of length n' symbols after RS decoding contains an undetectable error pattern. For CRC-32, it is known that P_e can be estimated by using the following bound [68],

$$P_e \leq 2^{-32}[1 + (1 - 2p_b)^{8k+32} - 2(1 - p_b)^{8k+32}] \leq 2^{-32} \approx 10^{-10}, \tag{6.3}$$

for $0 \leq p_b \leq 0.5$, where $8k+32$ is the length of a CRC-32 codeword in bits. Because P_e is negligible, CRC-32 can be considered to provide ideal error detection.

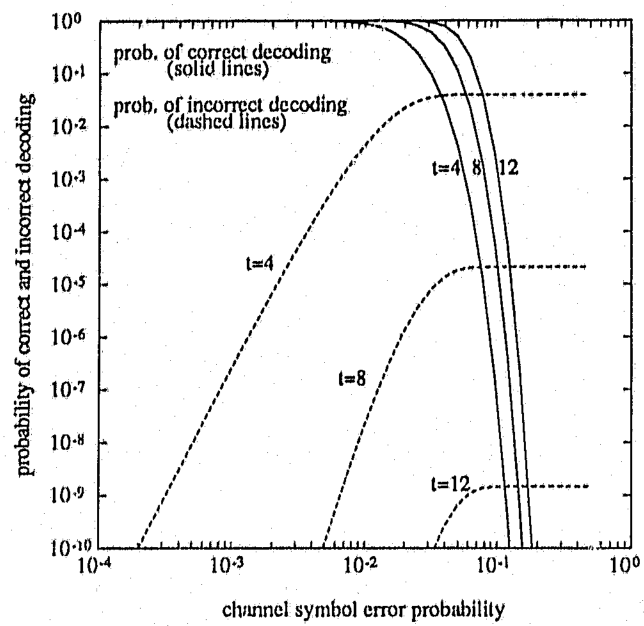

Figure 6.2: The relative values of P_{CD} and P_{ICD}

We now consider the worst case, where the channel condition is always unmatched with the code rate in the first transmission ($\nu = 0$). The more general case ($\nu > 0$) will be considered in section 6.5. Consider any point in time and let P'_{ED} and P_{ED} be the RS decoding failure probability evaluated before and after the code-rate adaptation, respectively. For a codeword to be successfully accepted by the receiver, the average number of transmissions needed is

$$\begin{aligned}
T_{match} &= 1 \cdot P'_{ED} + 2 \cdot (1 - P'_{ED})\frac{n}{n-2t}(1 - P_{ED}) + 3 \cdot (1 - P'_{ED})\frac{n}{n-2t}P_{ED}(1 - P_{ED}) + \cdots \\
&+ \; l \cdot (1 - P'_{ED})\frac{n}{n-2t}P_{ED}^{l-2}(1 - P_{ED}) + \cdots \\
&= P'_{ED} + \frac{n}{n-2t}\frac{1 - P'_{ED}}{P_{ED}}[2P_{ED}(1 - P_{ED}) + 3P_{ED}^2(1 - P_{ED}) + \cdots + lP_{ED}^{l-1}(1 - P_{ED}) + \cdots] \\
&= P'_{ED} + \frac{n}{n-2t}\frac{1 - P'_{ED}}{P_{ED}}\left[\frac{1}{1 - P_{ED}} - (1 - P_{ED})\right]. \qquad (6.4)
\end{aligned}$$

Hence, the throughput of the proposed protocol is

$$\eta_{match} = \frac{1}{T_{match}} \cdot \left(\frac{8k}{8k+32}\right). \tag{6.5}$$

In the above result, notice that k and t are determined prior to each transmission based on the ACKed or NAKed block received at the transmitter. It is this dynamic strategy of the new protocol that results in a better match with the time-varying channel conditions.

Since most real channels are in good condition most of time, the first transmission of a data block may frequently be made with error detection only. For this important special case, we can simply replace P'_{ED} in the above results by P_c and obtain an expression for the throughput

$$\eta_{match} = \left\{P_c + \frac{n}{n-2t}\frac{1-P_c}{P_{ED}}\left[\frac{1}{1-P_{ED}} - (1-P_{ED})\right]\right\}^{-1} \cdot \left(\frac{8k}{8k+32}\right). \tag{6.6}$$

As a comparison, we can analyze the throughput performance of the selective-repeat type-II hybrid-ARQ protocol using a half-rate RS code of the same block length. This protocol is the same as the protocol proposed by Wang and Lin [111] except that an RS code is used as the half-rate invertible code instead of a binary $(2n, n)$ invertible code.

Suppose that CRC-32 is also used for error detection and a (256, 128) RS code is used with the capability of correcting up to t symbol errors ($t = 64$). The number of information symbols is equal to 124 because 4 symbols are reserved for CRC bits.

Through a straightforward application of the results in [111], the average number of transmissions needed to successfully transmit a codeword can be found as

$$T_{typeII} = P_c + 2(1-P_c)P_t, \tag{6.7}$$

where P_t is the conditional probability that the receiver recovers a data block (first half of the $(256,128)$ RS codeword) after receiving a parity block (second half of the $(256,128)$ RS codeword).

With P_c given by (6.2), and $n = 128$, P_t can be found as

$$P_t = P_c + (1 - P_c)\frac{q_0 - y}{1 - y}, \tag{6.8}$$

where

$$q_0 = \sum_{j=0}^{t} \binom{2n}{j} p_s^j (1 - p_s)^{2n-j}, \tag{6.9}$$

and

$$y = (1 - p_s)^n \left[2\sum_{l=0}^{t} \binom{n}{l} p_s^l (1 - p_s)^{n-l} - (1 - p_s)^n \right]. \tag{6.10}$$

Substituting (6.8) into (6.7), we obtain the following result for the average number of transmissions needed to successfully transmit a codeword using the type-II hybrid protocol:

$$T_{typeII} = P_c + 2(1 - P_c)(P_c + (1 - P_c)\frac{q_0 - y}{1 - y}). \tag{6.11}$$

Hence, the throughput of the type-II hybrid-ARQ protocol is

$$\eta_{typeII} = \frac{1}{T_{typeII}} \cdot \left(\frac{8k}{8k + 32} \right). \tag{6.12}$$

Notice that the number of information symbols in the type-II hybrid protocol is fixed. Only two code rates are available for use in response to the channel conditions. Regardless of the length of time poor channel conditions persist, the type-II hybrid-ARQ protocol always starts with the high code rate in the first transmission.

6.4 The Matched-Rate Adaptive Algorithm

As described in section 6.2, the proposed protocol sends back the received block through the feedback channel for the purpose of channel estimation. Now we need to find the new error correction capability t by estimating the channel symbol error probability p_s whenever the channel condition changes.

In the proposed protocol, the channel condition may be different each time a received block is sent back to the transmitter. If the block has already been retransmitted once or more, the newly calculated t is simply discarded. Otherwise, the new t is used to modify the code rate for subsequent transmissions, regardless of the previous code rate.

For an estimated channel symbol error probability p_s^*, one can expect that on average $p_s^* n$ symbols will be in error after a forward transmission. If the length of codewords is not too short, say $n = 256$, as in our case, a good estimate of p_s can be obtained by

$$p_s^* \approx \frac{dis(N, N^*)}{n}. \tag{6.13}$$

The new code rate clearly depends on the channel condition characterized by p_s^*. We need to find out how to estimate the value of p_s^* at the transmitter side. Although n and N are known, N^* is not known at the transmitter.

Notice that when a received block is sent back to the transmitter, we also have

$$p_s^* \approx \frac{dis(N^*, N^{**})}{n}. \tag{6.14}$$

A block which has experienced both a forward transmission and a feedback transmission will have a concatenated channel symbol error probability, denoted by p'_s, equal to

$$p'_s = 1 - (1 - p^*_s)(1 - p^*_s). \tag{6.15}$$

We then have

$$dis(N, N^{**}) = n[1 - (1 - p^*_s)(1 - p^*_s)] = np^*_s(2 - p^*_s). \tag{6.16}$$

Solving the above equation for p^*_s, we obtain

$$p^*_s = 1 - \frac{1}{n}\sqrt{n^2 - n \cdot dis(N, N^{**})}. \tag{6.17}$$

Hence, the new code rate can be found by adapting the error correction capability t based on a function of p^*_s, that is

$$t = F(p^*_s) = F(1 - \frac{1}{n}\sqrt{n^2 - n \cdot dis(N, N^{**})}). \tag{6.18}$$

We need to determine the function F in the above algorithm. Intuitively, one may think that the error correction capability t should be equal to the estimated average number of symbol errors in a codeword. This is in fact incorrect. In order to clarify this point and find the correct form for the function F, let us examine the throughput performance of a hybrid-ARQ protocol which uses an RS code with fixed error correction capability, also known as a type-I hybrid-ARQ.

Notice first that the throughput of an ideal SR ARQ protocol using CRC-32 is given by [68]

$$\eta_{SR} = P_c \cdot \left(\frac{8k}{8k + 32}\right). \tag{6.19}$$

By using the analytical results presented in section 6.3, the throughput of a type-I hybrid-ARQ protocol using an RS code can be readily derived. Notice that the average number of transmissions needed to transmit a codeword is

$$\begin{aligned} T_{typeI} &= 1\cdot(P_{CD}+P_{ICD})+2\cdot P_{ED}(P_{CD}+P_{ICD})+3\cdot P_{ED}^2(P_{CD}+P_{ICD})+\cdots \\ &= \frac{1}{1-P_{ED}}. \end{aligned} \tag{6.20}$$

The throughput of a type-I hybrid-ARQ protocol is thus

$$\eta_{typeI}=(P_{CD}+P_{ICD})\cdot\left(\frac{8k}{8k+32}\right). \tag{6.21}$$

The throughputs of a type-I hybrid-ARQ protocol with error correction capabilities $t=26$ and $t=36$ is shown in Figure 6.3. Also shown is the throughput of the ideal SR ARQ protocol as a comparison. At a channel symbol error probability of 10^{-1}, one may expect on average to have 26 symbols in error in a code of length 256 symbols. If the code rate is mis-matched in such a way that $t = ceil(np_s^*)$, where the function $ceil(x)$ is defined as the smallest integer which is equal or greater than x, it can be seen that its throughput is much worse than the protocol using $t=36$ at the same channel condition of 10^{-1}. Therefore, the coding rate should be determined by the criteria of maximum throughput instead of by matching the expected number of symbols in error.

In order to determine the function F, consider an (n,k) RS code defined over $GF(2^8)$ that is used for t-error correction. A received block will be decoded correctly if it contains t or fewer symbol errors. That is, if the received block is contained within a radius-t sphere surrounding the transmitted codeword, decoding will be successful. However, the probability of u symbol errors occurring in an RS codeword of length n, denoted by P_u, is

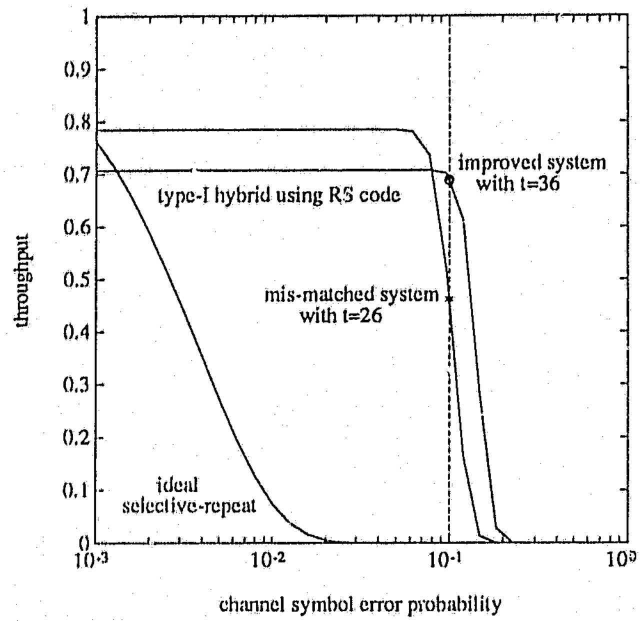

Figure 6.3: Throughput of a mis-matched protocol and an improved type-I hybrid-ARQ protocol, where CRC codeword length is 260 symbols and RS codeword length is 256 symbols

$$P_u = \binom{n}{u} p_s^u (1-p_s)^{n-u}. \tag{6.22}$$

For the given channel symbol error probability 10^{-2}, the probability distribution of u symbol errors occurring in an RS codeword of length 256 symbols is shown in Figure 6.4. For different channel conditions, the envelope of the probability distribution of k symbol errors occurring in an RS codeword is shown in Figure 6.5. We can see that the probability that the received codeword contains more than np_s symbols in error, denoted by P_{np}, cannot be neglected. We have

$$P_{np} = \sum_{u=ceil(np_s)}^{n} \binom{n}{u} p_s^u (1-p_s)^{n-u}. \tag{6.23}$$

In general, the error correction capability of an RS code is proportional to

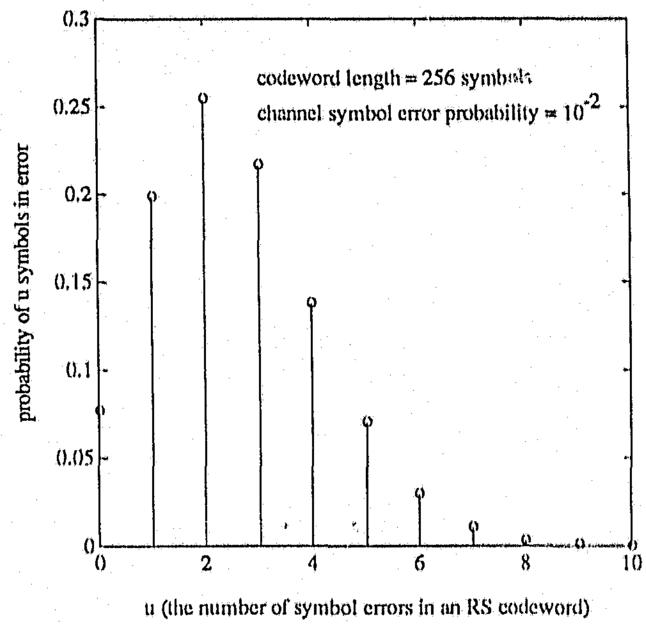

Figure 6.4: Probability distribution of symbol errors in an RS codeword

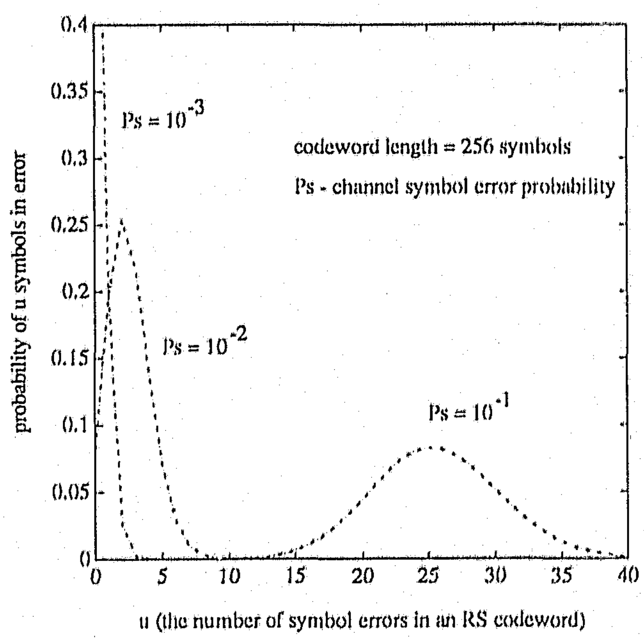

Figure 6.5: The envelopes of symbol error probability distributions

the number the redundant parity symbols, but these parity symbols reduce the throughput by increasing the overhead. Therefore, more error correction capability may not lead to better throughput. The code rate to match with the estimated channel condition should be determined such that the following ratio is maximized

$$\max_{0 \le t \le \frac{d-1}{2}} \left\{ \frac{\text{throughput increase due to the adapted error correction capability}}{\text{throughput decrease due to the overhead of the adapted code}} \right\} \quad (6.24)$$

In other words, matched-rate adaptive error control is achieved by maximizing the throughput under various channel conditions.

Recall that the throughput of the proposed protocol is determined by both P_{CD} and P_{ICD}, given by (3.64) and (3.65), respectively. The analytical optimization of the above ratio is difficult to carry out, if not impossible. However, there is an easier way to find the adaptive function F. In Figure 6.6, the throughputs of a type-I hybrid-ARQ protocol using an RS code with different error correction capabilities are shown. Notice that when $t = 0$ the type-I hybrid-ARQ protocol becomes the ideal SR ARQ. It can be seen that throughput can be maximized if the new code rate is chosen such that the throughput of the adapted protocol is always located on the envelope of all possible throughputs provided by the type-I hybrid-ARQ protocol at the estimated channel condition. This envelope may be called the *ideal matched-rate throughput.*

Of course, one may numerically calculate all possible throughput curves of a type-I hybrid protocol and obtain a set of data which approximates the envelope and thus can be used for the calculation of the new code rate. Analytically, we have also obtained an empirical formula which is a good approximation of the required function F for adapting code rates. The function is given by

$$t = F(p_s^*) = ceil(np_s^* + \gamma_0(1 - e^{-np_s^*})) = ceil(np_s^* + 5.5(1 - e^{-np_s^*})), \quad (6.25)$$

where the constant γ_0 has been determined experimentally for the proposed protocol. For any given channel symbol error probability, the empirical formula has

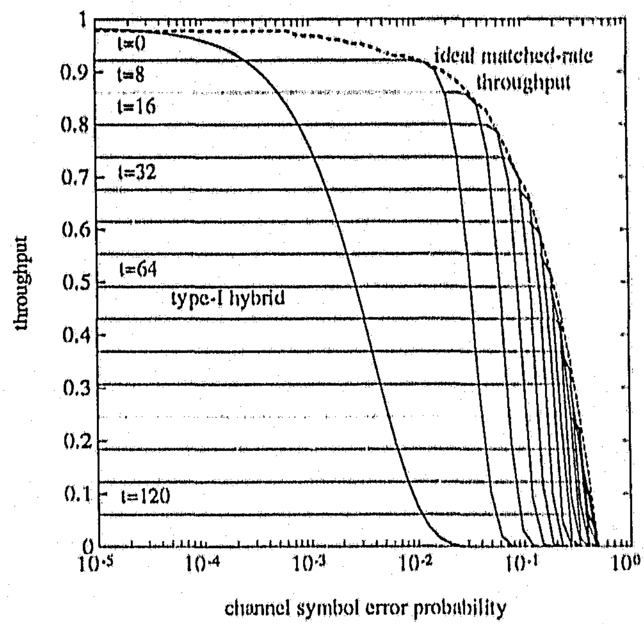

Figure 6.6: Ideal matched-rate throughput observed from the maximum throughput envelope

been found to yield a calculated code rate which provides a throughput very close to the ideal matched-rate throughput.

6.5 Numerical Results and Discussion

In this section, numerical examples concerning the throughput for different time-varying channel conditions are presented. As a comparison, the throughputs of the type-I and type-II hybrid-ARQ protocols using an RS code of the same block length will also be presented. The $(256, n')$ t-error-correcting RS code and CRC-32 are assumed to be used in all these protocols. As in the previous sections, the throughput of the ideal selective-repeat ARQ protocol is also shown as a reference.

The throughput performance of two typical type-I hybrid-ARQ protocols and a type-II hybrid-ARQ protocol with a half-rate RS code are shown in Figure 6.7. It can be seen that the performance of the type-II hybrid-ARQ protocol is better than that of the type-I hybrid-ARQ protocol when the channel is in a very good

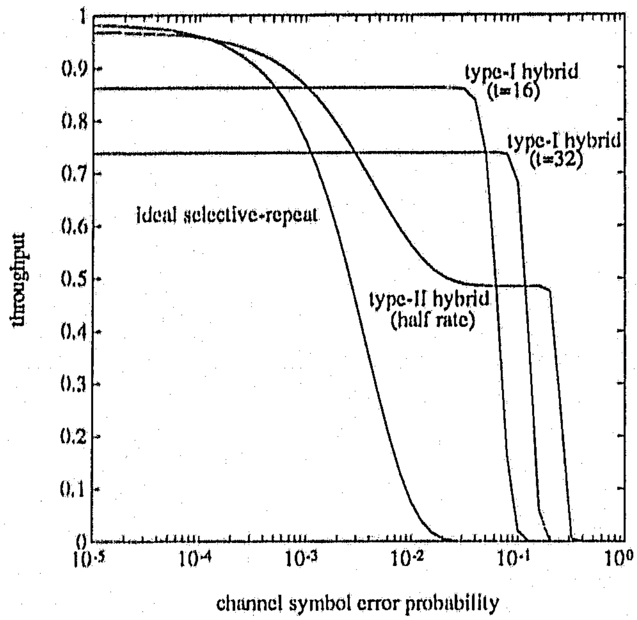

Figure 6.7: Throughput performance of type-I and type-II hybrid-ARQ protocols, where CRC codeword length is 260 symbols and RS codeword length is 256 symbols

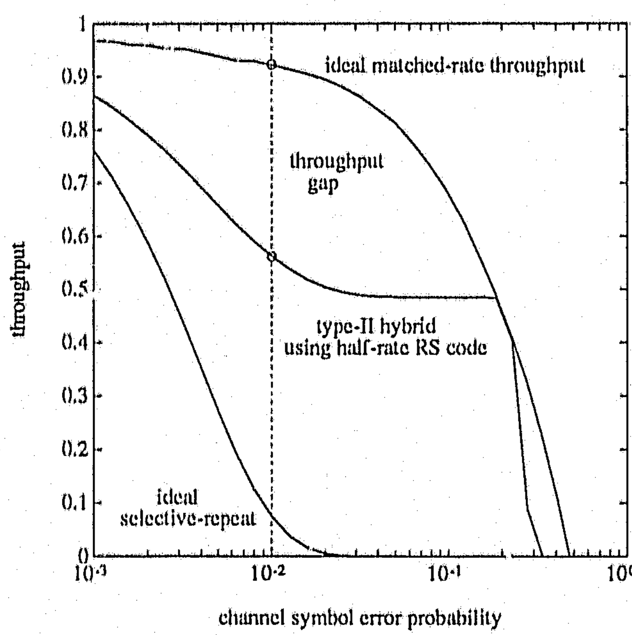

Figure 6.8: Throughput of the type-II hybrid-ARQ protocol compared with the ideal matched-rate throughput

($< 10^{-3}$) or in a very poor ($> 10^{-1}$) condition. For a channel with a symbol error probability between 10^{-1} and 10^{-3}, the performance of a type-I hybrid-ARQ protocol is better. As we have explained in the previous sections, this is due to the matching of the error correction capability with the channel conditions.

Now let us have a further look at the case of type-II hybrid-ARQ protocols. The throughput performance of a type-II hybrid-ARQ protocol is shown in Figure 6.8, permitting comparison with the ideal matched-rate throughput performance. It can be seen that under most channel conditions, the code rate of a type-II hybrid-ARQ protocol using a half-rate code is not matched with the channel conditions. In other words, there exists a significant throughput gap between the type-II hybrid-ARQ protocol performance and the ideal performance.

Is it possible that the proposed protocol can approach the ideal matched-rate throughput? The answer is yes. In section 6.3, we derived an expression for the throughput of the proposed protocol under the worst channel conditions, which occur when the channel is always unmatched with the code rate in the first transmission. Now consider the ideal situation, in which the code rate is always matched with the channel condition. In this case the formula to calculate the throughput of the proposed protocol is the same as the result for a type-I hybrid-ARQ protocol, which is given by

$$\eta_{match} = (P_{CD} + P_{ICD}) \cdot \left(\frac{8k}{8k + 32} \right), \tag{6.26}$$

except that the values of P_{CD} and P_{ICD} are obtained based on the different code rate calculated by using the adaptive formula (6.25) for each channel symbol error probability. The throughput of this ideal matched-rate adaptive protocol is very close to the ideal matched-rate throughput. In practice, although the ideal matched-rate adaptivity may not be achieved, we may reasonably expect that the real throughput performance of the proposed protocol will be between the ideal case and the worst case. For example, let us consider the situation in which the code rate of the proposed protocol is matched with the channel condition 90% of the time, and is completely unmatched, as considered in the worst case, 10% of the time. The throughput performance under these conditions is shown in Figure 6.9. It can be seen that the throughput performance of the proposed protocol approaches the ideal matched-rate performance when the matching percentage increases. The actual percentage depends on the time-varying channel condition. In general, the throughput of the proposed protocol may be described as

$$\begin{aligned} \eta_{match} &= \nu \cdot (P_{CD} + P_{ICD}) \cdot \left(\frac{8k}{8k+32}\right) \\ &+ (1-\nu) \cdot \frac{1}{P_c + \frac{n}{n-2t}\frac{1-P_c}{P_{ED}}\left[\frac{1}{1-P_{ED}} - (1-P_{ED})\right]} \cdot \left(\frac{8k}{8k+32}\right), \end{aligned} \tag{6.27}$$

where ν may be called the *matching percentage*. As a comparison, if the code rate is determined to be equal to $ceil(np_s^*)$ (mis-matched), the corresponding throughputs for different matching percentages are shown in Figure 6.10. It can be seen that the mis-matched performance is worse than the performance of a system which adapts the code rate using the function given in (6.25).

In comparison to other known adaptive error control protocols, such as the protocol presented in [100], the proposed protocol is able to provide faster channel estimation. The reason is simply that the proposed protocol can estimate the time-varying channel condition by comparing the corrupted codeword and the original codeword at the transmitter after every round-trip transmission time period. In other words, the channel condition is estimated by counting the error symbols in a block rather than the number of blocks.

In comparison to a type-II or a generalized type-II hybrid-ARQ protocol, the proposed protocol could also provide faster code-rate adaptation when a poor channel condition persists. In the proposed protocol, the code rate in the first transmission of any data block is determined by the estimated channel condition in the preceding time period, while in a type-II hybrid-ARQ protocol, the initial code

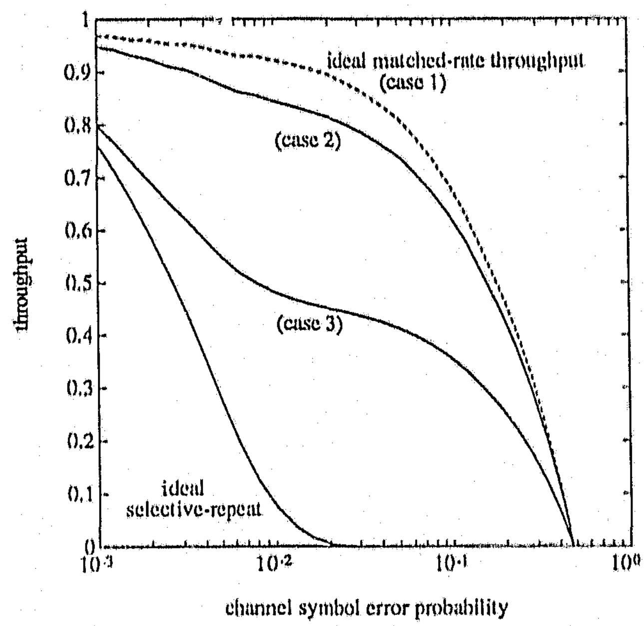

Figure 6.9: Performance of the proposed protocol for different time-varying channel conditions — $\nu = 1$ (case 1); $\nu = 0.9$ (case 2); $\nu = 0$ (case 3)

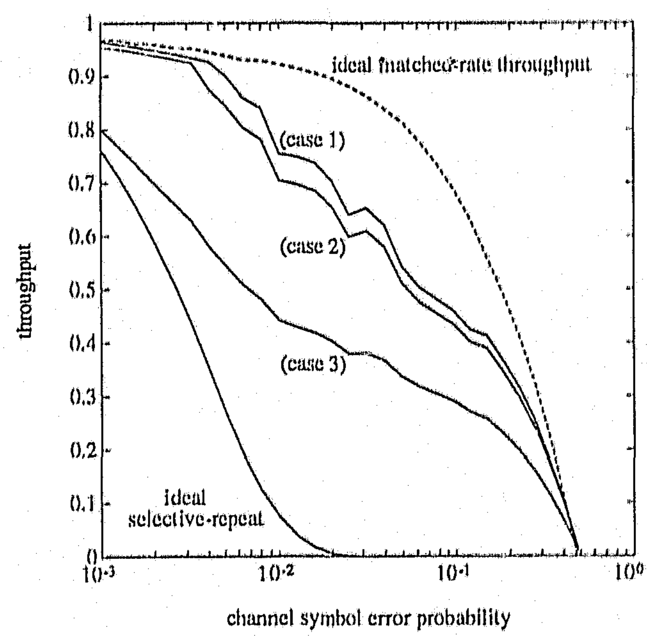

Figure 6.10: Performance of the mis-matched protocol for different time-varying channel conditions — $\nu = 1$ (case 1); $\nu = 0.9$ (case 2); $\nu = 0$ (case 3)

rate is always the highest rate (error detection only). This dynamic adaptation strategy has been shown in Figure 6.8 to provide better throughput efficiency than a conventional type-II hybrid-ARQ protocol.

6.6 Summary

The ideal matched-rate throughput achievable by a hybrid-ARQ error control protocol has been identified in this chapter. A matched-rate adaptive error control protocol is proposed to reduce the throughput gap between the ideal performance and that of a type-II hybrid-ARQ protocol using a half-rate Reed-Solomon code. It is shown that significant throughput improvement can be obtained by using the proposed protocol. In particular, under certain time-varying channel conditions, an empirical algorithm indicates that the throughput of the proposed protocol approaches the ideal matched-rate throughput.

To the best of our knowledge, this is the first time that the dynamic adaptation of error correction coding using feedback transmissions has been presented. Although our new protocol and its throughput estimates are discussed for a time-varying BSC only, and may appear a little empirical, we have shown the possible performance improvement offered by the new protocol.

Our discussion on the new protocol with feedback transmissions has been based on the hybrid-ARQ protocol using Reed-Solomon codes. However, the concept of matched-rate adaptive error control coding can be readily integrated into protocols using other error correction codes. For example, it may also be applied to a hybrid-ARQ error control protocol with punctured convolutional codes [44] to achieve the matched-rate condition.

Chapter 7

Numerical Optimization of Coding Parameters

7.1 Introduction

The previous chapter on matched-rate adaptive error control coding has been based on the idea of type-II hybrid-ARQ protocols. In applications where channel statistics are relatively time invariant, a properly designed type-I hybrid-ARQ protocol with a t-error-correcting code is often a more cost-effective choice. One of the most important issues in investigating type-I hybrid protocols is to choose a proper code since crossover points for throughput efficiency exist among plain ARQ, type-I and type-II hybrid-ARQ, as shown in Figure 7.1.

Research regarding the optimum block length has already appeared in the literature [18]. In this chapter, the optimum code for error correction and the optimum block length in type-I hybrid-ARQ protocols are investigated by a pragmatic optimization method. Two new criteria called "overall throughput" and "modified throughput" are proposed to measure the performance capability in order to carry out optimizations. The optimum system parameters for a type-I hybrid-ARQ protocol with BCH error correcting codes and RS codes are obtained by numerical optimization. The optimum design offers the local maximum of overall throughput performance with the least system complexity.

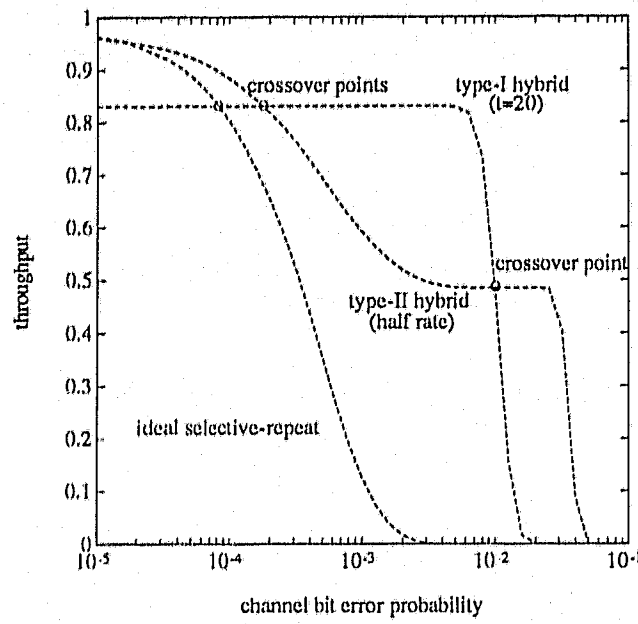

Figure 7.1: Throughput performance of hybrid error control protocols, where CRC codeword length is 260 symbols and RS codeword length is 256 symbols

7.2 Channel Statistics and Error Measurement

Two basic performance measures of a hybrid-ARQ error-control system are reliability and throughput efficiency. Reliability is determined by the error detecting code used in the system. The optimum cyclic redundancy codes for error detection have been examined in the literature [79]. In this chapter we consider error correcting codes only. In order to carry out the system optimization, we need to form a composite performance criterion which reflects the manner in which each system parameter influences the whole. Throughput efficiency provides such a possibility. But the throughput of ARQ or hybrid-ARQ protocols depends on the channel characteristics. As shown in Figure 7.1, the throughput of plain ARQ and type-I hybrid-ARQ is related to the channel bit error probability. Without involving channel error statistics, we cannot tell whether type-I hybrid-ARQ is better than plain ARQ or not.

In practice, we must deal with data from the real world. The error statistics on many real channels are time-dependent. This leads to the design of adaptive error control systems [73]. But when real systems are measured under real operating conditions in the field, the errors are found to arrive mostly in bursts. Often adaptation of code rates cannot catch up with many short burst changes. Therefore, channel error measurement must be considered in the evaluation of performance in a practical system.

One parameter used to describe the performance of a digital system is the bit error probability, i.e. the probability of the incorrect reception of a single bit. Experimentally, the most often used parameter is the bit error rate (BER). This is defined as [15]

$$BER = N_e/N_t = N_e/(r_b t_0), \tag{7.1}$$

where

N_e = the number of bit errors in the time interval t_0 ;

N_t = the number of transmitted bits in the time interval t_0 ; and

r_b = the bit rate of a binary signal at the point where the measurement is performed.

Where the error generation process is random and stationary and the errors are counted in a sufficiently long interval t_0, equation (7.1) can give an estimate of the error probability. The accuracy of the estimate increases as N_e increases. If there are only a few bit errors in the time interval t_0, i.e. for small N_e, the measuring time interval t_0 becomes inordinately large if N_e is to be sufficiently large for reasonably good estimation. However, practical requirements on the measuring time interval usually limit the values which can be obtained for N_e. The minimum acceptable value of N_e seems to be about 10, in which case the true error probability is contained in a range equal to ±50 percent of N_e/N_t with a confidence coefficient of 90 percent [105].

For the purpose of system design and production testing, error performance measurement is tested under out-of-service conditions, i.e. a digital transmission

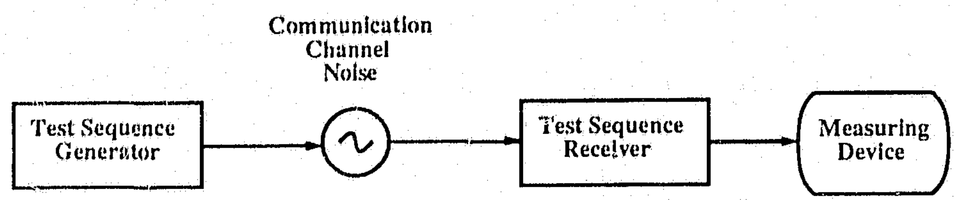

Figure 7.2: Block diagram of bit error rate measurement

link is taken out of revenue or monitoring service. The test is done by inserting a test pattern into the input of the link, which simulates the normal traffic data stream. The output of the link is compared bit by bit with a locally generated error-free reference pattern in an error detector. The test pattern is usually a pseudo-random binary sequence which has a repetition period of $2^n - 1$ generated by a shift-register circuit. To reliably measure the inter-symbol interference effects, the register length n should be greater than, or equal to, the number of pulses which are affected by the channel response to a single pulse. As this requirement depends on characteristics of the transmission channel which are often unknown, a conservative value of n is usually chosen. To check the suitability of the test sequence, the error rates observed with different register lengths can be compared.

One method to obtain the BER is to use a fixed-length test sequence. Bit error rate measurement is performed according to the scheme in Figure 7.2. The generator produces a test sequence of r bits. After transmission through the communication channel the test sequence is received at the receiver. Comparing the received sequence and the locally generated test sequence (which are synchronized) we find the number of errors k. The ratio k/r is the bit error rate which is shown on the display of the measuring device.

Consider the results of a BER measurement taking the following form:

BER between 10^{-7} and 10^{-6} for x_1 percent of the time;

BER between 10^{-6} and 10^{-5} for x_2 percent of the time;

BER between 10^{-5} and 10^{-4} for x_3 percent of the time;

BER between 10^{-4} and 10^{-3} for x_4 percent of the time;

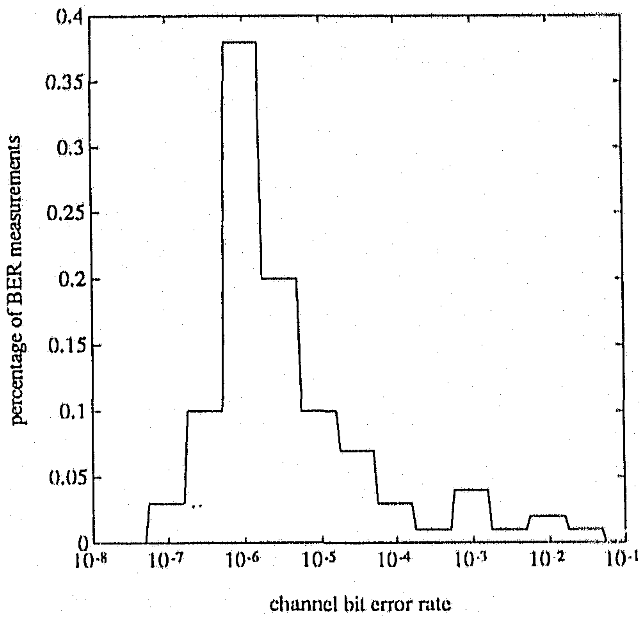

Figure 7.3: The results of a typical bit error rate measurement

BER between 10^{-3} and 10^{-2} for x_5 percent of the time;

BER between 10^{-2} and 10^{-1} for x_6 percent of the time;

BER within other ranges for 0 percent of the time.

Clearly, we have $\sum_{i=1}^{6} x_i = 1$. An example of the possible measurement results of the BER, which may reflect daily BER fluctuations of a telephone trunk [107], is as follows:

$x(10^{-7}) = 0.03$; $x(10^{-6.5}) = 0.1$; $x(10^{-6}) = 0.38$; $x(10^{-5.5}) = 0.2$;

$x(10^{-5}) = 0.1$; $x(10^{-4.5}) = 0.07$; $x(10^{-4}) = 0.03$; $x(10^{-3.5}) = 0.01$;

$x(10^{-3}) = 0.04$; $x(10^{-2.5}) = 0.01$; $x(10^{-2}) = 0.02$; $x(10^{-1.5}) = 0.01$;

$x(other\ ranges) = 0$.

Here $x(10^{-7})$ is the percent of the time that the BER is between $10^{-7.25}$ and $10^{-6.75}$, and so on. The results of this typical BER measurement are shown in Figure 7.3.

7.3 Performance Criteria for Optimization

Let us introduce a new performance criterion called *overall throughput* η_o defined as follows:

$$\eta_o = \int_0^1 x(p_b)\eta(p_b)\, dp_b, \tag{7.2}$$

where p_b is the channel bit error probability, $x(p_b)$ is the bit error probability density function, and $\eta(p_b)$ is the throughput density function. In order to make the optimization numerically tractable, the following definition in discrete form is used:

$$\eta_o = \sum_i x(p_{bi})\eta(p_{bi}). \tag{7.3}$$

Compared with the conventional definition of throughput efficiency, which is a function of channel bit error rate, the overall throughput provides a better description of a system for the purposes of evaluation of a real hybrid error control system.

Based on the definition of overall throughput, the performance optimization problem of any type of error control protocol is of the form:

$$maximize \sum_i x(p_{bi})\eta(p_{bi}). \tag{7.4}$$

The objective of any optimization exercise is to select a vector of system parameters $Y = [y_1, y_2, ..., y_n]$ in such a way that some measure of system quality $Q(Y)$ is optimized. Overall throughput η_o provides such a proper measure of system quality for ARQ or hybrid-ARQ protocols. In the following section, numerical optimization determines the optimum system parameters for any real channel condition which is obtained by the BER measuring method described above. We call it a pragmatic approach since the results are based on the practical BER measurement.

The performance of hybrid-ARQ protocols, as well as any of the possible variations, is governed by the interrelationship between several key parameters, including error rate, data rate, channel propagation delay, block length, required throughput efficiency, and error correcting capability of the code used.

7.4 Optimum Overall Throughput

7.4.1 Design of Hybrid Error Control with BCH Codes

We denote the bit error rate as p_b, the data rate as R_d, the propagation delay as T_p, and the block length as $n+h$, where h is the block overhead used for the purpose of error-detection, block synchronization, and other necessary control functions, and n is the length of the t-error-correcting inner code in a type-I hybrid-ARQ protocol. For given values of R_d, T_p, and h, and the channel bit error rate measurement, the optimization problem is to find optimal values for n and t which maximize the overall throughput with the least implementation complexity.

The inner code of the type-I ARQ protocol uses a binary BCH code with the following parameters:

code length:	$n = 2^m - 1$;
number of parity-check digits:	$n - k \leq mt$;
minimum distance:	$d_{min} \geq 2t + 1$.

This code is capable of correcting any combination of t or fewer errors in a received codeword. The BCH codes are defined by the generator polynomial $g(X)$ which is specified in terms of its roots from the Galois field $GF(2^m)$ [12]. In order to carry out the numerical computation, we have to change the inequalities in the above definition to equalities. Fortunately, for systems with reasonable implementation complexity, we are restricted to small values of t (e.g., less than 10). By doing a computer search, it can be found that for $n = 511$ or greater, $n = 2^m - 1$ and $n - k = mt$ as long as $t \leq 16$. In practice, the maximum throughput normally occurs with n greater than 511.

The hybrid error control can be implemented using various protocols. If the GBN protocol is used, the last N blocks in the buffer are retransmitted when one retransmission is requested. The throughput of a GBN protocol can be found as follows:

$$\eta = \left(\frac{n}{n+h}\right) \frac{1 - P_E}{1 + (a-1)P_E}, \tag{7.5}$$

where the block error probability P_E is

$$P_E = 1 - (1 - p_b)^{n+h}, \tag{7.6}$$

and the parameter a is the round-trip delay in blocks. In the following, if we assume that the decoder processing time is equal to the transmission period of three blocks, then $a = 3 + 2T_p R_d/(n + h)$ [118].

If a selective-repeat protocol is used, only the block identified as being in error is retransmitted. The throughput is

$$\eta = \left(\frac{n}{n+h}\right)(1 - P_E), \tag{7.7}$$

where P_E is given by equation (7.6).

By using the above formulas, it is easy to show that a go-back-N type-I hybrid-ARQ protocol with BCH codes has a throughput given by

$$\eta = \left(\frac{n - mt}{n+h}\right) \frac{1 - P_E}{1 + (a-1)P_E}, \tag{7.8}$$

where P_E is

$$P_E = 1 - (1 - p_b)^{n+h} - \sum_{i=1}^{t} \binom{n+h}{i} p_b^i (1 - p_b)^{n+h-i}. \tag{7.9}$$

If the selective-repeat protocol is used in a type-I hybrid-ARQ, then the throughput can be found as

$$\eta = \left(\frac{n - mt}{n + h}\right)(1 - P_E), \tag{7.10}$$

where P_E is given by equation (7.9).

The numerical optimization is carried out by

$$maximize \sum_i x(p_{bi})\,\eta(n, t, p_{bi}|\,given\; R_d, h, T_p) \tag{7.11}$$

$$subject\ to: \quad n > 63 \quad and \quad t < 12\ ,$$

where the system constraints are determined by implementation complexity issues as well as throughput efficiency. From the engineering point of view, it is necessary to find the maximum overall throughput with both n and t as small as possible. With system constraints, only a local maximum may be found.

For the optimization problem addressed, it is difficult to use the well-known nonlinear optimization algorithms such as the *Fletcher-Reeves Method* [72], since line search steps may not be determined in this case. Actually, based on the knowledge related to the problem, we find only limited computation is required to obtain the optimum result.

Our computation results are based on the assumptions that the feedback channel is noiseless, a typical satellite link is used which has a propagation delay $T_p = 250$ ms and the error statistics measurement results given in section 7.2, the block overhead h is 48 bits, and the data rate R_d is 4800 bps.

Some of the computation results are shown in Table 7.1 and Table 7.2. In order to locate the optimum point, the contour lines of the overall throughput with a GBN protocol are plotted in Figure 7.4.

We see in the contour plot that there exists a local maximum point, ($n = 4095$, $t = 8$), so that the optimum block length of a type-I hybrid-ARQ is about

Table 7.1: Computation results of overall throughput for GBN with BCH codes

	Overall Throughput							
	t=0	2	4	6	8	10	12	14
n=63	0.5161	0.4476	0.3474	0.2417	0.1349	0.0270	0.0811	0.1892
127	0.6582	0.6264	0.5562	0.4811	0.4030	0.3248	0.2456	0.1657
255	0.7597	0.7612	0.7178	0.6726	0.6239	0.5725	0.5211	0.4700
511	0.8186	0.8449	0.8236	0.7965	0.7699	0.7431	0.7133	0.6820
1023	0.8443	0.8865	0.8836	0.8703	0.8548	0.8386	0.8234	0.8092
2047	0.8463	0.9001	0.9102	0.9088	0.9019	0.8946	0.8861	0.8767
4095	0.8303	0.9021	0.9099	0.9190	0.9236	0.9214	0.9172	0.9133
8191	0.7977	0.8994	0.9067	0.9096	0.9139	0.9211	0.9279	0.9306
16383	0.7469	0.8882	0.9037	0.9072	0.9091	0.9098	0.9102	0.9116

Table 7.2: Computation results of overall throughput for S-R with BCH codes

	Overall Throughput							
	t=0	2	4	6	8	10	12	14
n=63	0.5497	0.4554	0.3503	0.2431	0.1351	0.0270	0.0811	0.1892
127	0.6974	0.6364	0.5617	0.4841	0.4053	0.3256	0.2457	0.1657
255	0.8005	0.7710	0.7261	0.6769	0.6264	0.5755	0.5239	0.4717
511	0.8584	0.8539	0.8298	0.8040	0.7758	0.7454	0.7144	0.6834
1023	0.8822	0.8987	0.8881	0.8733	0.8585	0.8445	0.8300	0.8142
2047	0.8849	0.9145	0.9175	0.9123	0.9050	0.8963	0.8869	0.8777
4095	0.8759	0.9109	0.9221	0.9275	0.9265	0.9233	0.9198	0.9163
8191	0.8564	0.9057	0.9110	0.9170	0.9250	0.9311	0.9333	0.9324
16383	0.8230	0.8989	0.9075	0.9100	0.9109	0.9118	0.9144	0.919[illegible]

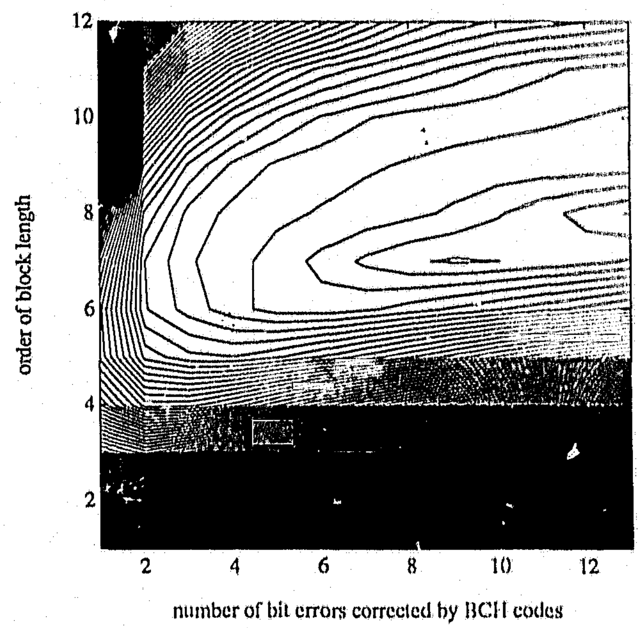

Figure 7.4: Contour plot of overall throughput for GBN with BCH Codes

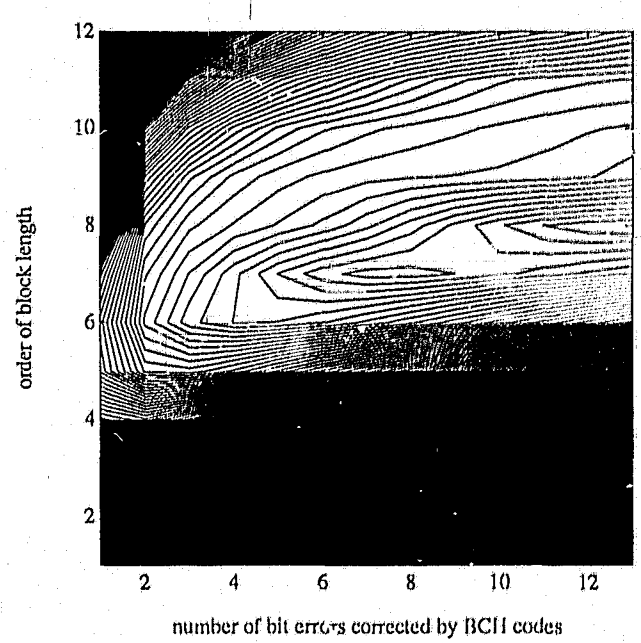

Figure 7.5: Contour plot of overall throughput for S-R with BCH Codes

twice the length of a plain ARQ [97]. In general, there may exist more local maximum points if we let n and t become large simultaneously without considering the implementation complexity. Therefore, what we found by our contour plotting are those local maximum points associated with the least complex system. The contour plot of the overall throughput for the SR-based protocol is shown in Figure 7.5, where a data rate of $R_d = 4800$ bps is also assumed. Similar results can be obtained for other data rates.

7.4.2 Design of Hybrid Error Control with RS Codes

For data communications over noisy channels such as a land mobile radio channel, the error correction capability of BCH codes is not enough to provide data transfer with high reliability and low time delay. A type-I hybrid error control protocol based on RS codes for an interleaved land mobile radio channel has been recently proposed and analyzed in [113]. It was shown that the protocol using RS codes can provide high-speed and high-reliability for voiceband data communications. However, whether or not the optimum overall throughput points can also be found for hybrid error control with RS codes is unknown and will be investigated in this section.

In practice, when a hybrid error control protocol is to be integrated in a data transmission system, extra overhead bits must be introduced to meet certain standard format. These overhead bits may include the counter, address, control and synchronization bits [96]. Therefore, we consider the case in which RS codes are used in a type-I hybrid error control protocol with extra overhead bits added. A selective-repeat protocol and an interleaved channel are assumed. The RS code in the protocol is used to correct the maximum number of symbol errors which are within the capability of the code. Error detection is performed by a CRC-16 or CRC-32 code which is concatenated with the RS code. Codeword symbols are transmitted in bit-serial format as described in Chapter 6. The noisy channel representing a mobile link is assumed to have the BER measurement shown in

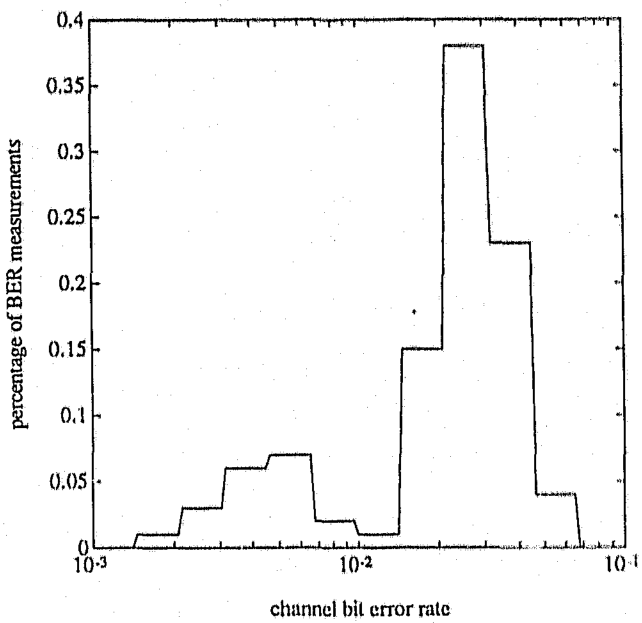

Figure 7.6: The results of a BER measurement for a noisy channel

Figure 7.6.

Let h be the number of overhead symbols in an RS codeword reserved for CRC bits. When each block contains only one codeword, the throughput formula for the SR type-I hybrid-ARQ protocol can be written as

$$\eta_{typeI} = (P_{CD} + P_{ICD}) \cdot \left(\frac{n - 2t - h}{n} \right). \tag{7.12}$$

Now consider the more general case where each block contains B codewords. Let h_B be the number of overhead symbols in a block reserved for the counter, address, control and synchronization. The block overhead is assumed to be separately coded and the block retransmissions due to overhead errors will not be considered. The probability of block retransmission is thus

$$\begin{aligned} P_{block} &= P_{ED} + (1 - P_{ED})P_{ED} + (1 - P_{ED})^2 P_{ED} + \cdots + (1 - P_{ED})^B P_{ED} \\ &= 1 - (1 - P_{ED})^B = 1 - (P_{CD} + P_{ICD})^B. \end{aligned} \tag{7.13}$$

The system throughput based on block transmissions can then be found as

$$\begin{aligned} \eta_{block} &= (1 - P_{block}) \cdot \left(\frac{B(n-2t-h)}{B\,n + h_B} \right) \\ &= (P_{CD} + P_{ICD})^B \cdot \left(\frac{B(n-2t-h)}{B\,n + h_B} \right). \end{aligned} \tag{7.14}$$

Under the ideal interleaving assumption, the channel bit error probability is related to the RS codeword symbol error probability by

$$p_s = 1 - (1 - p_b)^m. \tag{7.15}$$

The overall throughput of the type-I hybrid-ARQ protocol for the given BER measurement can then be numerically found by

$$\eta_o = \sum_i x(p_{bi}) \eta_{block}(p_{bi}). \tag{7.16}$$

The goal of system design in this case is to determine the error correction capability of the RS code (t) and the number of codewords contained in a block (B), given the selected codeword length and overhead parameters. That is

$$maximize \sum_i x(p_{bi})\, \eta_{block}(B, t, p_{bi} | \, given \, n, h, h_B). \tag{7.17}$$

In the following, it is assumed that a (64,k) RS code is being used for error correction. Each codeword symbol contains 6 bits. Suppose that a CRC-16 code is used for error detection, and the block overhead is 16 symbols. The computation results for overall throughput are shown in Table 7.3. The contour plot and the 3-dimensional overall throughput surface are shown in Figure 7.7 and Figure 7.8, respectively.

Table 7.3: Computation results of overall throughput for S-R with RS codes

	Overall Throughput							
	B=1	2	3	4	5	6	7	8
t=0	0.0388	0.0139	0.0055	0.0024	0.0011	0.0005	0.0003	0.0001
1	0.7316	0.8018	0.8213	0.8260	0.8245	0.8199	0.8134	0.8059
2	0.4063	0.2773	0.1945	0.1471	0.1195	0.1023	0.0908	0.0825
3	0.2363	0.1568	0.1333	0.1236	0.1172	0.1119	0.1072	0.1029
4	0.2054	0.1582	0.1452	0.1396	0.1361	0.1334	0.1310	0.1288
5	0.2298	0.1797	0.1611	0.1526	0.1479	0.1451	0.1431	0.1417
6	0.2697	0.2128	0.1853	0.1709	0.1623	0.1567	0.1528	0.1499
7	0.3111	0.2546	0.2202	0.1994	0.1861	0.1770	0.1703	0.1651
8	0.3480	0.2999	0.2635	0.2381	0.2202	0.2073	0.1977	0.1902
9	0.3769	0.3423	0.3094	0.2837	0 2635	0.2475	0.2347	0.2245
10	0.3964	0.3770	0.3508	0.3285	0.3098	0.2939	0.2803	0.2684
11	0.4064	0.4015	0.3828	0.3652	0.3499	0.3367	0.3250	0.3144
12	0.4079	0.4155	0.4041	0.3910	0.3790	0.3685	0.3592	0.3509
13	0.4022	0.4198	0.4152	0.4066	0.3976	0.3891	0.3815	0.3746
14	0.3908	0.4155	0.4167	0.4125	0.4067	0.4006	0.3946	0.3890
15	0.3752	0.4042	0.4097	0.4092	0.4066	0.4031	0.3992	0.3953
16	0.3564	0.3876	0.3958	0.3979	0.3977	0.3961	0.3946	0.3925
17	0.3354	0.3673	0.3769	0.3805	0.3818	0.3820	0.3815	0.3808
18	0.3130	0.3444	0.3547	0.3591	0.3611	0.3621	0.3624	0.3624
19	0.2896	0.3197	0.3302	0.3350	0.3374	0.3388	0.3395	0.3399

Based on these numerical results, we can see that the maximum overall throughput is achieved at $t = 1$. However, this conclusion is not justified because the measurement data for the noisy channel clearly shows that most of time the channel will cause more than one codeword symbol error. What is wrong ?

7.4.3 Modified Throughput and More Numerical Results

Recall that in the throughput formula (7.14), both the probability of correct decoding P_{CD} and the probability of incorrect decoding P_{ICD} contribute to the numerical value of throughput. When the channel is very noisy, the value of P_{ICD} would be much larger than the value of P_{CD}. The optimum parameters thus found to provide the maximum overall throughput will not make any sense in practice because

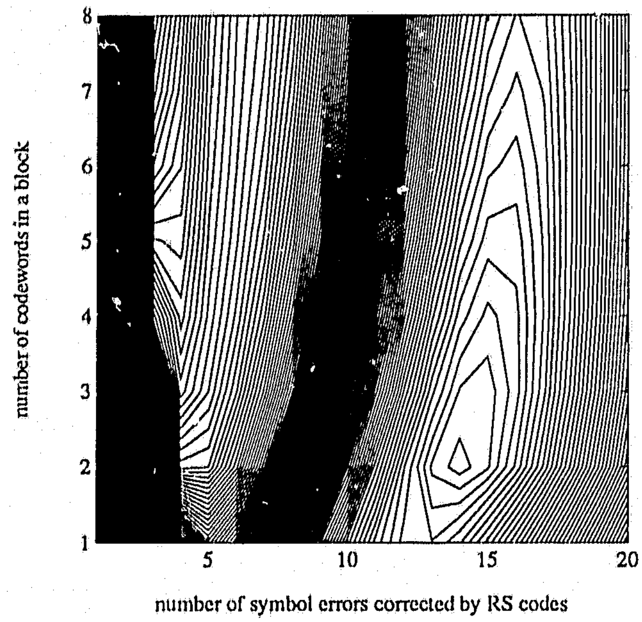

Figure 7.7: Contour plot of overall throughput (CRC-16 and 16 symbols overhead)

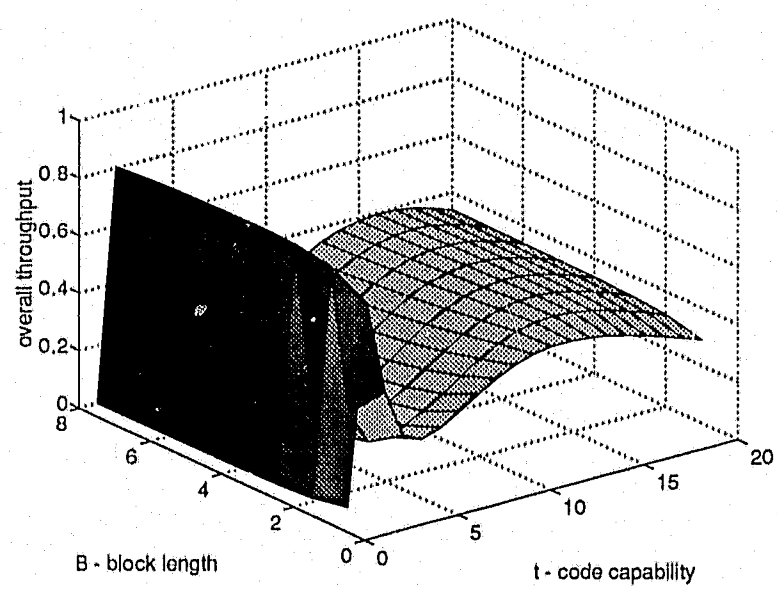

Figure 7.8: Illustration of 3-dimensional overall throughput surface

most information symbols received are in error. In other words, we must modify the definition of throughput in order to carry out justified numerical optimization. Therefore, we modify the throughput definition as

$$\eta' = \frac{\text{average no. of information symbols correctly received per unit time}}{\text{total no. of symbols that could be transmitted per unit time}}. \quad (7.18)$$

Based on this definition, the *modified (overall) throughput* is the overall throughput excluding the contribution of the probability of incorrectly decoding. Under the definition of modified throughput, the throughput formulas should be written as

$$\eta'_{typeI} = P_{CD} \cdot \left(\frac{n - 2t - h}{n} \right), \quad (7.19)$$

and

$$\eta'_{block} = P^{B}_{CD} \cdot \left(\frac{B(n - 2t - h)}{B\, n + h_B} \right). \quad (7.20)$$

For the previous example with the same data, the computation results for the modified throughput are shown in Table 7.4. The contour plot and the 3 dimensional modified throughput surface are shown in Figure 7.9 and Figure 7.10, respectively. The parameters which result in the maximum modified throughput can be found at $t = 13$ and $B = 2$. The same values of t and B were found to achieve the local maximum for the overall throughput.

As comparisons, we have also obtained numerical results for cases with different given parameters. The contour plots based on the computation results for the modified throughput in these other cases are shown in Figures 7.11 to 7.15. From these contour plots, it can be seen that the optimum parameters can be determined by the local maximum point in the contour plots. Notice that when block overhead

Table 7.4: Computation results of modified throughput for S-R with RS codes

	Modified Throughput							
	B=1	2	3	4	5	6	7	8
t=0	0.0388	0.0139	0.0055	0.0024	0.0011	0.0005	0.0003	0.0001
1	0.0863	0.0581	0.0406	0.0293	0.0218	0.0166	0.0129	0.0102
2	0.1241	0.1027	0.0891	0.0780	0.0687	0.0610	0.0546	0.0492
3	0.1555	0.1322	0.1237	0.1173	0.1116	0.1063	0.1015	0.0970
4	0.1887	0.1538	0.1434	0.1385	0.1353	0.1327	0.1303	0.1281
5	0.2272	0.1788	0.1607	0.1523	0.1478	0.1450	0.1430	0.1416
6	0.2694	0.2126	0.1852	0.1709	0.1623	0.1567	0.1527	0.1499
7	0.3111	0.2546	0.2202	0.1994	0.1861	0.1770	0.1703	0.1651
8	0.3480	0.2999	0.2635	0.2381	0.2202	0.2073	0.1977	0.1902
9	0.3769	0.3423	0.3094	0.2837	0.2635	0.2475	0.2347	0.2245
10	0.3964	0.3770	0.3508	0.3285	0.3098	0.2939	0.2803	0.2684
11	0.4064	0.4015	0.3828	0.3652	0.3499	0.3367	0.3250	0.3144
12	0.4079	0.4155	0.4041	0.3910	0.3790	0.3685	0.3592	0.3509
13	0.4022	0.4198	0.4152	0.4066	0.3976	0.3891	0.3815	0.3746
14	0.3908	0.4155	0.4167	0.4125	0.4067	0.4006	0.3946	0.3890
15	0.3752	0.4042	0.4097	0.4092	0.4066	0.4031	0.3992	0.3953
16	0.3564	0.3876	0.3958	0.3979	0.3977	0.3964	0.3946	0.3925
17	0.3354	0.3673	0.3769	0.3805	0.3818	0.3820	0.3815	0.3808
18	0.3130	0.3444	0.3547	0.3591	0.3611	0.3621	0.3624	0.3624
19	0.2896	0.3197	0.3302	0.3350	0.3374	0.3388	0.3395	0.3399

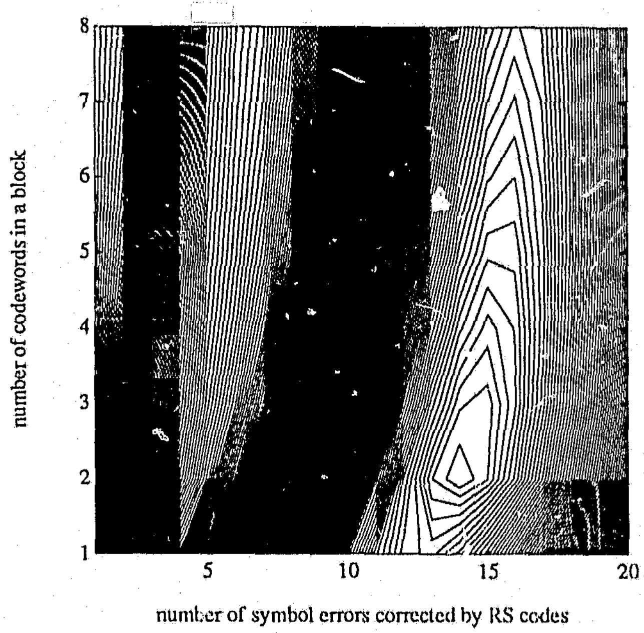

Figure 7.9: Contour plot of modified throughput (CRC-16 and 16 symbols overhead)

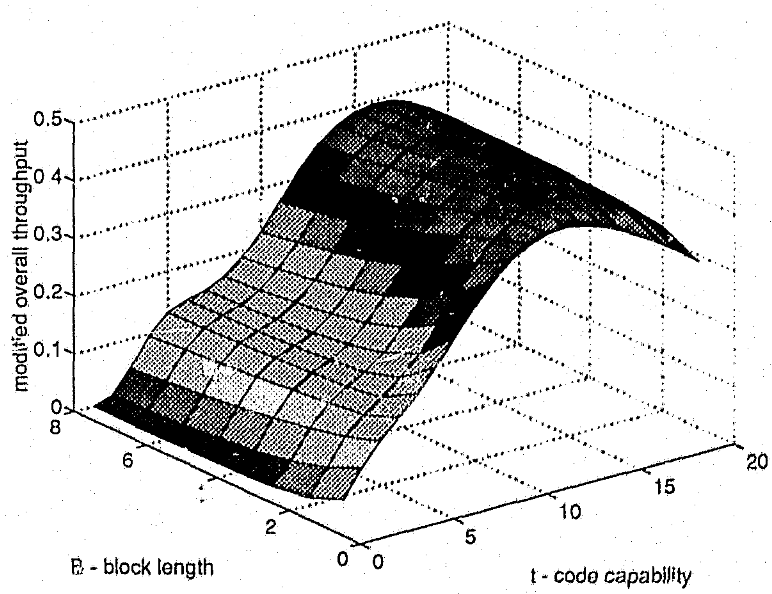

Figure 7.10: Illustration of 3-dimensional modified throughput surface

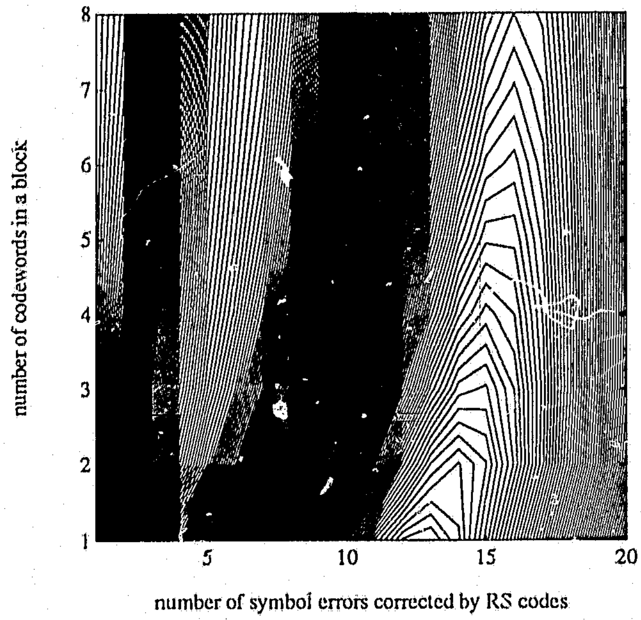

Figure 7.11: Contour plot of modified throughput (CRC-16 and 8 symbols overhead)

is small, say 8 symbols, the optimum design is achieved by assigning a single codeword to a block instead of using multi-codeword blocks.

7.5 Summary

The choice of error-control techniques and code parameters is primarily based on the channel characteristics. In this chapter, a pragmatic approach to the design of an optimum type-I hybrid error control protocol with a BCH code or an RS code is proposed by introducing the *overall throughput* and *modified throughput* as system performance measures. These quantities are based on real channel measurements. The method can also be used to carry out numerical optimization of type-II hybrid-ARQ protocols if necessary. It has been shown that the proposed method has great significance in practical system design, since it improves the overall throughput efficiency with minimal implementation complexity.

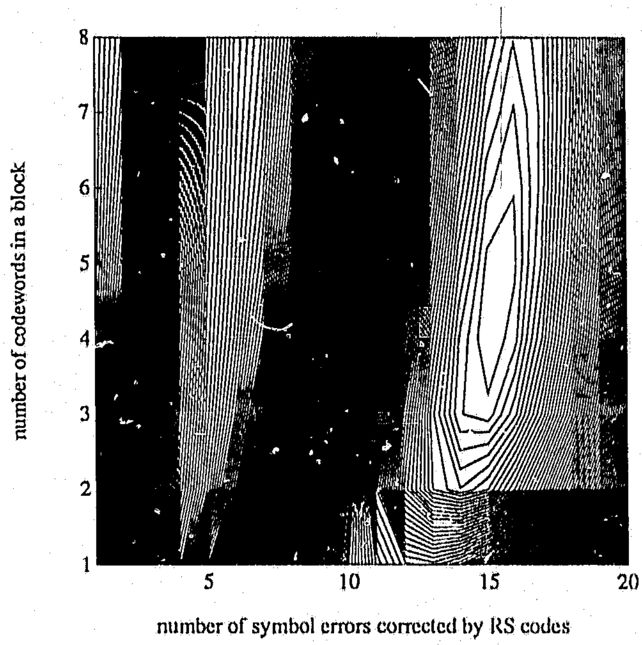

Figure 7.12: Contour plot of modified throughput (CRC-16 and 32 symbols overhead)

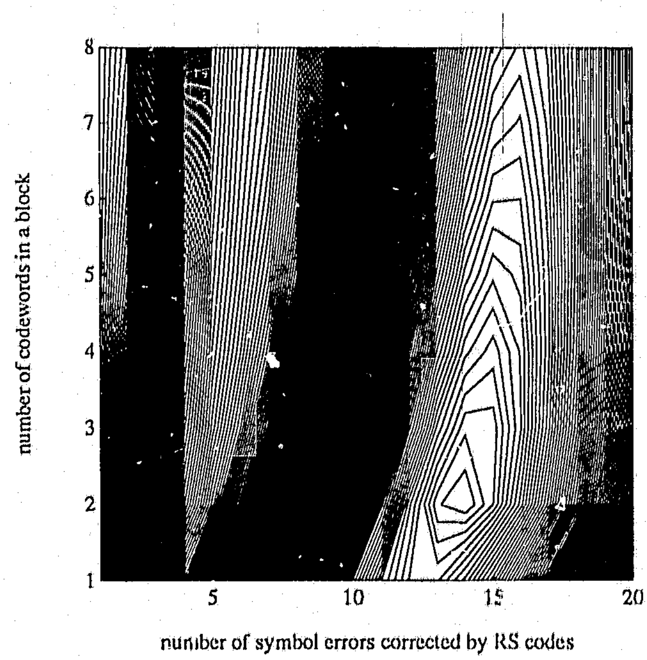

Figure 7.13: Contour plot of modified throughput (CRC-32 and 16 symbols overhead)

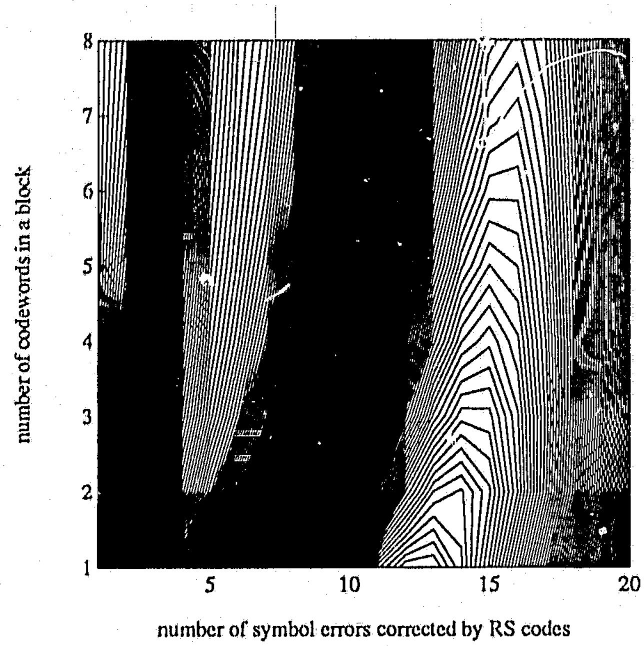

Figure 7.14: Contour plot of modified throughput (CRC-32 and 8 symbols overhead)

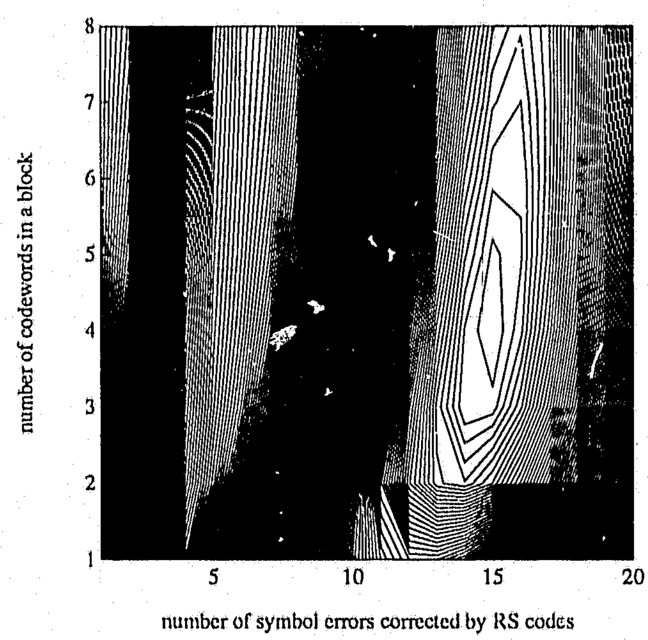

Figure 7.15: Contour plot of modified throughput (CRC-32 and 32 symbols overhead)

Chapter 8

Conclusions and Further Research

8.1 Summary of the Dissertation

For noisy channels, simple ARQ-based error control techniques fail to provide high speed reliable communications. Advanced hybrid error control techniques must be considered in order to achieve real-time communications over noisy channels. We have investigated the five major performance issues (delay, packet loss probability, coding gain, overall throughput and modified throughput) of hybrid error control techniques. By proper design of hybrid error control protocols, we have shown that performance improvement can be achieved with reduced delay and system complexity.

Throughput and reliability are by far the most extensively used performance measures for hybrid error control systems. However, throughput and reliability only may not provide sufficient tools to evaluate system performance for many power- and delay-limited applications. To compare the performance of various error control coding systems on a common basis, we have developed or extended three unified evaluation methods:

1. Imbedded Markov chain method;
2. Generalized coding gain method;

3. Modified overall throughput method.

These unified methods will form the fundamental basis for performance evaluation of other existing or new hybrid error control coding techniques.

To the best of our knowledge, this is the first time that the idea of matched-rate error control coding using feedback transmissions has been presented and investigated. The dynamic adaptation of error correction coding using fast channel estimation methods such as feedback transmissions seems to have opened a rich research area.

8.2 Suggestions for Future Work

The emphasis of this work has been given to the performance aspects of hybrid error control systems using BCH or Reed-Solomon codes. This was partly because analytic solutions can often be obtained for these codes so that our attention could be directed to the more fundamental issues. Our efforts in developing unified performance evaluation methods and showing the substantial performance improvements obtained by using the proposed hybrid error control techniques constituted the major portion of this work. In our opinion, this research can be further continued in the following directions:

- investigating the delay-related performance characteristics of hybrid-ARQ error control protocols with punctured convolutional codes [44] using the methods developed in this dissertation.

- exploring the possible performance improvement offered by the proposed matched-rate adaptive error control protocol under other nonstationary channels and mobile fading channels.

- implementing the proposed hybrid error control strategies based on industrial standards such as CCITT V.42 error-control architecture. The resulting products could be directly applicable to digital mobile cellular networks and mobile satellite communications.

8.3 Concluding Remarks

Hybrid error control coding has been a very active area of research in recent years. A large amount of literature is now available on hybrid error control and its applications. We have included a bibliography for further reference. More comprehensive lists of references are available in [9] [13] [68].

Error control coding has been used extensively in digital communication systems because of its cost-effectiveness in achieving efficient, reliable digital transmission. Coding now plays an important role in the design of modern communication systems. Over the past ten years, VLSI technology has reduced the cost of coding systems by many orders of magnitude. Future generations of technology are expected to continue this trend. Indeed, more complicated hybrid error control schemes will certainly become an economic necessity.

Appendix A

List of Symbols

C	Channel capacity
$ceil(x)$	The smallest integer which is equal to or greater than x
d	Minimum Hamming distance of a linear block code
D_q	Block delay (queueing delay)
D_t	Transmission delay
$dis(N, N^*)$	Hamming distance between block N and block N^*
E_b	The received energy per signal bit
$erfc(x)$	Complementary error function
h	Number of overhead bits or symbols
H	Entropy of an information source
$I_0(\cdot)$	Modified Bessel function of zeroth order
N_0	One-sided noise spectral density
p_b	Bit error probability
p_s	Symbol error probability
$P(z)$	Probability generating function
P_b	Average bit error probability
P_B	Block error probability
P_c	Probability that a received block contains no error
P_{CD}	Probability of correct decoding
P_d	Probability that a received block contains an uncorrectable but detectable error
P_e	Probability that a received block contains an undetectable error
P_E	Probability of block retransmission
P_{ED}	Probability of error detection
P_{ICD}	Probability of incorrect decoding
P_{loss}	Probability of packet loss

$Pr(E)$	—	Decoding error probability (protocol reliability)
R_d	—	Transmission data rate
t	—	Code error correction capability (number of correctable errors)
T_a	—	Receiver acknowledgment time
T_p	—	Channel one-way propagation delay
$U(N)$	—	Unit step function
α	—	ACK failure probability
γ	—	Nonselective fading envelope
$\Gamma(m)$	—	Gamma function of m
η	—	Throughput
η'	—	Modified throughput
η_o	—	Overall throughput
η_o'	—	Modified overall throughput
λ	—	Data arrival rate of a Poisson process
μ	—	Transmission efficiency
ν	—	Matching percentage
ρ	—	System load

Appendix B

List of Abbreviations

ACK	Acknowledgment
ARQ	Automatic repeat request
$ATDM$	Asynchronous time-division multiplexing
$AWGN$	Additive white Gaussian noise
BCH	Bose-Chaudhuri-Hocquenghem codes (a class of linear block codes)
BER	Channel bit error rate
bps	Bit per second
BSC	Binary symmetric channel
CRC	Cyclic redundancy check
dB	Decibels
DMC	Discrete memoryless channel
$DPSK$	Binary differential phase shift keying modulation
FEC	Forward error correction
GBN	Go-back-N
$GF(q)$	Galois field of q elements
$HDLC$	High data link control standard
$M/G/1$	A queueing model with Poisson arrival, general service and a single server
MDS	A class of linear block codes with maximum separable distance
ms	Millisecond
NAK	Negative acknowledgment
PSK	Phase shift keying modulation
RS	Reed-Solomon codes (a class of MDS codes)
SAW	Stop-and-wait
SNR	Signal-to-noise ratio

SR	—	Selective-repeat
TNB	—	Total number of blocks
$TNBE$	—	Total number of blocks in error
$type - I$	—	Hybrid error control with fixed code rate
$type - II$	—	Hybrid error control with variable code rate

Bibliography

[1] F. Adachi, S. Ito, and K. Ohno, "Performance analysis of a time diversity ARQ in land mobile radio," *IEEE Trans. Commun.*, vol. COM-37, pp. 177-183, Feb. 1989.

[2] M. E. Anagnostou and E. N. Protonotarios, "Performance analysis of the selective repeat ARQ protocol," *IEEE Trans. Commun.*, vol. COM-34, pp. 127-135, Feb. 1986.

[3] T. Aulin, "Characteristics of a digital mobile radio channel," *IEEE Trans. Veh. Technol.*, vol. VT-30, pp. 45-53, May 1981.

[4] B. B. Barrow, *Error Probabilities for Data Transmission over Fading Radio Paths*, Assen, The Netherlands: Van Gorcum, 1962.

[5] G. Benelli, "An ARQ scheme with memory and integrated modulation," *IEEE Trans. Commun.*, COM-35, pp. 689-697, July 1987.

[6] R. J. Benice and A. H. Frey Jr., "An analysis of retransmission systems," *IEEE Trans. Commun. Technol.*, vol. COM-12, pp. 135-145, Dec. 1964.

[7] E. R. Berlekamp, *Algebraic Coding Theory*, McGraw-Hill, New York, 1968.

[8] E. R. Berlekamp, "The technology of error-correcting codes," *Proc. IEEE*, vol. 68, pp. 564-593, May 1980.

[9] E. R. Berlekamp, R. E. Peile, and S. P. Pope, "The application of error control to communications," *IEEE Commun. Mag.*, vol. 25, pp. 44-57, Apr. 1987.

[10] D. Bertsekas, R. Gallager, *Data Networks*, Prentice-Hall, New Jersey, 1987.

[11] V. K. Bhargava, "Forward error correction schemes for digital communications," *IEEE Commun. Mag.*, vol. 21, pp. 11-19, Jan. 1983.

[12] V. K. Bhargava, D. Haccoun, P. Matyas, and P. Nuspl, *Digital Communication by Satellite*, Wiley, New York, 1981.

[13] V. K. Bhargava, Q. Yang, and D. Peterson, "Coding theory and its applications in communication systems," to appear in *Defense Science Journal* (invited paper), Dec. 1992.

[14] R. E. Blahut, *Theory and Practice of Error Control Codes*, Addison-Wesley, Reading, MA, 1983.

[15] M. B. Brilliant, "Observations of errors and error rates on T1 digital repeatered lines," *Bell Syst. Tech. J.*, vol. 57, pp. 711-746, Mar. 1978.

[16] H. Bruneel and M. Moeneclaey, "On the throughput performance of some continuous ARQ strategies with repeated transmissions," *IEEE Trans. Commun.*, vol. COM-34, pp. 244-248, Mar. 1986.

[17] H. O. Burton and D. D. Sullivan, "Errors and error control," *IEEE Proc.*, vol. 60, pp. 1293-1301, 1972.

[18] W. Bux, K. Kummerle, and H. L. Truong, "Balanced HDLC procedures: a performance analysis," *IEEE Trans. Commun.*, vol. COM-28, pp. 1889-1898, Nov. 1980.

[19] J. B. Cain, G. C. Clark, and J. Geist, "Punctured convolutional codes of rate $(n-1)/n$ and simplified maximum likelihood decoding," *IEEE Trans. Inform. Theory*, IT-25, pp. 97-100, Jan. 1979.

[20] J. P. Castro, "Statistical observations of data transmission over land mobile satellite channels," *IEEE J. Select. Areas Commun.*, vol. SAC-10, pp. 1227-1235, Oct. 1992.

[21] D. Chase, "Code combining - a maximum-likelihood decoding approach for combining an arbitrary number of noisy packets," *IEEE Trans. Commun.*, COM-33, pp. 385-393, May 1985.

[22] K. M. Cheung, "More on the decoder error probability for Reed-Solomon codes," *IEEE Trans. Inform. Theory*, vol. IT-35, pp. 895-900, 1989.

[23] W. W. Chu, "Optimal message block size for computer communications with error detection and retransmission strategies," *IEEE Trans. Commun.*, COM-22, pp. 1516-1525, Oct. 1974.

[24] G. C. Clark and J. B. Cain, *Error Correction Coding for Digital Communications*, Plenum Press, New York, 1981.

[25] T. Cover and S. Pombra, "Gaussian feedback capacity," *IEEE Trans. Inform. Theory*, IT-35, pp. 37-43, Jan. 1989.

[26] R. A. Comroe and D. J. Costello Jr., "ARQ schemes for data transmission in mobile radio systems," *IEEE J. Selected Areas in Commun.*, SAC-2, pp. 472-481, July 1984.

[27] P. J. Crepeau, "Uncoded and coded performance of MFSK and DPSK in Nakagami fading channels," *IEEE Trans. Commun.*, vol. COM-40, pp. 487-493, Mar. 1992.

[28] D. Cygan, "Analytical evaluation of average bit error rate for the land mobile satellite channel," *Int. J. Satellite Commun.*, vol. 7, pp. 99-102, 1989.

[29] A. Drukarev and D. J. Costello Jr., "A comparison of block and convolutional codes in ARQ error control schemes," *IEEE Trans. Commun.*, COM-30, 2449-2455, Nov. 1982.

[30] J. Du, M. Kasahara, and T. Namekawa, "Separable codes on Type-II hybrid ARQ systems," *IEEE Trans. Commun.*, vol. COM-36, pp. 1089-1097, Oct. 1988.

[31] P. M. Ebert, "The capacity of the Gaussian channel with feedback," *Bell Syst. Tech. J.*, vol. 49, pp. 1705-1712, Oct. 1970.

[32] E. O. Elliott, "Estimates of error rates for codes on burst-noise channels," *Bell Syst. Tech. J.*, vol. 42, pp. 1977-1997, Sept. 1963.

[33] R. Esposito, "Error probabilities for the Nakagami channel," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 145-148, Jan. 1967.

[34] R. Fantacci, "Performance evaluation of some ARQ schemes using efficient modulation techniques and noncoherent detection," *IEEE Trans. Commun.*, COM-39, pp. 445-451, Mar. 1991.

[35] R. Fantacci, "Efficient ARQ schemes for error control of satellite channels," *Int. J. Satellite Commun.*, vol. 7, pp. 193-200, 1989.

[36] P. M. Feldman and V. O. K. Li, "Delay analysis for adaptive hybrid ARQ protocol," *Proc. IEEE Military Comm. Conf.*, Washington D. C., pp. 388-392, 1987.

[37] G. D. Forney, *Concatenated Codes*, MIT Press, Cambridge, MA, 1966.

[38] B. D. Fritchman, "A binary channel characterization using partitioned Markov chains," *IEEE Trans. Inform. Theory*, IT-13, pp. 221-227, Apr. 1967.

[39] C. Fujiwara, S. Hirasawa and W. W. Chu, "Feedback error control system with limited number of retransmissions," *Proc. 3rd Symp. Information Theory and Its Applications of Japan*, pp. 328-332, Nov. 1980.

[40] R. G. Gallager, *Information Theory and Reliable Communication*, Wiley, New York, 1968.

[41] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, pp. 1253-1266, Sept. 1960.

[42] S. G. Glisic, "Efficiency of block error-correcting codes," *Electron. Lett.*, vol. 23, pp. 557-558, May 1987.

[43] J. G. Gruber and N. H. Le, "Performance requirements for integrated voice/data networks," *IEEE J. Selected Areas Commun.*, vol. SAC-1, pp. 981-1005, Dec. 1983.

[44] J. Hagenauer, "Rate compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, COM-36, pp. 389-400, Apr. 1988.

[45] J. F. Hayes, *Modeling and Analysis of Computer Communications Networks*, Plenum Press, New York, pp. 137-141, 1984.

[46] H. Imai, (ed.), *Essentials of Error-Control Coding Techniques*, Academic Press, San Diego, CA, 1990.

[47] V. I. Johannes, "Improving on bit error rate," *IEEE Commun. Mag.*, vol. 22, no. 12, pp. 18-20, Dec. 1984.

[48] S. Kallel, "Analysis of a type-II hybrid ARQ scheme with code combining," *IEEE Trans. Commun.*, COM-38, pp. 1133-1137, Aug. 1990.

[49] S. Kallel and D. Haccoun, "Generalized type-II hybrid-ARQ schemes using punctured convolutional codes," *IEEE Trans. Commun.*, COM-38, pp. 1938-1946, Nov. 1990.

[50] S. Kallel and D. Haccoun, "Sequential decoding with an efficient partial retransmission ARQ strategy," *IEEE Trans. Commun.*, COM-39, pp. 208-213, Feb. 1991.

[51] S. Kallel, "Analysis of memory and incremental redundancy ARQ schemes over a nonstationary channel," *IEEE Trans. Commun.*, COM-40, pp. 1474-1480, Sept. 1992.

[52] L. N. Kanal and A. R. K. Sastry, "Models for channels with memory and their applications to error control," *Proc. IEEE*, vol. 66, pp. 724-744, July 1978.

[53] T. Kasami and S. Lin, "On the probability of undetected error for the maximum distance separable codes," *IEEE Trans. on Commun.*, COM-32, pp. 998-1006, Sept. 1984.

[54] T. Kasami, T. Klove, and S. Lin, "Error detection with linear block codes," *IEEE Trans. Inform. Theory*, IT-29, pp. 131-136, Jan. 1983.

[55] D. G. Kendall, "Some problems in the theory of queues," *J. Royal Statistical Society*, **B13**, pp. 151-185, 1951.

[56] D. G. Kendall, "Stochastic processes occurring in the theory of queues and their analysis by means of the imbedded Markov chain," *The Annals of Mathematical Statistics*, **24**, pp. 338-354, 1953.

[57] R. S. Kennedy, *Fading dispersive communication channels*, Wiley, New York, 1969.

[58] M. H. Khan, T. Le-Ngoc and V. K. Bhargava, "Adaptive forward error control for digital satellite systems," *IEEE Trans. Aerospace and Electronic Sys.*, vol. AES-21, pp. 547-558, July 1985.

[59] L. Kleinrock, *Queueing Systems, vol. 1, Theory*, John Wiley & Sons, New York, pp. 191-194, 1975.

[60] A. G. Konheim, "A queueing analysis of two ARQ protocols," *IEEE Trans. Commun.*, vol. COM-28, pp. 1004-1014, July 1980.

[61] M. A. Kousa and M. Rahman, "An adaptive error control system using hybrid ARQ schemes," *IEEE Trans. Commun.*, COM-39, pp. 1049-1057, July 1991.

[62] H. Krishna and S. D. Morgera, "A new error control scheme for hybrid ARQ systems," *IEEE Trans. Commun.*, COM-35, pp. 981-989, Oct. 1987.

[63] W. C. Y. Lee, "Estimate of channel capacity in Rayleigh fading environment," *IEEE Trans. Vehicular Technol.*, VT-39, pp. 187-189, Aug. 1990.

[64] C. H. C. Leung, Y. Kikumoto and S. A. Sorensen, "The throughput efficiency of the Go-Back-N ARQ scheme under Markov and related error structures," *IEEE Trans. Commun.*, COM-36, pp. 231-234, Feb. 1988.

[65] S. K. Leung-Yan-Cheong and M. E. Hellman, "Concerning a bound on undetected error probability," *IEEE Trans. Inform. Theory*, IT-22, pp. 235-237, Mar. 1976.

[66] S. K. Leung-Yan-Cheong, E. R. Barnes, and D. U. Friedman, "Some properties of undetected error probability of linear codes," *IEEE Trans. Inform. Theory*, IT-25, pp. 110-112, Jan. 1979.

[67] S. Lin and D. J. Costello Jr., *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, New Jersey, pp. 85-116, 1983.

[68] S. Lin, D. J. Costello Jr., and M. J. Miller, "Automatic-repeat-request error-control schemes," *IEEE Commun. Mag.*, vol. 22, pp. 5-17, Dec. 1984.

[69] S. Lin and P. S. Yu, "A hybrid ARQ scheme with parity retransmission for error control of satellite channels," *IEEE Trans. Commun.*, vol. COM-30, pp. 1701-1719, July 1982.

[70] J. D. C. Little, "A proof of the queueing formula $L = \lambda W$", *Operations Res.*, vol. 9, no. 3, pp. 383-387, 1961.

[71] C. Loo and N. Secord, "Computer models for fading channels with applications to digital transmission," *IEEE Trans. Veh. Technol.*, vol. VT-40, pp. 700-707, Nov. 1991.

[72] D. G. Luenberger, *Linear and Nonlinear Programming*, 2nd ed., Addison-Wesley, Reading, MA, 1984.

[73] L. R. Lugand, D. J. Costello Jr., and R. H. Deng, "Parity retransmission hybrid ARQ using rate 1/2 convolutional codes on a nonstationary channel," *IEEE Trans. Commun.*, vol. COM-37, pp. 755-765, July 1989.

[74] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, New York, 1977.

[75] D. M. Mandelbaum, "An adaptive-feedback coding scheme using incremental redundancy," *IEEE Trans. Inform. Theory*, IT-20, pp. 388-389, May 1974.

[76] D. M. Mandelbaum, "On forward error correction with adaptive decoding," *IEEE Trans. Inform. Theory*, IT-21, pp. 230-233, Mar. 1975.

[77] J. A. C. Martins and J. D. C. Alves, "ARQ protocols with adaptive block size perform better over a wide range of bit error rates," *IEEE Trans. Commun.*, COM-38, pp. 737-739, June 1990.

[78] R. J. McEliece and L. Swanson, "On the decoder error probability for Reed-Solomon codes," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 701-703, 1986.

[79] P. Merkey and E. C. Posner, "Optimum cyclic redundancy codes for noisy channels," *IEEE Trans. Inform. Theory*, vol. IT-30, pp. 865-867, Nov. 1984.

[80] J. J. Metzner, "Improvements in block-retransmission schemes," *IEEE Trans. Commun.*, COM-27, pp. 524-532, Feb. 1979.

[81] A. M. Michelson and A. H. Levesque, *Error-control Techniques for Digital Communication*, Wiley, New York, 1985.

[82] M. J. Miller and S. Lin, "The analysis of some selective-repeat ARQ schemes with finite receiver buffer," *IEEE Trans. Commun.*, COM-29, pp. 1307-1315, Sept. 1981.

[83] M. Moeneclaey and H. Bruneel, "Efficient ARQ schemes for high error rate channels," *Electron. Lett.*, vol. 20, pp. 986-987, Nov. 1984.

[84] J. N. Morris, "Another go-back-N ARQ under high error rate conditions," *IEEE Trans. Commun.*, vol. COM-26, pp. 187-189, Jan. 1978.

[85] J. N. Morris, "Optimal blocklengths for ARQ error control schemes," *IEEE Trans. Commun.*, vol. COM-27, pp. 488-493, Feb. 1979.

[86] M. Nakagami, "The m-distribution — A general formula of intensity distribution of fading," in *Statistical Methods in Radio Wave Propagation*, W. C. Hoffman, (ed.), London, England: Pergamon, 1960.

[87] M. Nakamura and T. Kodama, "Performance evaluation for ARQ schemes in power and/or bandwidth limited systems," *The Trans. of The IEICE*, vol. E72, pp. 494-501, May 1989.

[88] M. Nesenbergs, "Binary error probability due to an adaptable fading model," *IEEE Trans. Commun.*, vol. COM-12, pp. 64-73, Mar. 1964.

[89] J. P. Odenwalder, "Error control," in *Data Communications, Networks, and Systems*, Chap. 10, T. C. Bartee, (ed.), H. W. Sams and Co., 1985.

[90] W. W. Peterson and E. J. Weldon Jr., *Error Correcting Codes*, M. I. T. Press, Cambridge, MA, 1972.

[91] E. Y. Rocher and R. L. Pickholtz, "An analysis of the effectiveness of hybrid transmission schemes," *IBM J. Res. Dev.*, pp. 426-433, July 1970.

[92] B. H. Saeki and I. Rubin, "An analysis of a TDMA channel using stop-and-wait, block, and select-and-repeat ARQ error control," *IEEE Trans. Commun.*, vol. COM-30, pp. 1162-1173, May 1982.

[93] A. R. K. Sastry, "Improving automatic-repeat-request (ARQ) performance on satellite channels under high error rate conditions," *IEEE Trans. Commun.*, vol. COM-23, pp. 436-439, Apr. 1975.

[94] A. R. K. Sastry, "Performance of hybrid error control schemes on satellite channels," *IEEE Trans. Commun.*, COM-23, pp. 689-694, July 1975.

[95] T. Sato, K. Tokuda, M. Kawabe, and T. Kato, "Simulation of burst error models and an adaptive error control scheme for high speed data transmission over analog cellular systems," *IEEE Trans. Vehic. Technol.*, vol. VT-40, pp. 443-452, May 1991.

[96] T. Sato, M. Kawabe, T. Kato, and A. Fu' awa, "Protocol configuration and verification of an adaptive error control scheme over analog cellular networks," *IEEE Trans. Vehic. Technol.*, vol. VT-41, pp. 69-76, Feb. 1992.

[97] M. Schwartz, *Telecommunication Networks: Protocols, Modeling and Analysis*, Addison-Wesley, Reading, MA, 1987.

[98] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 17, pp. 379-423, July 1948.

[99] C. E. Shannon, "The zero error capacity of a noisy channel," *IRE Trans. on Inform. Theory*, IT-2, pp. 8-19, Sept. 1956.

[100] A. Shiozaki, K. Okuno, K. Suzuki, and T. Segawa, "A hybrid ARQ scheme with adaptive forward error correction for satellite communications," *IEEE Trans. Commun.*, COM-39, pp. 482-484, Apr. 1991.

[101] A. K. Sinha, B. Agrawal and W. W. Wu, "Trends in satellite communication technology, techniques and applications," *Int. J. Satellite Commun.*, vol. 8, pp. 283-294, 1990.

[102] L. B. Sklar, "Efficiency factors in data communications," *IEEE Commun. Mag.*, vol. 22, no. 6, pp. 33-36, June 1984.

[103] B. Stevan, "Techniques for bit error rate measurement using Chebyshev inequality," *Electron. Lett.*, vol. 25, pp. 927-929, July 1989.

[104] H. Suzuki, "A statistical model for urban radio propagation," *IEEE Trans. Commun.*, vol. COM-25, pp. 673-680, July 1977.

[105] A. A. R. Townsend, *Digital Line-of-sight Radio Links*, New York: Prentice Hall, New Jersey, 1988.

[106] D. Towsley and J. K. Wolf, "On the statistical analysis of queue lengths and waiting times for statistical multiplexers with ARQ retransmission," *IEEE Trans. Commun.*, vol. COM-27, pp. 693-703, Apr. 1979.

[107] D. A. Tugal and O. Tugal, *Data Transmission*, McGraw-Hill, New York, 1989.

[108] A. J. Viterbi, "Convolutional codes and their performance in communication systems," *IEEE Trans. Commun. Technol.*, COM-19, pp. 751-772, Oct. 1971.

[109] B. Vucetic, "An adaptive coding scheme for time-varying channels," *IEEE Trans. Commun.*, COM-39, pp. 653-663, May 1991.

[110] B. Vucetic, D. Drajic and D. Perisic, "Algorithm for adaptive error control system synthesis," *IEE Proceedings*, vol. 135, *Pt. F*, pp. 85-94, Feb. 1988.

[111] Y. M. Wang and S. Lin, "A modified selective-repeat type-II hybrid ARQ system and its performance analysis," *IEEE Tran Commun.*, COM-31, pp. 593-608, May 1983.

[112] E. J. Weldon Jr., "An improved selective-repeat ARQ strategy," *IEEE Trans. Commun.*, COM-30, pp. 480-486, Mar. 1982.

[113] S. B. Wicker, "High-reliability data transfer over the land mobile radio channel using interleaved hybrid-ARQ error control," *IEEE Trans. Vehicular Technol.*, VT-39, pp. 48-55, Feb. 1990.

[114] S. B. Wicker, "Reed-Solomon error control coding for Rayleigh fading channels with feedback," *IEEE Trans. Vehic. Technol.*, vol. VT-41, pp. 124-133, May 1992.

[115] S. B. Wicker, "Type-II hybrid-ARQ protocols using punctured Reed-Solomon codes," *Proc. IEEE Military Comm. Conf.*, MacLean, Virginia, pp. 52.2.1 - 52.2.6, Nov. 1991.

[116] J. Wolfowitz, "Memory increases capacity," *Inform. Control*, vol. 11, pp. 423-428, Oct. 1967.

[117] W. W. Wu, D. Haccoun, R. E. Peile, and Y. Hirata, "Coding for satellite communication," *IEEE J. Select. Areas Commun.*, vol. SAC-5, pp. 724-748, May 1987.

[118] Q. Yang and V. K. Bhargava, "Optimum coding design for type-I hybrid ARQ error control schemes," *Electron. Lett.*, vol. 25, pp. 1595-1596, Nov. 1989.

[119] Q. Yang and V. K. Bhargava, "Reducing time delay of ARQ protocols by simple error-correcting codes," *Electron. Lett.*, vol. 26, pp. 1447-1448, Aug. 1990.

[120] Q. Yang and V. K. Bhargava, "Delay and coding gain analysis of a truncated type-II hybrid ARQ protocol over Nakagami fading channels," *IEEE Trans. Veh. Technol.*, vol. VT-42, No. 1, pp. 22-32, Feb. 1993.

[121] Q. Yang, V. K. Bhargava and Q. Wang, "Imbedded Markov chain analysis of delay and packet loss for ATDM with hybrid-ARQ error control," submitted to *IEEE/ACM Trans. Networking*, Aug. 1992.

[122] Q. Yang and V. K. Bhargava, "Performance evaluation of error control protocols in mobile satellite communications," *Proc. IEEE Int. Conf. on Wireless Commun.*, Vancouver, B.C., pp. 368-371, June 1992.

[123] Q. Yang and V. K. Bhargava, "Matched-rate adaptive error control using Reed-Solomon codes with feedback transmissions," submitted to *IEEE Trans. on Commun.*, Dec. 1992.