The Analysis and Simulation of a Generalised Hybrid ARQ System for a Burst-Noise Channel

Frédéric Simard
Department of Electrical Engineering
McGill University, Montréal

March, 1990

A Thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Engineering

Abstract

As digital systems become widespread, the importance of error control in these systems increases. Further, since most channels cannot be realistically modelled by a simple Binary Symmetric Channel, it is required to reliably estimate the performance of error control schemes on real channels. This thesis considers the analysis and simulation of a Generalised Hybrid ARQ Type II (GH-ARQ II) error control scheme on a channel modelled by the Gilbert-Elliott model. The analysis is easily extended to higher order systems and channels modelled by first-order Markov chains. The results indicate that the performance of the GH-ARQ II scheme improves as the errors in the channel become burstier in nature and that this scheme is well suited to channels with relatively slowly varying error statistics. Further, it is found that the roundtrip delay of the selective repeat retransmission strategy may affect the performance of this error control scheme on burst-noise channels.

Précis

Compte tenu que les systèmes numériques sont devenus très courants, le contrôle d'erreurs dans ces systèmes est une considération importante dans leur conception. De plus, la plupart des canaux de communications ne peuvent pas être convenablement décrits par un canal symétrique binaire. Cette thèse consiste en l'analyse et la simulation d'un système de contrôle d'erreurs ARQ hybride généralisé type II (GH-ARQ II) lorsque le canal est décrit par le modèle de Gilbert et Elliott. L'analyse peut être facilement étendue à des systèmes d'ordre supérieur et à des canaux représentés par des chaînes Markov de premier ordre. Les résultats démontrent que la performance de ce système GH-ARQ II s'améliore lorsque les erreurs deviennent plus concentrées. De plus, le délai aller-retour dans la stratégie de retransmission peut affecter la performance de ce type de système de contrôle d'erreurs.

Acknowledgements

This thesis would surely not have seen the light of day had it not been for the support of several important people. First and foremost, I am greatly indebted to *Salvatore Morgera* who initially introduced me to this topic. His guidance, encouragement and pertinent suggestions are sincerely appreciated. In the end, he was more than just a teacher; he was a mentor.

My sincere thanks to *Bernard Armour* for always putting things in their proper perspective. Our daily lunches in Place Montréal Trust were a welcome relief from the daily pressures of school. I would also like to recognize the contributions made by *Vitalice Oduol*. My discussions with him were extremely helpful in the early stages of my work. Acknowledgement is also owed to *Salvatore Torrente*, dedicated system manager of the Integrated Networks and Systems Laboratory. His diligence and assistance are greatly appreciated and significantly eased my transition into these new facilities. The Natural Sciences and Engineering Research Council of Canada is also acknowledged for its financial support.

To my family for their patience and moral support during the occasional times of despair that seem inevitable whenever one seeks to shed some light on an obscure corridor in science. To *Calvin & Hobbes* for a daily smile. Finally, to *Robert Caissy* and *Bob Turnbull*, with whom I learned some of life's most important lessons late at night under the starlight of a clear Vermont night.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The importance of error control coding has steadily increased since its origin in the late 1940's. Error control coding can be used to ensure the reliability of digital systems; for example, a single error in an air traffic control system could have disastrous consequences. It can also be used to render digital systems more economical as in trellis coded modulation [7] which can transmit more information in a given bandwidth than uncoded modulation. Error control coding may allow a decrease in the required transmission power of satellite systems while maintaining their reliability, thereby allowing closer spacing of these satellites in geosynchronous orbit. It has also been sucessfully applied in the memory architecture of the Cray 1 supercomputer [9]. The reliability and capacity of this computer's memory modules were increased by using a code capable of correcting one error and detecting two errors. In summary then, the applications of error control are varied and not restricted to communications systems.

1.1 Error Control Techniques

The basic elements of a typical communications systems are illustrated in Figure 1.1.

The *source* generates the message $\underline{I}$ which is to be communicated. The *transmitter* suitably encodes this message to form the *transmitted packet* $\underline{t}$. This packet then travels through the *channel*[1] which, depending upon its state, may or may not cause errors. The *receiver* then accepts this possibly noisy *received packet* $\underline{r}$ and uses the code to correct or detect the errors that may have occurred. After the receiver has decoded the received packet, its estimate of the message $\tilde{\underline{I}}$ is delivered to the *sink*.

[1]To distinguish it from the feedback channel, this channel is sometimes referred to as the *forward channel.*

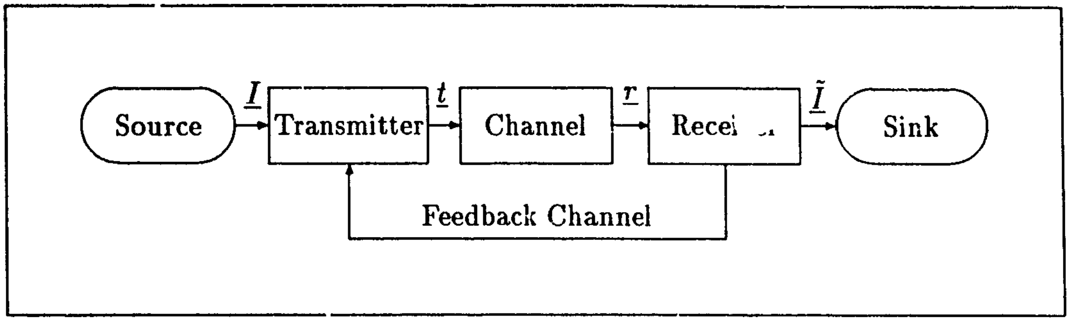

Figure 1.1: Basic Elements of a Communications System

If the communications system is to be useful, the message at the source $\underline{I}$ should be the same as the message at the sink $\underline{\tilde{I}}$ with high probability[2]. The *feedback channel* allows the receiver to send messages back to the transmitter. Depending upon the type of error control scheme and protocol, it may or may not be present; if it is, it may be the same as the forward channel.

The performance of a communications system is measured by two quantities: *throughput efficiency* and *reliability*. The definition of throughput efficiency, or simply throughput, varies according to the type of the analysis. In this context,[3] it is defined to be the ratio of the number of message bits accepted by the receiver to the total number of bits transmitted. It measures, in a sense, the overhead[4] that is required to communicate the message. The reliability, or probability of undetected error, of the system is the probability that the receiver delivers a message containing errors to the data sink. The goal when designing a communications system is to make it as efficient as possible while maintaining an acceptable reliability Usually there is some tradeoff between throughput and reliability; a highly reliable system tends to have low throughput (high overhead), while a highly efficient (low overhead) system tends to be less reliable. This idea translates into two classes of communications systems: Forward Error Correction (FEC) and Automatic Repeat Request (ARQ).

The *Forward Error Correction* technique encodes the message $\underline{I}$ with a code that the receiver will use to correct the errors that may have occurred during transmission. If the receiver is unable to correct the errors in the received packet, the decoded message $\underline{\tilde{I}}$ will contain errors. Consequently, the throughput efficiency of this scheme is

[2]Some systems are less sensitive to errors than are others For example, a voice system may be able to tolerate a limited frequency of errors without a significant quality degradation, while a data system probably cannot tolerate a single error.

[3]In other contexts, the throughput efficiency is defined to be the number of transmitted bits per unit time.

[4]The overhead is understood to comprise two components· the parity bits required for error correction or detection and the number of retransmissions that might be requested by the receiver via a feedback channel.

constant and equal to the rate of the code (the ratio of the number of information bits to the total number of encoded bits). However, since the code is only capable of correcting a given number of errors, its reliability deteriorates as the channel conditions worsen. This scheme does not require the feedback channel and may be the only choice when no such channel is available.

The receiver of an *Automatic Repeat Request* system uses the coding to detect the presence of errors in the received packet If errors are detected, the receiver will discard the received packet and will request that the transmitter retransmit the encoded message $\underline{t}$ This request is achieved by sending a negative acknowledgement message, or NAK, through the feedback channel to the transmitter. Otherwise, if the receiver does not detect errors in the received packet, it sends a positive acknowledgement message, or ACK, to the transmitter and delivers the recovered message $\underline{\tilde{I}}$ to the data sink. The receiver will continue to request that the packet $\underline{t}$ be retransmitted until it fails to detect the presence of errors in the received packet $\underline{r}$. Note that the receiver may deliver an erroneous message to the sink if the number of errors in the received packet exceeds the code's error detection capability. Since a given code is capable of detecting more errors than it can correct, the reliability of the ARQ scheme is higher than that of a comparable FEC scheme. This comes at the expense of a lower throughput efficiency, since the ARQ scheme may require more than one transmission to communicate a message $\underline{I}$. Interestingly, Shannon showed that use of a feedback channel does not increase the capacity of a memoryless forward channel, but can increase its reliability at all rates below the channel's capacity [45]. Also, it is usually assumed that the feedback channel is noiseless, so that the messages from the receiver to the transmitter are always correctly communicated.

In summary, the throughput efficiency of FEC systems is constant and usually greater than that of ARQ systems (provided they use the same code), but this comes at the expense of lower reliability (higher probability of undetected error), especially when the channel conditions are poor. For a given code and packet length, the complexity of error correction is greater than that of error detection. However, since the ARQ system requires additional circuitry to support the feedback channel as well as one of the *retransmission strategies* discussed in Section 1.3, its complexity is greater than that of a comparable FEC system. The performance analysis of these error control schemes has generated a wealth of literature; see, for example, [24, 38, 39, 60, 61].

1.2 Hybrid ARQ Variations

In order to overcome the limitations of the FEC and ARQ error control schemes, several hybrid systems have been proposed. The goal of these systems is to combine the efficiency of FEC with the reliability of ARQ. There are two broad types of hybrid ARQ; the first is well suited to channels whose characteristics are well-known and stable, the second is appropriate when the channel conditions are unknown or time-varying.

Type I Hybrid ARQ combines both error detection and error correction in an attempt to benefit from the advantages of FEC and ARQ. Rocher and Pickholtz [17] considered a scheme whereby the message to be communicated is encoded with a code that will be used by the receiver to correct and detect errors. The receiver attempts to correct the errors that may have occurred during transmission and then performs error detection to check for any remaining uncorrectable errors In this scheme, the error correction characteristic of FEC may decrease the required number of retransmissions (compared to a pure ARQ scheme), while the error detection characteristic of ARQ may improve its reliability (compared to a pure FEC scheme). They found that on binary symmetric channels, the use of Hybrid ARQ became attractive when error rates exceed 10^{-4}. Sastry [44] investigated the performance of this system on satellite channels where the roundtrip retransmission delay is long; he found that it offered a "substantial" performance improvement over FEC and ARQ. Kasami *et al.* [37] considered a concatenated coding scheme that utilized two codes: the inner code is used for both error correction and detection, while the outer code is used only for error detection. Deng and Costello [6] found that this approach can provide extremely high throughput and low probability of undetected error. They concluded their analysis by suggesting that this scheme is a good candidate for use on high speed channels such as satellite and file-transfer systems.

This type of hybrid ARQ is well suited to channels whose characteristics are known and stable so that an appropriate error correction code may be used. However, if the channel is quieter than expected, the additional redundancy included for error correction is wasted. Whereas, if the channel is noisier, that redundancy is still wasted because the receiver will be unable to correct all the errors caused by the channel and will have to request a retransmission from the transmitter.

Type II Hybrid ARQ differs from Type I by its adaptive nature; the parity bits used for error correction are sent to the receiver only when they are required. This scheme was initially proposed by Mandelbaum in 1974 [14]. It requires less *a priori* knowledge about the channel and, hence, is well suited to conditions in which the error rate is nonstationary.

Lin and Yu proposed a Type II Hybrid ARQ scheme using two codes for use on satellite channels [21]. The outer (N, k) block code $\mathcal{C}_0$ is designed for error detection only, while the inner $(2k, k)$ block code $\mathcal{C}_1$ is designed for both error detection and error correction. Further, $\mathcal{C}_1$ is *invertible* in the sense that the message $\underline{I}$ may be uniquely determined by its parity bits $P(\underline{I})$. The k information bits to be communicated are coded with the inner code $\mathcal{C}_1$. Assuming that this code is systematic, the result is a

$2k$-bit codeword $(\underline{I} \mid P(\underline{I}))$ in C_1, where $\underline{I}$ is the message and $P(\underline{I})$ corresponds to the parity bits generated by C_1. The requirement that C_1 be systematic is not stringent, and merely for notational convenience[5]. The k message bits that form $\underline{I}$ are then coded with the outer error detection code C_0 to form the packet $(\underline{I}, \underline{Q})$ which is then transmitted. Here, $\underline{Q}$ corresponds to the parity bits resulting from the error detection encoding

At the receiver, the syndrome of the received packet $\underline{r} = (\hat{\underline{I}}, \hat{\underline{Q}})$ is computed. If the syndrome is zero, the transmission is assumed successful, the receiver returns ACK to the transmitter and the recovered information bits $\tilde{\underline{I}} = \hat{\underline{I}}$ are delivered to the data sink. However, if the syndrome is nonzero, errors have been detected. The receiver returns NAK to the transmitter and stores the currently received packet in its memory. When the transmitter receives the NAK from the receiver, it encodes the parity block $P(\underline{I})$ with the error detection code C_0 to form $(P(\underline{I}), \underline{Q}')$ and transmits this packet. Here $\underline{Q}'$ corresponds to the parity bits resulting from the error detection encoding of $P(\underline{I})$ with C_0. Upon reception of this second transmission, the receiver begins by attempting to detect occurs that might be present in the received packet $(\widehat{P(\underline{I})}, \hat{\underline{Q}}')$. In the case that no errors are detected, the receiver returns ACK to the transmitter; $\tilde{\underline{I}}$ is then recovered from $\widehat{P(\underline{I})}$ by inversion and delivered to the data sink. Otherwise, the receiver combines this packet with the previously received one to form $(\hat{\underline{I}}, \widehat{P(\underline{I})})$. This combined packet may then be decoded using the C_1 code in an attempt to correct the errors. If the error pattern is correctable, then the receiver returns ACK to the transmitter and the recovered information $\tilde{\underline{I}}$ is delivered to the data sink. Conversely, in the case that the receiver determines that the error pattern is uncorrectable, it returns NAK to the transmitter and the packet $(\hat{\underline{I}}, \hat{\underline{Q}})$ is replaced by $(\widehat{P(\underline{I})}, \hat{\underline{Q}}')$ in its memory. The transmitter alternates transmissions of $(\underline{I}, \underline{Q})$ with $(P(\underline{I}), \underline{Q}')$ until it receives a positive acknowledgement from the receiver.

Lin and Yu analyzed the performance of this system on a discrete memoryless channel with a finite-buffer selective repeat request retransmission strategy. They found that this scheme offered higher throughput efficiency than a simple ARQ error control scheme without error correction. It was also found that its reliability is the same as that of this simple ARQ scheme, if the error detection capability of the inner code C_1 is sufficiently large. Since the Type II hybrid scheme requires an inversion circuit and an error detection circuit for C_0, its complexity is greater than that of a Type I hybrid scheme. This additional complexity may be compensated for by better performance if the throughput efficiency is sufficiently increased while maintaining an acceptable reliability.

Although the scheme proposed by Lin and Yu used block codes for C_0 and C_1, it is

[5] **In fact, a system that uses a nonsystematic error correction code will be considered in a later chapter.**

possible to use convolutional codes instead. Lugand *et al.* [4] analyzed the performance of such a scheme on a channel described by a simple burst-noise model. They found that its throughput efficiency increases and its reliability improves as the noise in the channel becomes progressively burstier.

Wang and Lin [20] proposed a Type II Hybrid ARQ scheme similar to that proposed by Lin and Yu. In this scheme, $\mathcal{C}_0$ is an (N, k) code used for error detection and $\mathcal{C}_1$ is a $(2N, N)$ invertible code used for error correction only. Unlike the Lin-Yu scheme, this scheme uses the full error correction capability of $\mathcal{C}_1$. The Wang-Lin scheme begins by encoding the message $\underline{I}$ with the error detection code $\mathcal{C}_0$ to form the packet $\underline{t} = (\underline{I}, \underline{Q})$, where $\underline{Q}$ represents the error detection parity bits. Then, $\underline{t}$ is encoded with the error correction code $\mathcal{C}_1$ to form $(\underline{t}, P(\underline{t}))$. Again, it is assumed for notational convenience that $\mathcal{C}_1$ is a systematic code. The packet $\underline{t}$ is then transmitted.

Upon reception of the possibly noisy packet $\underline{r} = (\hat{\underline{I}}, \hat{\underline{Q}})$, the receiver attempts to detect the presence of errors by using $\mathcal{C}_0$. If no errors are detected, the packet is assumed error-free, the recovered message $\tilde{\underline{I}} = \hat{\underline{I}}$ is delivered to the data sink, and ACK is returned to the transmitter. Otherwise, errors have been detected, the receiver stores $\hat{\underline{t}}$ in its buffer in preparation for the next transmission and returns NAK to the transmitter. Upon receiving the NAK message from the receiver, the transmitter will transmit the packet $P(\underline{t})$. The receiver begins by inverting $\underline{r} = \widehat{P(\underline{t})}$ to form $P^{-1}[\widehat{P(\underline{t})}]$. This inverted packet is then checked for errors with the code $\mathcal{C}_0$; if none are detected, the transmission is positively acknowledged and the recovered message is delivered to the sink. Otherwise, the receiver combines this packet with the previously transmitted packet to form the combined packet $(\hat{\underline{t}}, \widehat{P(\underline{t})})$. The error correction code $\mathcal{C}_1$ is then used to attempt to correct the errors in the combined packet. Let the resulting decoded packet be denoted by $(\underline{I}^\star, \underline{Q}^\star)$. This packet is then checked for errors; if none are detected, the receiver returns ACK to the transmitter and delivers $\tilde{\underline{I}} = \underline{I}^\star$ to the sink. Otherwise, the receiver discards $\hat{\underline{t}}$ from its memory, replaces it by $\widehat{P(\underline{t})}$ and returns NAK to the transmitter. Note that the transmitter alternately transmits $\underline{t}$ and $P(\underline{t})$. When the packet length is small, Wang and Lin found that their approach resulted in slightly higher throughput efficiency than the scheme proposed by Lin and Yu, probably because it uses the full error correction capability of the inner code.

The Type-II Hybrid ARQ error control schemes considered up to now have used a half-rate error correction code; however, this scheme may be generalised so that the parity bits of a code with progressively greater minimum distance are sent to the receiver on each retransmission. For example, instead of sending the packet $(\underline{I}, \underline{Q})$ on the second retransmission, it would be advantageous to transmit parity bits that the receiver could use for decoding with a rate one-third error correction code. In this scheme, the receiver begins by inverting the received packet and then checking for errors. If the inversion is unsuccessful, the receiver then combines this packet with the two previously received packets and uses a $(3N, N)$ code to attempt to correct the errors. In principle, there is no limit on the number of retransmissions by the transmitter until it returns to transmitting the original message. This approach is called Generalised Hybrid ARQ Type II (GH-ARQ II) and was first introduced by Morgera and Krishna [1, 12]. Note that the Type II Hybrid ARQ scheme proposed by Wang and Lin is a special case of the generalised approach.

The main advantage of GH-ARQ II is its ability to adapt to varying channel conditions to a higher degree than Type II Hybrid ARQ. However, this scheme's adaptability and gain in performance may be offset by its decoding complexity unless a suitable code is chosen for $\mathcal{C}_1$. In the worst case, a decoder is required for each error correction code: $(2N, N)$, $(3N, N)$,... (mN, N). Further, as Metzner suggested [48], it is desirable to partition the usually long packets into smaller blocks so that a simpler and shorter error correction code may be used. This makes soft-decision decoding possible [3]. This performance of this error control scheme has been analyzed by Krishna and Morgera [1] on a binary symmetric channel, but little is known about its performance on burst-noise channels. Given its adaptive characteristic, it seems likely that this GH-ARQ II error control scheme would outperform other schemes on this type of channel. The GH-ARQ II error control scheme proposed by Morgera and Krishna will be described in more detail in Section 3.1.

Figure 1.2 illustrates the differences in performance of three ARQ error control schemes. The simple ARQ scheme uses a code for error detection only, the Type I Hybrid ARQ scheme uses one code for error detection and correction (like the scheme proposed by Rocher and Pickholtz [17]), and the Type II Hybrid ARQ scheme is like the one proposed by Wang and Lin which uses two codes. At low error rates (below 10^{-5}) the throughput efficiency of the simple ARQ and Type II ARQ schemes is identical; that of the Type I Hybrid scheme is lower because the added redundancy for error correction is wasted. As the error rate increases, the efficiency of the simple ARQ scheme decreases quickly and becomes virtually useless at error rates greater than 10^{-3}. At error rates between 10^{-4} and 10^{-2}, the throughput of the Type I Hybrid ARQ scheme is greater than that of the Type II scheme. This is because the error correction coding in the Type I scheme suffices to correct most of the errors that occur during transmission, while the Type II scheme must request a retransmission since it has no inherent error correction capability on the first transmission of a packet. Finally, at very high error rates, the throughput of the Type II Hybrid ARQ scheme is superior to that of the other error control schemes because the error correction capability of its half-rate code $\mathcal{C}_1$ exceeds that of the Type I Hybrid scheme.

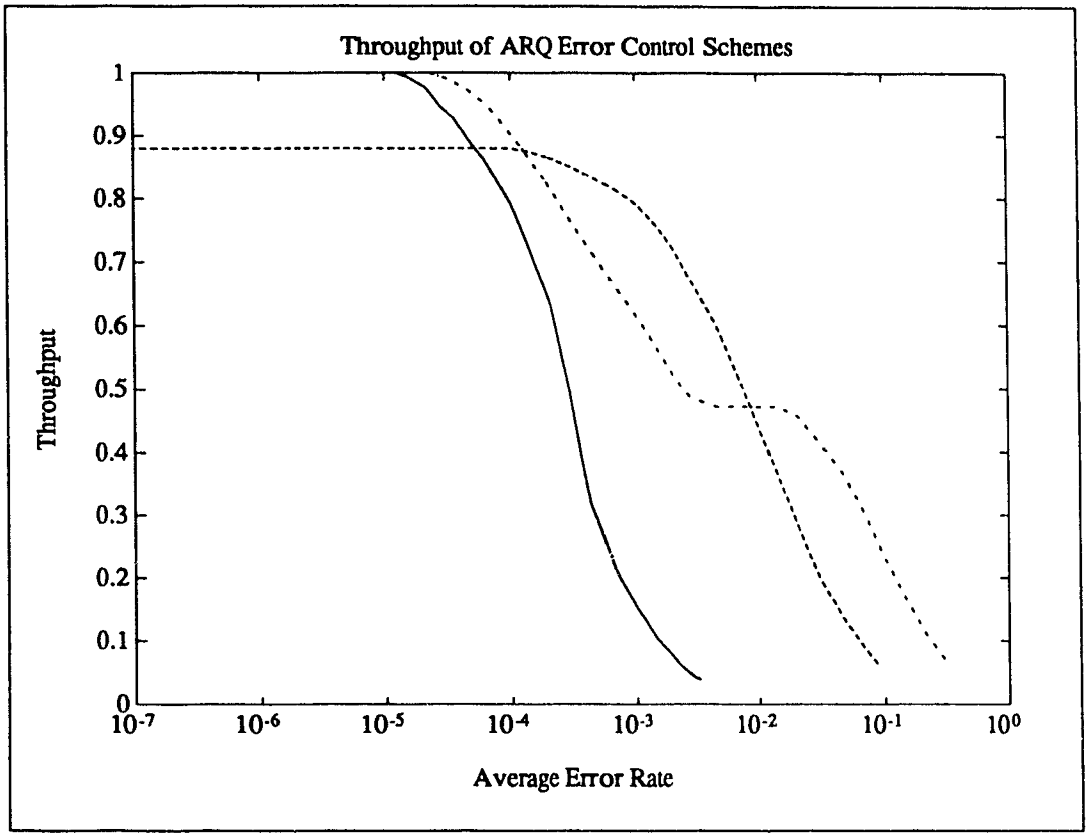

Figure 1.2: Comparison of Error Control Schemes [24]. Legend: — Simple ARQ – – – Hybrid ARQ Type I –·– Hybrid ARQ Type II.

1.3 Retransmission Strategies

The retransmission strategy in an ARQ-type system refers to the protocol that manages the utilization of the forward and feedback channels. There are three major types of retransmission strategies which are described in detail in [24]. Again, it will be seen that there is a tradeoff between the complexity of a strategy and its efficiency.

The stop-and-wait retransmission strategy, illustrated in Figure 1.3, is the simplest and least efficient. According to this strategy, the transmitter transmits a packet and then waits for the receiver's response before transmitting another one. This strategy is simple to implement, but represents a waste of the channel's resources because of the time in which it is idle. This is especially significant if the roundtrip delay is large as in a satellite channel for example.

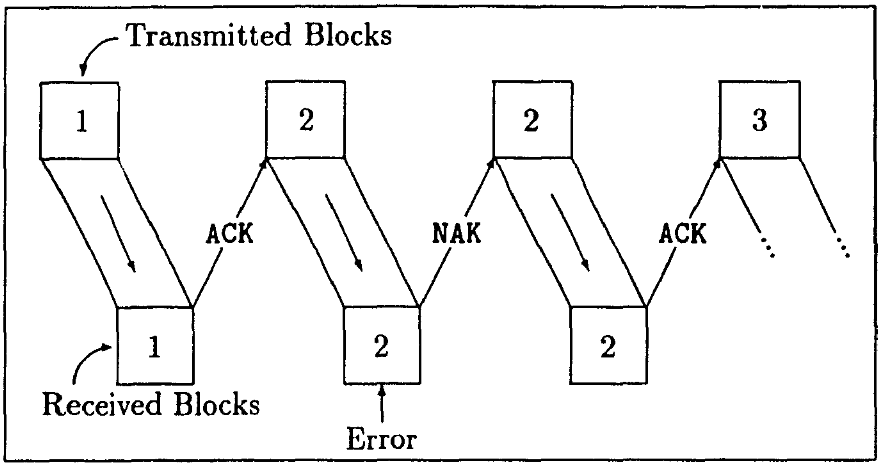

Figure 1.3: Stop and Wait Retransmission Strategy

The second retransmission strategy is Go-Back-N as shown in Figure 1.4. It is assumed N packets may be transmitted in the time that it takes for one roundtrip. In the Go-Back-N retransmission strategy, the transmitter continuously transmits packets to the receiver – it does not pause between transmissions to wait for the receiver's response as it did in the simple stop-and-wait strategy. If the transmitter receives a negative acknowledgement from the receiver, it goes back N packets and resumes transmitting from there.

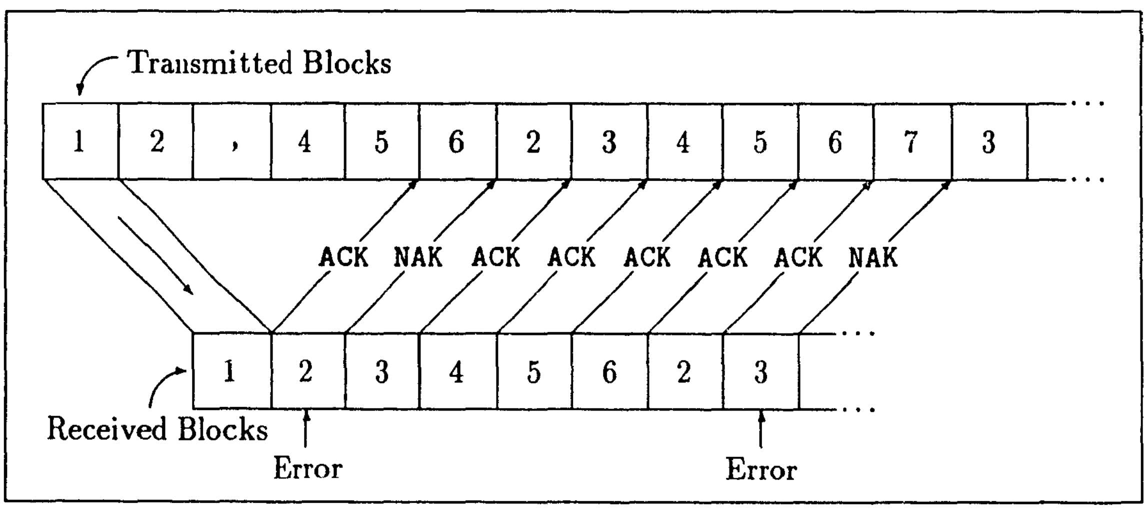

Figure 1.4: Go-Back-N Retransmission Strategy with N=5.

This approach is more complex to implement than a simple stop-and-wait retransmission strategy, yet is significantly more efficient when the channel is quiet and the roundtrip delay is large because the channel is not idle for long periods of time. However, when the channel conditions are poor, this retransmission scheme is no better than a simple stop-and-wait scheme because much of the time is spent going back N packets to retransmit packets received in error. This inefficiency is illustrated in Figure 1.4. When the first error occurs, it forces the transmitter to back up to packet #2. Notice that packets #3 to #6 were correctly received after the error occurred, but were discarded by the receiver so as to keep the packets correctly ordered at the data sink. Herein lies the inefficiency of Go-Back-N; this issue will be addressed by the next retransmission strategy.

The selective-repeat retransmission strategy is the most complex and most efficient of the three strategies that will be considered. Like the Go-Back-N strategy, the transmitter continuously transmits packets to the receiver. However, unlike Go-Back-N, when a negative acknowledgement is returned by the receiver, the transmitter retransmits only that packet which was received containing errors - not the previous N packets. The factors that limit the performance of this retransmission strategy are its requirement of a buffer at the receiver (theoretically, an infinite buffer is required) and the added overhead required to number the packets so that they are correctly ordered at the data sink. It has been found, though, that this scheme is far more efficient than the stop-and-wait and Go-Back-N retransmission strategies. This retransmission strategy, and variations of it, is analyzed in detail in [2, 43, 45, 50].

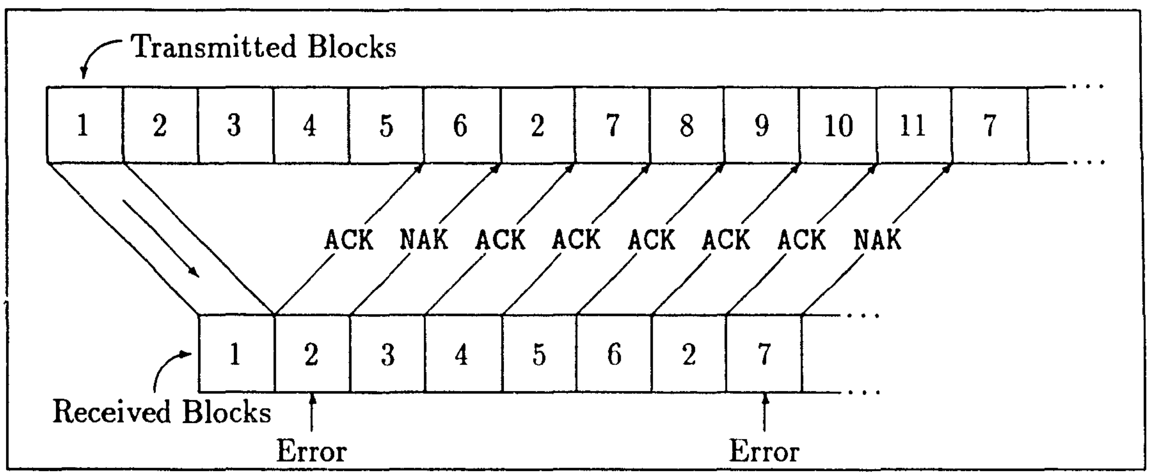

Figure 1.5: Selective Repeat Request Retransmission Strategy

1.4 In This Thesis...

Most analyses of error control schemes have modelled the channel as a simple memoryless channel. This is usually an oversimplification of reality – few channels are so easily modelled. This thesis analyzes and simulates a GH-ARQ II communications scheme when a short block code is used for error correction and the channel may be modelled as a first-order Markov chain. It is organized as follows.

In Chapter 2 the channel model is described and the performance of a simple ARQ error control scheme on this channel is analyzed. Then, the problem of integer partitioning is described and solved by using integer partitioning trees. The solution to this problem is required to make the analysis of the GH-ARQ II error control scheme more efficient.

The GH-ARQ II error control scheme is described in more detail in Chapter 3. This chapter also considers the effect of the error correction code upon the performance of the error detection code in such a scheme; two examples are provided to illustrate this effect. Further, the computation of the weight distribution of some high-rate codes is described.

The performance of the GH-ARQ II scheme on a burst-noise channel is then analyzed in Chapter 4. The receiver state transition diagram presented in this chapter is well suited to describing this type of error control scheme. The throughput and probability of undetected error are then computed in terms of the transition probabilities in the receiver state transition diagram. The analysis is then completed by computing these transition probabilities. The approach taken to analyze this GH-ARQ II error control scheme is easily extendible to higher order systems and other channels that are modelled by first-order Markov chains.

This is followed by the results of some analyses and simulations considering various GH-ARQ II schemes and channel parameters in Chapter 5. The experiments include the simulation of a GH-ARQ II system. The throughput estimated by simulation is then compared with that obtained from the preceeding analysis. Then, the effect of the roundtrip delay in the various retransmission strategies upon the performance of the GH-ARQ II error control scheme is investigated. As well, the throughput efficiency and reliability of the GH-ARQ II scheme is compared to that of a simple ARQ scheme. The effectiveness of these error control schemes is also considered on some mobile radio channels whose channel model parameters have been measured. This chapter concludes by determining the effect of the error correction code's depth upon the throughput efficiency of the GH-ARQ II schemes.

Chapter 6 concludes this thesis and suggests some further topics of interest. Appendix A describes an approximation that significantly decreases the complexity of the throughput efficiency computation. Then, it describes the analysis modifications that are required to account for error correction codes consisting of non-invertible generator matrices. This appendix concludes by extending the performance analysis of the GH-ARQ II error control scheme to depth-3 and depth-4 systems. Appendix B discusses some issues related to the Gilbert-Elliott burst-noise channel model. Finally, Appendix C summarizes the error correction capabilities of some selected KM codes.

Chapter 2

Preliminary Issues

This chapter describes a burst-noise channel model as well as its related parameters. The performance of a simple ARQ error control scheme on this channel is then analyzed. Finally, the integer partioning problem and a method to enumerate the partitions of an integer are presented. This problem will arise when the performance of the GH-ARQ II error control scheme is analyzed and its solution decreases the computational complexity of this analysis.

2.1 The Channel Model

Channel models, in general, fall into one of two categories: generative or descriptive. *Generative* models are those that attempt to describe a model which generates error sequences with similiar properties[1] to those generated by the real channel. Note that the *error sequence* $\underline{e}$ is the sequence produced by the modulo-2 addition of the transmitted and received packets, $\underline{t}$ and $\underline{r}$ respectively. *Descriptive* models describe the structure of the the real channel's error sequence by a suitable set of statistics. In this thesis, only generative models are considered; descriptive models are discussed in more detail in [13].

The simplest channel model is the *Binary Symmetric Channel* (BSC) shown in Figure 2.1. This model is memoryless in the sense that errors occur at random intervals with constant probability p. Thus, the properties of this model are simply summarized by p. Unfortunately, the BSC is not adequate to describe the characteristics of several real channels of interest.

[1]For example, these properties might be the average error rate or the error gap distribution of the channel in question [13].

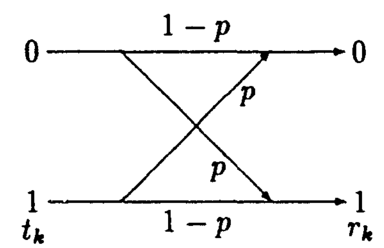

Figure 2.1: The Binary Symmetric Channel. The k-th bit in the transmitted and received packets is denoted by t_k and r_k respectively.

Real channels tend to have some memory which causes errors to occur in bursts. These may be caused, for example, by radio static, switching transients or varying weather conditions [10]. Unlike the errors that occur in the BSC, it is in general very difficult to describe the statistics of the error bursts. Collectively, channels in which the errors occur in bursts are referred to as *burst noise channels* or *fading channels*.

The first model to adequately describe the behavior of some fading channels was proposed by Gilbert in 1960 [10]. The Gilbert model consists of a two-state first-order Markov chain. When the channel is in the "Good" state, no errors occur; while in the "Bad" state, errors occur with probability ϵ_b. In effect, the bad state is a BSC with probability of error equal to ϵ_b. The transitions between the two states are determined by the transition probabilities from the good to bad state P_{gb} and from the bad to good state P_{bg}. Since the channel is in the bad state with probability $\frac{P_{gb}}{P_{gb}+P_{bg}}$, and since errors can only occur in this state with probability ϵ_b, the overall average probability of error $\bar{\epsilon}$ is $\frac{P_{gb}\epsilon_b}{P_{gb}+P_{bg}}$.

In this thesis, the burst-noise channel model that will be considered is the extension of the Gilbert model proposed by Elliott [11] in 1963 and shown in Figure 2.2. It accounts for the possibility that errors can occur in the good state with probability ϵ_g. Typically $\epsilon_b \gg \epsilon_g$. Let the current state of the channel be denoted by Ω_i. The average probability of error $\bar{\epsilon}$ in this model may then be computed as follows:

$$\begin{aligned}
\bar{\epsilon} &= \Pr(\text{Error}, \Omega_i = \text{G}) + \Pr(\text{Error}, \Omega_i = \text{B}) \\
&= \Pr(\text{Error} \mid \Omega_i = \text{G}) \Pr(\Omega_i = \text{G}) + \Pr(\text{Error} \mid \Omega_i = \text{B}) \Pr(\Omega_i = \text{B}) \\
&= \frac{P_{gb}\epsilon_b + P_{bg}\epsilon_g}{P_{gb} + P_{bg}},
\end{aligned}$$

where the results that $\Pr(\text{State G}) = \frac{P_{bg}}{P_{gb}+P_{bg}}$ and $\Pr(\text{State B}) = \frac{P_{gb}}{P_{gb}+P_{bg}}$ are used and are derived in Appendix B.

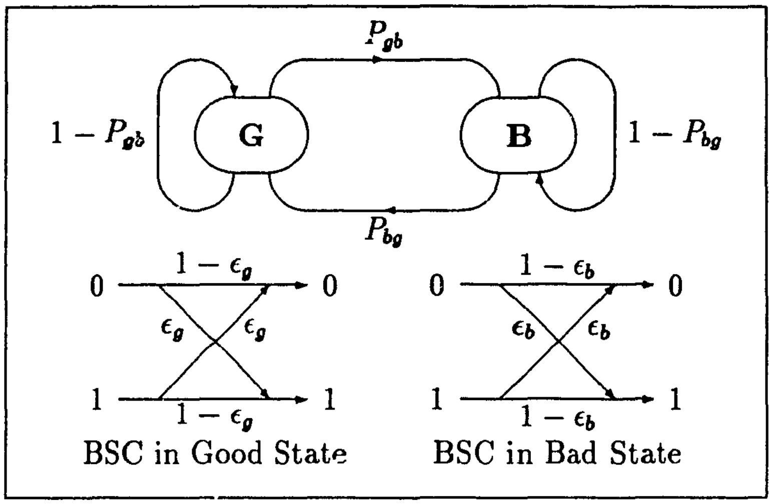

Figure 2.2: The Gilbert-Elliott Model for Burst-Noise Channels

The "burstiness" of the Gilbert-Elliott model may be more explicitly described by an alternative set of parameters as proposed by Lugand *et al.* [4]. This approach qualifies the bursts as diffuse or dense. A *dense burst* channel is characterized by infrequent bursts of several errors, while a *diffuse burst* is distinguished by frequent bursts of few errors.

In order to quantify the burstiness of this channel model, several definitions are required. The *average burst length* $\bar{b}$ is the average number of packets transmitted while the channel is in the bad state and is simply $\bar{b} = 1/P_{bg}$. Let the probability that the channel model is in the bad state be denoted by P_b, and finally, define the high-to-low bit error rate ratio to be $\rho = \epsilon_b/\epsilon_g$. The Gilbert-Elliott model may then be completely described by the four parameters $\bar{b}$, P_b, $\bar{\epsilon}$, and ρ instead of P_{gb}, P_{bg}, ϵ_g, and ϵ_b. It is now possible to describe the burstiness of this channel model more concretely. A dense burst channel is one in which P_b is small and ρ is large, while a diffuse burst channel is one in which P_b is large and ρ is slightly greater than unity. The limiting case of the diffuse burst channel model is the BSC in which $\epsilon_g = \epsilon_b$ ($\rho = 1$); the transition probabilities in this case are irrelevant.

In order for this channel model to be qualified as previously described, it must satisfy four conditions as described in [4]. The one additional constraint required in order for the model to meet these conditions is that $\epsilon_g = \bar{\epsilon}P_b$. This constraint decreases by one the number of degrees of freedom in this channel model and simplifies the analysis of the performance of error control schemes on this channel model. Therefore, the parameters $\bar{b}$, P_b and $\bar{\epsilon}$ with the constraint $\epsilon_g = \bar{\epsilon}P_b$ completely describe the Gilbert-Elliott channel model. The original channel model parameters may then be computed from these three parameters by using the following expressions:

$$\begin{aligned} P_{bg} &= \frac{1}{\overline{b}} \\ P_{gb} &= \frac{P_{bg}P_b}{1-P_b} \\ \epsilon_g &= \overline{\epsilon}P_b \\ \epsilon_b &= \begin{cases} \frac{\overline{\epsilon}}{P_b} - (1-P_b)\overline{\epsilon} \text{ for } P_b > \frac{\overline{\epsilon}+\frac{1}{2}-\sqrt{(\frac{1}{2}-\overline{\epsilon})(\frac{1}{2}+3\overline{\epsilon})}}{2\overline{\epsilon}} \\ \frac{1}{2}, \text{ otherwise} \end{cases} . \end{aligned}$$

2.2 Analysis of a Simple ARQ Scheme

In this section, a simple ARQ scheme is described and its performance on a burst-noise channel is analyzed. Suppose that the message to be communicated $\underline{I}$ is a k-bit block denoted by a vector $\underline{I} = (i_0, \ldots, i_{k-1})$. This message is encoded with an error detection block code C to produce an N-bit packet $\underline{t}$ which is suitable for transmission. Since this code is assumed to be a block code, it may be described in general by a $(k \times N)$ *generator matrix*, $\boldsymbol{G}$. The encoding process is then simply described by $\underline{t} = \underline{I}\boldsymbol{G}$, assuming that the k-bit message is a $(1 \times k)$ vector. Let the *weight distribution* of this code C be denoted by the set $\{A_k : k = 0, 1, \ldots, N\}$, where A_k is the number of codewords in C with weight k.

The receiver checks for the presence of errors in the received packet by computing its *syndrome*. The *parity check matrix*, denoted by $\boldsymbol{H}$ is an $(N-k \times N)$ matrix with the property that $\boldsymbol{G}\boldsymbol{H}^T = \mathbf{0}$, where $\mathbf{0}$ is the $(k \times (N-k))$ zero matrix. The syndrome $\underline{S}$ of the received packet is then

$$\underline{S} = \underline{r}\boldsymbol{H}^T.$$

If $\underline{S} = \underline{0}$, where $\underline{0}$ is the zero vector, then the receiver decides that no errors have occurred and delivers the recovered message $\tilde{\underline{I}}$ to the data sink. Otherwise, if $\underline{S} \neq \underline{0}$ the receiver requests that the transmitter retransmit the packet $\underline{t}$. This process continues until a packet with zero syndrome is received. The computation of the syndrome may be illustrated as shown in Figure 2.3.

If the error sequence added by the channel is equal to a codeword in the error detection code, the receiver will deliver a message $\tilde{\underline{I}}$ containing errors to the data sink. This is because the syndrome of the error sequence, $\underline{S}_e = \underline{e}\boldsymbol{H}^T$, is $\underline{0}$ and, thus, is undetectable.

The throughput efficiency of this scheme is defined to be the reciprocal of the average number of transmissions required before the receiver positively acknowledges reception of the data. The reliability of the simple ARQ scheme is defined to be the probability that the receiver delivers a packet containing errors to the data sink.

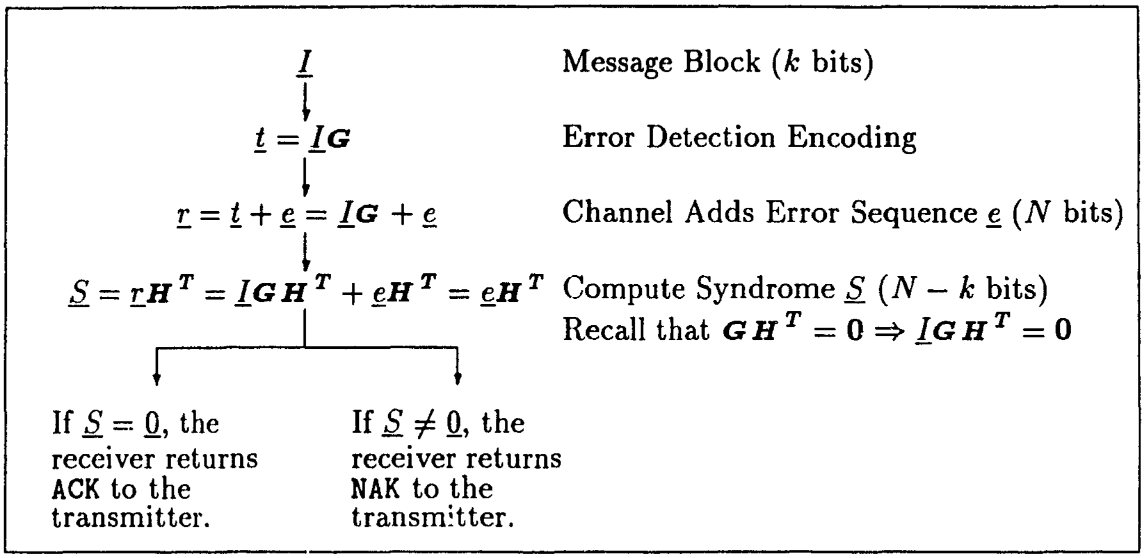

Figure 2.3: Syndrome Computation in the Simple ARQ Error Control Scheme

Assume that the transitions between the states of the channel model occur only between transmissions of packets. This assumption is common [4, 6] and ensures that the error rate during the transmission of a packet is constant. Let,

$$\begin{aligned} C &= \Pr(\underline{S} = \underline{0}, \underline{e} = \underline{0}) \\ Q &= \Pr(\underline{S} = \underline{0}, \underline{e} \neq \underline{0}) \\ P &= \Pr(\underline{S} \neq \underline{0}), \end{aligned}$$

where, as before, $\underline{S}$ is the syndrome of the received packet. These probabilities may be interpreted as follows. The quantity C is the probability of receiving an error-free transmission, Q is the probability of receiving an error sequence equal to a non-zero codeword in the error detection code $\mathcal{C}$, and, finally, P is the probability of receiving a message containing a detectable error sequence.

By definition, the average number of transmissions before the receiver accepts the transmitted packet, E[Number of Transmissions], is

$$\text{E[Number of Transmissions]} = \sum_{i=1}^{\infty} i \Pr(i \text{ Transmissions}).$$

Since $\Pr(i \text{ Transmissions}) = P^{i-1}(1-P)$,

$$\begin{aligned} \text{E[Number of Transmissions]} &= \sum_{i=1}^{\infty} i P^{i-1}(1-P) \\ &= \sum_{i=0}^{\infty}(i+1)P^i - \sum_{i=1}^{\infty} i P^i \end{aligned}$$

$$
\begin{aligned}
\text{E[Number of Transmissions]} &= 1+\sum_{i=1}^{\infty}[(i+1)P^i - iP^i] \\
&= \sum_{i=0}^{\infty} P^i \\
&= \frac{1}{1-P} \quad \text{assuming } P<1.
\end{aligned}
$$

The throughput, $\eta(\mathbf{ARQ})$, of this simple ARQ scheme is then,

$$\eta(\mathbf{ARQ}) = 1 - P. \tag{2.1}$$

The reliability (probability of undetected error) of this scheme $P_{ud}(\mathbf{ARQ})$ may be computed in a similar manner. That is,

$$
\begin{aligned}
P_{ud}(\mathbf{ARQ}) &= \sum_{i=1}^{\infty} \Pr(\text{Undetected Error on } i\text{-th Transmission}) \\
&= \sum_{i=1}^{\infty} Qi \Pr(i \text{ Transmissions}) \\
&= Q\sum_{i=0}^{\infty} P^i \\
&= \frac{Q}{1-P}, \quad \text{assuming } P<1.
\end{aligned}
$$

Therefore, the reliability of the simple ARQ error control scheme is given by

$$P_{ud}(\mathbf{ARQ}) = \frac{Q}{1-P}. \tag{2.2}$$

There remains the computation of the probabilities C, Q and P, which is easily accomplished as shown below. Let Ω_i be the current state of the channel. Then,

$$
\begin{aligned}
C &= \Pr(\underline{S}=\underline{0}, \underline{e}=\underline{0}) \\
&= \Pr(\underline{S}=\underline{0}, \underline{e}=\underline{0}, \Omega_i = \mathrm{G}) + \Pr(\underline{S}=\underline{0}, \underline{e}=\underline{0}, \Omega_i = \mathrm{B}) \\
&= \Pr(\underline{S}=\underline{0}, \underline{e}=\underline{0} \mid \Omega_i = \mathrm{G}) \Pr(\Omega_i = \mathrm{G}) \\
&\quad + \Pr(\underline{S}=\underline{0}, \underline{e}=\underline{0} \mid \Omega_i = \mathrm{B}) \Pr(\Omega_i = \mathrm{B}) \\
&= (1-\epsilon_g)^N \frac{P_{bg}}{P_{gb}+P_{bg}} + (1-\epsilon_b)^N \frac{P_{gb}}{P_{gb}+P_{bg}} \\
C &= \frac{1}{P_{gb}+P_{bg}} \left[P_{bg}(1-\epsilon_g)^N + P_{gb}(1-\epsilon_b)^N\right].
\end{aligned}
$$

Let d be the minimum distance of the error detection code $\mathcal{C}$. Then,

$$\begin{aligned}
Q &= \Pr(\underline{S} = \underline{0}, \underline{e} \neq \underline{0}, \Omega_i = \mathrm{G}) + \Pr(\underline{S} = \underline{0}, \underline{e} \neq \underline{0}, \Omega_i = \mathrm{B}) \\
&= \Pr(\underline{S} = \underline{0}, \underline{e} \neq \underline{0} \mid \Omega_i = \mathrm{G}) \Pr(\Omega_i = \mathrm{G}) \\
&\quad + \Pr(\underline{S} = \underline{0}, \underline{e} \neq \underline{0} \mid \Omega_i = \mathrm{B}) \Pr(\Omega_i = \mathrm{B}) \\
&= \frac{P_{bg}}{P_{gb} + P_{bg}} \sum_{k=d}^{N} A_k \epsilon_g^k (1 - \epsilon_g)^{N-k} + \frac{P_{gb}}{P_{gb} + P_{bg}} \sum_{k=d}^{N} A_k \epsilon_b^k (1 - \epsilon_b)^{N-k} \\
&= \frac{1}{P_{gb} + P_{bg}} \sum_{k=d}^{N} A_k \left[P_{bg} \epsilon_g^k (1 - \epsilon_g)^{N-k} + P_{gb} \epsilon_b^k (1 - \epsilon_b)^{N-k} \right].
\end{aligned}$$

Finally, since $C + P + Q = 1$, the probability $P = \Pr(\underline{S} \neq \underline{0})$ is simply

$$P = 1 - C - Q.$$

Substitution of these expressions for C, Q and P into the expressions for the throughput and reliability (Equations 2.1 and 2.2 respectively) completes the performance analysis of this simple ARQ error control scheme on a burst-noise channel.

$$\begin{aligned}
\eta(\mathrm{ARQ}) = \frac{1}{P_{gb} + P_{bg}} \Bigg\{ & P_{bg}(1 - \epsilon_g)^N + P_{gb}(1 - \epsilon_b)^N \\
& + \sum_{k=d}^{N} A_k \left[P_{bg} \epsilon_g^k (1 - \epsilon_g)^{N-k} + P_{gb} \epsilon_b^k (1 - \epsilon_b)^{N-k} \right] \Bigg\}
\end{aligned}$$

$$P_{ud}(\mathrm{ARQ}) = \frac{\sum_{k=d}^{N} A_k \left[P_{bg} \epsilon_g^k (1 - \epsilon_g)^{N-k} + P_{gb} \epsilon_b^k (1 - \epsilon_b)^{N-k} \right]}{P_{bg}(1 - \epsilon_g)^N + P_{gb}(1 - \epsilon_b)^N + \sum_{k=d}^{N} A_k \left[P_{bg} \epsilon_g^k (1 - \epsilon_g)^{N-k} + P_{gb} \epsilon_b^k (1 - \epsilon_b)^{N-k} \right]}$$

2.3 The Integer Partition Problem

This section describes the general problem of integer partitioning which will be an issue when the GH-ARQ II error control scheme is analyzed. Further, it presents a method to enumerate the partitions of a given integer along with an example of it. Comments regarding the correctness and complexity of this enumeration method are then offered.

2.3.1 Description of the Problem

Consider the following problem. Given a packet of N bits with Hamming weight w and comprised of N/n blocks, enumerate the number of ways in which the weight w may be distributed (or partitioned) among the blocks. For example, suppose that a 6-bit packet comprised of 3 blocks has Hamming weight equal to 2; so $w = 2$, $N = 6$ and $n = 2$. There are two ways in which to partition the Hamming weight among the blocks in the packet. These are illustrated in Figure 2.4.

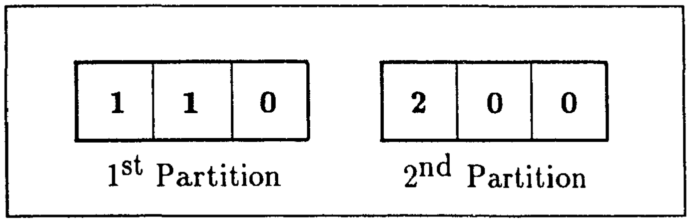

Figure 2.4: Example of Partitions

These partitions satisfy the following constraints:

- The Hamming weight of the packet is w.
- The Hamming weight of any given block does not exceed n.
- The order of the blocks is unimportant. That is, the partitions **1 1 0** and **1 0 1** are considered equivalent.

The problem then, is to efficiently enumerate these partitions when w, N and n are large enough so that simple inspection is infeasible.

The enumeration of these partitions is an issue in the broader integer partitioning problem which is summarized here and described in detail in [23]. Suppose w is a non-negative integer and that there exist integers α_i, $i = 1, \ldots, K$ ($K \geq 1$), such that $\alpha_K \geq \ldots \geq \alpha_2 \geq \alpha_1 > 0$. If $w = \sum_{i=1}^{K} \alpha_i$, then $\alpha = \{\alpha_1, \ldots, \alpha_K\}$ is a K-part partition of w. The enumeration of the partitioning of $w = 7$ is given in Table 2.1.

Regarding the integer partitioning problem in the present context, two issues will be addressed. The first is the number of partitions of an integer and the second is the description of a method to systematically enumerate these partitions.

Let $p(w)$ denote the number of partitions of an integer w. Further, let $[q^w]$ be an operator that returns the w-th coefficient of the polynomial upon which it operates so that if

$$f(q) = \sum_{i=0}^{n} a_i q^i = a_n q^n + a_{n-1} q^{n-1} + \ldots + a_1 q + a_0,$$

1-part	{7}
2-part	{1,6} {2,5} {3,4}
3-part	{1,1,5} {1,2,4} {1,3,3} {2,2,3}
4-part	{1,1,1,4} {1,1,2,3} {1,2,2,2}
5-part	{1,1,1,1,3} {1,1,1,2,2}
6-part	{1,1,1,1,1,2}
7-part	{1,1,1,1,1,1,1}

Table 2.1: Enumeration of the Partitions of $w = 7$

then

$$[q^w]f(q) = \left.\frac{\partial^w f(q)}{\partial q^w}\right|_{q=0} = a_w, \text{ for } 0 \leq w \leq n.$$

The number of partitions of the integer w is then given by [23]

$$p(w) = [q^w]\prod_{i=1}^{w}(1 - q^i)^{-1}.$$

The problem of enumerating these partitions will be the subject of the following section.

2.3.2 Construction of an Integer Partition Tree

The solution that was adopted to enumerate the partitions of an integer consists in constructing a tree, the *integer partition tree*[2], so that valid paths through the tree form partitions of the integer w. Begin by assuming that the maximum value of any α_i in the partition α is n (the number of bits in a block). Also, since there are N/n blocks in a packet, it is required to enumerate all partitions up to and including N/n-parts; that is $K \leq N/n$. To construct this tree, begin by assigning its root a cumulative weight w_t of zero. Now, add n leaves to the root. The cumulative weight of each leaf is then its index number. This is illustrated in Figure 2.5.

Each leaf of the tree is tested to ensure that it is still consistent with the requirements of a valid partition:

1. Its total cumulative weight w_t must be less than or equal to the integer being partitioned w. That is $w_t \leq w$.

2. The number of levels in the weight partition tree must be less than or equal to the number of blocks in the packet N/n. Note that the root is not counted as

[2]Since the integer in question is usually the Hamming weight of a packet, the term "Integer Partition Tree" is used interchangeably with "Weight Partition Tree".

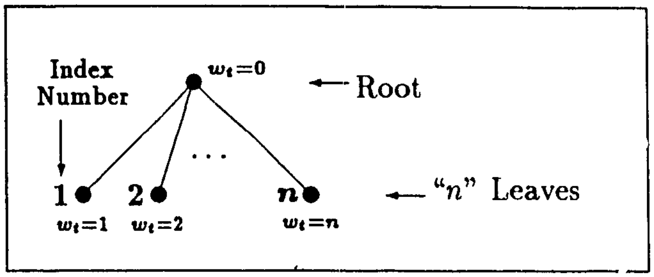

Figure 2.5: First Level of an Integer Partition Tree

a level in the tree. This ensures that no partition with more than N/n parts is generated.

If it is found that a leaf has total cumulative weight w_t greater than w, then it is eliminated from further "growth". If a leaf has total cumulative weight equal to the target integer w, then the path from it to the root is a valid partition of w. Finally, in the case that the leaf's total cumulative weight is less than the target weight, the growth is continued at least one more level. It is interesting to note that the j-th leaf on the first level will enumerate all those partitions with $\alpha_1 = j$.

The growth to a second level is illustrated in Figure 2.6.

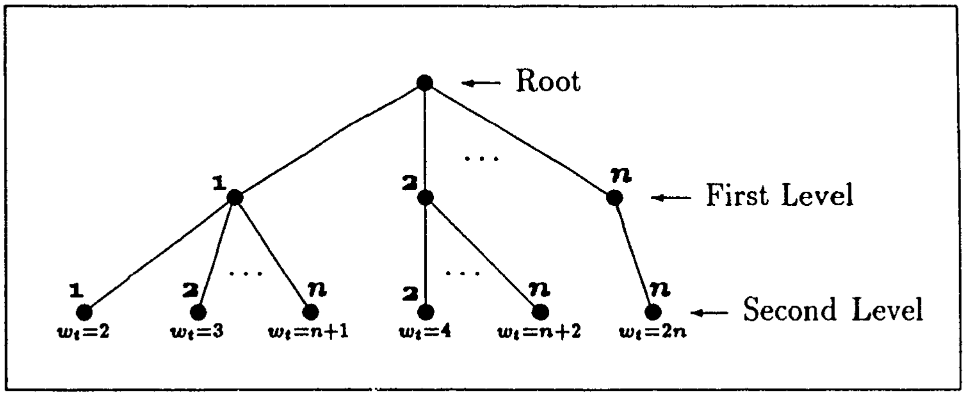

Figure 2.6: Second Level of an Integer Partition Tree

Note that the second level is extended from the first level in such a way that equivalent partitions will not be generated. Leaf **1** on the first level is extended by n leaves numbered $1 \ldots n$, leaf **2** on the first level is extended by $n-1$ leaves numbered $2 \ldots n$ and leaf **n** on the first level is extended by 1 leaf numbered n. In general, child $\boldsymbol{j}$ on the first level is extended by $n - j + 1$ leaves numbered $j \ldots n$ which ensures that $\alpha_2 \geq \alpha_1$. This procedures applies to the succeeding levels as well.

The growth of the integer partition tree is complete when no leaf may be extended in such a way that the previously outlined conditions are still respected. When this happens, the paths from those leaves with $w_t = w$ to the root enumerate the partitions of the integer w.

The correctness of this procedure to enumerate the partitions of an integer is easily demonstrated by recalling the definition of a partition; namely that the parts α_i should be such that

$$\alpha_K \geq \ldots \geq \alpha_2 \geq \alpha_1 > 0.$$

This is exactly how the partitions are generated with the integer partition trees. The first level in the partition tree generates α_1, the second level generates α_2 and in general the j-th level ($j \leq N/n$) generates α_j. Since a leaf on the i-th level may generate only children with index number greater than or equal to its index number, this ensures that $\alpha_{i+1} \geq \alpha_i$. Consequently, the integer partioning trees as previously described are a simple interpretation of the definition of the original integer partitioning.

The comnlexity of this scheme is an important consideration. If there is no restriction on the size and number of parts in a partition, then the complexity of any scheme is lower bounded by the number of partitions $p(w)$ of the integer w. However, given the restrictions that no part shall be greater than t and no partition shall be comprised of more than N/n parts, this constrains the number of allowable partitions of an integer w to be less than or equal to $p(w)$. It is, in general, a very difficult problem to determine the number of nodes that are generated in a partition tree when enumerating the partitions of an integer w.

The number of partitions enumerated by the weight partition tree is the number of terminal leaves in the tree. Note that not necessarily all of these partitions are valid, yet these must all be counted because they contribute to the complexity of the enumeration. Suppose that no part in any valid partition of an integer w may be greater than t and that no partition may contain more than N/n parts. Consequently, there are t nodes on the first level of the integer partition tree. The number of nodes on the second level is exactly

$$\sum_{i=1}^{t} i = \frac{t(t+1)}{2},$$

which can be upper bounded by t^2 (since $t \geq 1$). This upper bound may be obtained alternatively by imagining that each node on the first level is extended by t children to form the second level. Hence, the second level would contain t^2 children. In general, the l-th level of the tree would then contain at most t^l nodes. Since an integer partition tree may contain at most N/n levels, the upper bound on the number of terminal leaves is $t^{N/n}$. It has been found, though, that this bound is very loose especially when t and N/n are large.

2.3.3 An Example

This section presents an example of the enumeration of the integer partitions using an integer partition tree. Suppose that the Hamming weight of the packet is 6 and that there are $N = 12$ bits in a packet with $n = 3$ bits per block. In the language of integer partitioning, this example would be interpreted as the partitioning of the integer 6 such that no partition contains more than $N/n = 4$ parts and no part α_i is greater th◡n $n = 3$. The completed weight partition tree is illustrated in Figure 2.7.

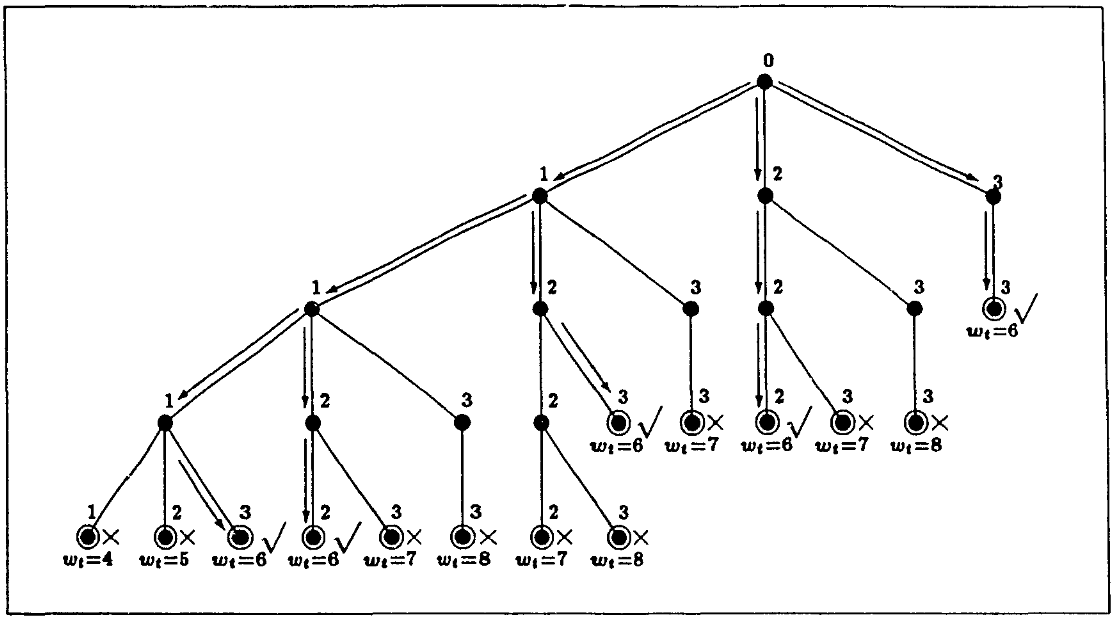

Figure 2.7: Example of an Integer Partition Tree

From the integer partition tree in Figure 2.7 it is seen that there are five partitions marked $\surd$ which satisfy the constraints $N = 12$, $n = 3$ and $w = 6$. These partitions are:

$$\begin{aligned} \mathcal{P}_1 &= \{1,1,1,3\} \\ \mathcal{P}_2 &= \{1,1,2,2\} \\ \mathcal{P}_3 &= \{1,2,3\} \\ \mathcal{P}_4 &= \{2,2,2\} \\ \mathcal{P}_5 &= \{3,3\} \end{aligned}$$

These partitions may be expressed more efficiently in terms of $\pi_j^{(k)} = (\pi_{j0}^{(k)}, \ldots, \pi_{jn}^{(k)})$, such that $\pi_{ji}^{(k)}$ is the number of parts equal to i in the j-th partition of the integer k.

Using this notation, the partitions for this example may be expressed as

$$\begin{aligned}
\mathcal{P}_1 &: \quad \pi_1^{(6)} = (0,3,0,1) \\
\mathcal{P}_2 &: \quad \pi_2^{(6)} = (0,2,2,0) \\
\mathcal{P}_3 &: \quad \pi_3^{(6)} = (1,1,1,1) \\
\mathcal{P}_4 &: \quad \pi_4^{(6)} = (1,0,3,0) \\
\mathcal{P}_5 &: \quad \pi_5^{(6)} = (2,0,0,2)
\end{aligned}$$

This notation will be convenient when the transition probabilities in the receiver state transition diagram of the GH-ARQ II system are computed in Chapter 4. Also, notice that these partitions satisfy the following constraints:

1. The number of parts in the partition of k should be equal to the number of blocks in the packet N/n. Note that the number of parts equal to zero should be included in the counting of these parts. That is,

$$\sum_{i=0}^{n} \pi_{j_i}^{(k)} = N/n.$$

2. The sum of the parts of the partition of the integer k must equal k. That is,

$$\sum_{i=1}^{n} i\pi_{j_i}^{(k)} = k.$$

Chapter 3

Description of a GH-ARQ II Scheme

This chapter presents some issues regarding the GH-ARQ II error control scheme which will be analyzed in the following chapter. It begins by describing this error control scheme in some detail. This is followed by a discussion of the effect of the error correction encoding upon the error detection code in the GH-ARQ II scheme. Finally, the weight distribution of dual, linearly altered and shortened codes is discussed.

3.1 Overview of the GH-ARQ II Scheme

In Section 1.2, the GH-ARQ II error control scheme was briefly described; this section will now describe it in more detail. The GH-ARQ II scheme that will be considered in this thesis utilizes two codes: C_0 is an (N, k) block code used for error detection only, while C_1 is an (mn, n) block code used for error correction only. The code C_1 is a key factor in the complexity of this type of system. Its generator matrix has the form

$$\boldsymbol{M} = [\boldsymbol{M}_1 | \; \boldsymbol{M}_2 | \ldots | \; \boldsymbol{M}_m \;],$$

where m is referred to as the *depth* of the code C_1 and $\boldsymbol{M}_i$ is an $(n \times n)$ square matrix. Let $C_1^{(x)}$ be generated by $\boldsymbol{M}^{(x)} = [\boldsymbol{M}_1 | \ldots | \boldsymbol{M}_x \;]$ and d_x be its corresponding minimum distance. In order for the code C_1 to be useful, it should have the property that $d_i < d_j$ for $i < j$. This will ensure that the minimum distance, and hence the error correction capability, of C_1 increases progressively with each retransmission. It is assumed that $\boldsymbol{M}_1$ is invertible. Otherwise, as will be seen in the description of this scheme that follows, the first transmission is ineffective because the message cannot be recovered. This requirement is not strictly necessary for $\boldsymbol{M}_i$, $i = 2, \ldots, m$.

Consider a depth-3 system so that $\boldsymbol{M} = [\boldsymbol{M}_1 | \boldsymbol{M}_2 | \boldsymbol{M}_3]$. The encoding process at the transmitter begins by encoding the k-bit message $\underline{I}$ with the error detection code $\mathcal{C}_0$. The resulting N-bit packet[1] $(\underline{I}, \underline{Q})$ is then coded with the first submatrix $\boldsymbol{M}_1$ of the error correction code's generator matrix $\boldsymbol{M}$. Let

$$\boldsymbol{G}_1 = \boldsymbol{M}_1 \otimes \boldsymbol{I}_{\frac{N}{n}},$$

where $\otimes$ denotes the Kronecker product of two matrices, and $\boldsymbol{I}_{\frac{N}{n}}$ is the identity matrix of order N/n [1]. The encoding by $\boldsymbol{M}_1$ is then completed by multiplying the N-bit packet $(\underline{I}, \underline{Q})$ by $\boldsymbol{G}_1$; the effect of this multiplication by $\boldsymbol{G}_1$ is to subdivide the packet $(\underline{I}, \underline{Q})$ into n-bit blocks and then to multiply each block by $\boldsymbol{M}_1$. The resulting packet $\underline{t}^{(1)}$ is ready for transmission.

Since $\boldsymbol{M}_1$ is invertible, the receiver begins by inverting the received packet $\underline{r} = (\hat{\underline{I}}, \hat{\underline{Q}})$ to remove the effect of the encoding by $\boldsymbol{M}_1$. The inverted packet is then checked for errors with the error detection code $\mathcal{C}_0$. If no errors are detected, the recovered information $\tilde{\underline{I}} = \hat{\underline{I}}$ is delivered to the data sink and the transmission is positively acknowledged. Otherwise, the transmission is negatively acknowledged and the received packet is stored in preparation for the next transmission. Note that there is no attempt to correct errors on the first transmission because no parity bits have been included for this purpose.

Upon receiving a negative acknowledgement from the receiver, the transmitter will encode the packet $(\underline{I}, \underline{Q})$ with $\boldsymbol{M}_2$ of the generator matrix $\boldsymbol{M}$ to form $\underline{t}^{(2)}$. When the receiver receives this packet from the transmitter it begins, again, by removing the effect of the encoding by $\boldsymbol{M}_2$ (if it is invertible) and then checking for errors with the error detection code $\mathcal{C}_0$. If no errors are detected, then the recovered information $\tilde{\underline{I}}$ is delivered to the data sink, the receiver's buffer is emptied of the previously received packet and the transmission is positively acknowledged.

If errors are detected, the receiver combines the currently received packet $\underline{r}^{(i)}$ with the previously received one $\underline{r}^{(i-1)}$, as shown in Figure 3.1, and attempts to correct the errors by using the error correction code $\mathcal{C}_1^{(2)}$ generated by $[\boldsymbol{M}_1 \mid \boldsymbol{M}_2]$. This is accomplished by combining the j-th block of the previously received packet $\mathcal{B}_j^{(i-1)}$ with the j-th block of the currently received packet $\mathcal{B}_j^{(i)}$ to form the combined block $[\mathcal{B}_j^{(i-1)} \mid \mathcal{B}_j^{(i)}]$ which is then decoded with the code $\mathcal{C}_1^{(2)}$. The result of the decoding is the j-th n-bit block $\mathcal{B}_j^{(o)}$ of the N-bit decoded packet. Note that this decoding may be performed independently in parallel for each of the N/n blocks in the packets. The decoded packet is then checked for errors once again with the error detection code.

If the receiver was unable to successfully correct the errors in the currently received

[1] **This description assumes that $\mathcal{C}_0$ is systematic; this assumption is not stringent and is made for notational convenience only.**

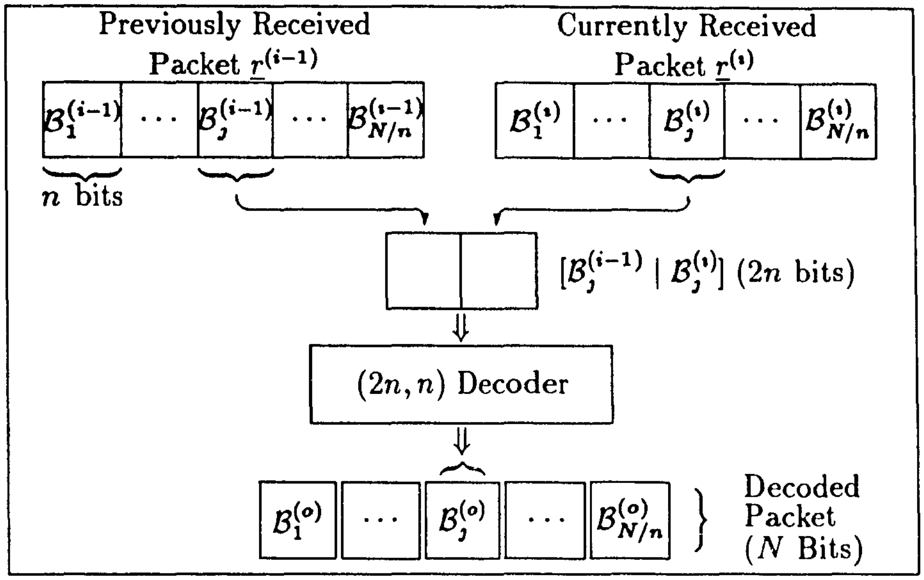

Figure 3.1: Combining Two Packets for Error Correction.

packet, the transmission is negatively acknowledged and the packet is stored with the previous one in preparation for the next transmission. The transmitter encodes $(\underline{I}, \underline{Q})$ with $\boldsymbol{G}_3 = \boldsymbol{M}_3 \otimes \boldsymbol{I}_{\frac{N}{n}}$ and transmits the resulting packet $\underline{t}^{(3)}$. Upon reception of this packet, the inversion and error detection process is initiated if $\boldsymbol{M}_3$ is invertible. If the inversion is unsuccessful or not possible, then this packet will be combined with the two previously transmitted packets in a manner similar to that illustrated in Figure 3.1. The attempt to correct the errors is accomplished by combining the j-th block in the current packet with the j-th blocks in the two previously received packets to form $[\mathcal{B}_j^{(i-2)} \mid \mathcal{B}_j^{(i-1)} \mid \mathcal{B}_j^{(i)}]$ and then decoding with the code $\mathcal{C}_1^{(3)}$ which is generated by $[\boldsymbol{M}_1 \mid \boldsymbol{M}_2 \mid \boldsymbol{M}_3]$. After decoding, the receivers checks for errors. If the decoding fails, the transmission is negatively acknowledged, the currently received packet is stored and the least recently received packet is discarded from the receiver's memory.

On the fourth transmission of the message, the transmitter returns to $\boldsymbol{M}_1$ of the error correction code $\mathcal{C}_1$ The inversion at the receiver proceeds as it did on the first transmission but the error correction decoding uses $\mathcal{C}_1^{(3)}$ to attempt to correct the errors in the received packet if the inversion is not possible or unsuccessful.

From the preceeding description of this GH-ARQ II error control scheme, it is seen that the error correction capability *adapts* to the channel conditions. The parity bits for error correction are transmitted as they are required. For poor channel conditions, this scheme will use up to a rate-1/3 error correction code; in general, a scheme with a depth-m code would use up to a rate-$1/m$ error correction code.

3.2 The Effect of $\mathcal{C}_1$ Upon $\mathcal{C}_0$

The process of error correction encoding after error detection encoding may have a significant effect upon the performance of the latter code. Kløve and Miller considered the effect of the error correction code on the error detection code in a scheme similar to the concatenated coding scheme described by Deng and Costello in [6]. They found that the reliability of an error detection code improves if the minimum distance of the error correction code is at least twice that of the error detection code. Unfortunately, their approach is not applicable to the problem under consideration because the inner code in this scheme has no inherent error capability by itself; in fact, its minimum distance is usually 1. Wang and Lin [20] recognized the effect of $\mathcal{C}_1$ on $\mathcal{C}_0$ but assumed that the codes in their scheme are chosen so that the performance of the error detection code is unaffected by the error correction encoding.

The encoding process may be viewed as two steps: the first consists in encoding the k information bits with the (N, k) error detection code $\mathcal{C}_0$ and the second consists in encoding this codeword with the appropriate submatrix of the error correction code $\mathcal{C}_1$. This is summarized in Figure 3.2.

$\underline{I}$	
↓	
$\underline{I}\boldsymbol{G}$	Error Detect Encode $\mathcal{C}_0$
↓	
$\underline{t} = \underline{I}\boldsymbol{G}\boldsymbol{G}_x$	"Linearly Alter" with $\boldsymbol{G}_x$ of $\mathcal{C}_1^{(x)}$
↓	
$\underline{r} = \underline{I}\boldsymbol{G}\boldsymbol{G}_x + \underline{e}$	The channel adds errors: $\underline{e}$
↓	
$\underline{r}\boldsymbol{G}_x^{-1} = \underline{I}\boldsymbol{G} + \underline{e}\boldsymbol{G}_x^{-1}$	Inversion to remove the effect of $\boldsymbol{G}_x$
↓	
$\underline{S} = \underline{r}\boldsymbol{G}_x^{-1}\boldsymbol{H}^T = \underline{I}\boldsymbol{G}\boldsymbol{H}^T + \underline{e}\boldsymbol{G}_x^{-1}\boldsymbol{H}^T$	Syndrome computation.
↓	
$\underline{S} = \underline{r}\boldsymbol{G}_x^{-1}\boldsymbol{H}^T = \underline{e}\boldsymbol{G}_x^{-1}\boldsymbol{H}^T$	Since $\boldsymbol{G}\boldsymbol{H}^T = \boldsymbol{0}$.

Figure 3.2: Syndrome Computation of a Linearly Altered Error Detection Code.

Referring to this figure, $\boldsymbol{G}$ is the $(k \times N)$ generator matrix of the error detection code and $\boldsymbol{G}_x$ is the $(N \times N)$ matrix corresponding to the x-th submatrix $\boldsymbol{M}_x$ of the error correction code. It is seen that, if $\boldsymbol{G_x}$ is invertible, the net effect of the inner code is to "linearly alter" the error detection code in such a way that the generator matrix of this code becomes $\boldsymbol{GG_x}$ and its parity check matrix becomes $\boldsymbol{G_x^{-1}H^T}$. This effect can be quite profound as the following examples will demonstrate.

Example 3.1

In this example, the error detection code is the BCH (15,11) code with generator polynomial $g(x) = x^4 + x + 1$. This code will be linearly altered by each of the following (5×5) matrices which generate the KM (15,5,5) code [1, 12].

$$\boldsymbol{M_1} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}, \boldsymbol{M_2} = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}, \boldsymbol{M_3} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

The weight distributions of the unaltered BCH (15,11) code as well as those obtained by linearly altering it with $\boldsymbol{M_1}$, $\boldsymbol{M_2}$ and $\boldsymbol{M_3}$ are shown in Table 3.1. In this table, A_k is the number of codewords with Hamming weight k in the code C_0, while $A_k^{(x)}$ is the number of codewords with Hamming weight k in the code C_0 which has been linearly altered by $\boldsymbol{M_x}$ of the error correction code.

Weight Distributions				
k	A_k	$A_k^{(1)}$	$A_k^{(2)}$	$A_k^{(3)}$
0	1	1	1	1
1	0	0	0	0
2	0	0	6	3
3	35	35	29	32
4	105	105	75	90
5	168	168	198	183
6	280	280	340	310
7	435	435	375	405
8	435	435	375	405
9	280	280	340	310
10	168	168	198	183
11	105	105	75	90
12	35	35	29	32
13	0	0	6	3
14	0	0	0	0
15	1	1	1	1

Table 3.1: Weight Distributions of the BCH (15,11) Linearly Altered Codes

It is seen from this table that the effect of the linear alteration by the matrices $\boldsymbol{M_2}$ and $\boldsymbol{M_3}$ is to decrease the minimum distance of the BCH (15,11) code from 3 to 2. Interestingly, the matrix $\boldsymbol{M}_1$ has no effect upon the weight distribution of the BCH (15,11) code. The probability of undetected error of these codes is plotted versus BSC probability of error in Figure 3.3.

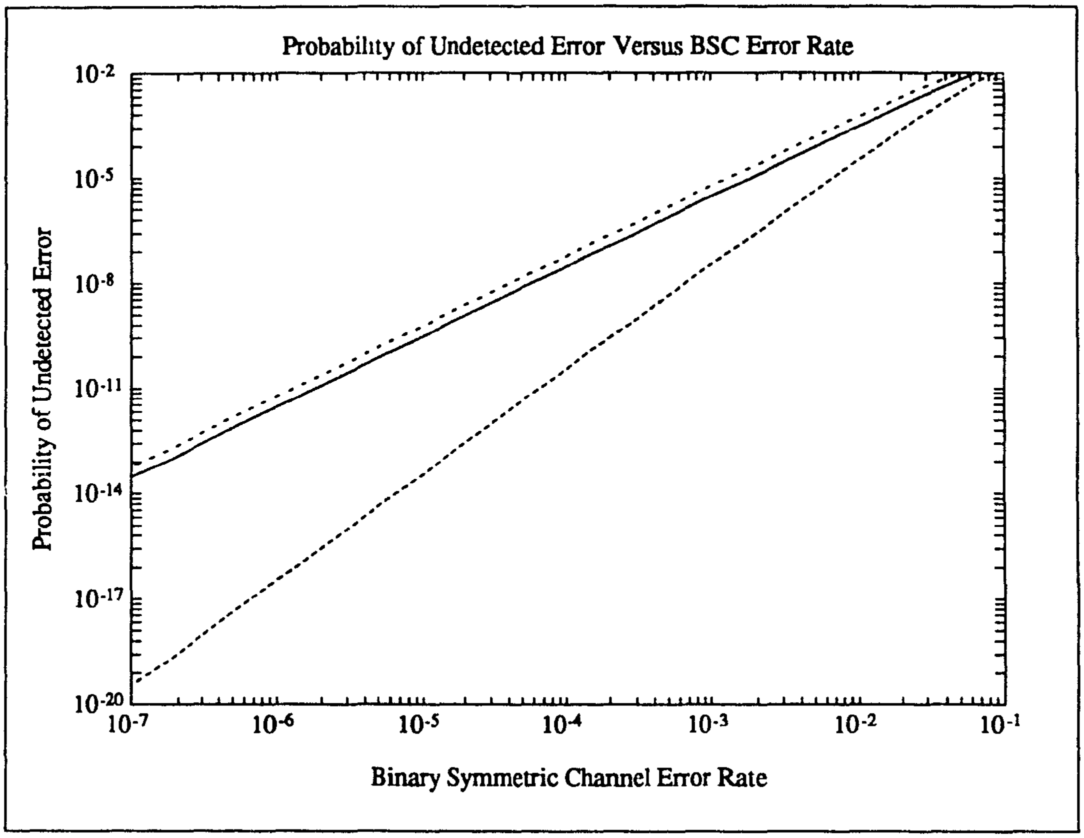

Figure 3.3: Reliability of the BCH (15,11) Code Linearly Altered by $\boldsymbol{M}_x$. Legend: – – – $\mathcal{C}_0$ and $\mathcal{C}_0^{(1)}$ – · – $\mathcal{C}_0^{(2)}$ — $\mathcal{C}_0^{(3)}$

The effect of the linear alteration is seen to be especially significant at low error rates. When the BSC error rate equals 10^{-7}, the probability of undetected error of $\mathcal{C}_0^{(2)}$ and $\mathcal{C}_0^{(3)}$ is approximately one million times worse than that of the unaltered BCH (15,11) code.

Example 3.2

The second example considers the extended Golay (24,12) code which is formed by appending a parity bit to the Golay (23,12) code so that the Hamming weight of every codeword is even. The generator polynomial of the Golay (23,12) code is $g(x) = x^{11} + x^9 + x^7 + x^6 + x^5 + x + 1$. This code is linearly altered by the following (4×4) matrices which generate the KM (12,4,5) code

$$\boldsymbol{M}_1 = \begin{pmatrix} 1&1&0&1 \\ 0&0&1&0 \\ 0&1&0&1 \\ 0&0&1&1 \end{pmatrix}, \boldsymbol{M}_2 = \begin{pmatrix} 1&1&0&0 \\ 1&1&1&0 \\ 1&0&1&0 \\ 1&1&0&1 \end{pmatrix}, \boldsymbol{M}_3 = \begin{pmatrix} 0&1&0&0 \\ 1&1&0&0 \\ 0&0&1&1 \\ 0&0&0&1 \end{pmatrix},$$

and by the following (6×6) matrices which generate the KM (18,6,6) code

$$\widehat{\boldsymbol{M}}_1 = \begin{pmatrix} 1&1&0&0&0&1 \\ 0&0&1&1&0&0 \\ 0&0&0&1&0&1 \\ 0&1&0&1&0&1 \\ 0&1&1&0&0&0 \\ 0&1&1&1&1&1 \end{pmatrix}, \widehat{\boldsymbol{M}}_2 = \begin{pmatrix} 1&0&1&1&0&1 \\ 1&0&1&0&1&1 \\ 1&1&0&1&1&0 \\ 1&1&1&0&0&1 \\ 1&1&0&0&1&1 \\ 1&0&0&1&1&0 \end{pmatrix}, \widehat{\boldsymbol{M}}_3 = \begin{pmatrix} 0&1&1&0&0&0 \\ 1&1&0&1&0&0 \\ 0&0&1&0&0&0 \\ 0&0&0&1&0&0 \\ 0&0&1&0&1&1 \\ 0&0&0&1&0&1 \end{pmatrix}.$$

The Hamming weight distributions of the unaltered Extended Golay (24,12) code as well as those obtained by linearly altering it are shown in Table 3.2. In this table, A_k is the number of codewords with Hamming weight k in the extended Golay (24,12) code, $A_k^{(x)}$ and $\hat{A}_k^{(x)}$ are the number of codewords with Hamming weight k in the Golay (24,12) code which is linearly altered by $\boldsymbol{M}_x$ and $\widehat{\boldsymbol{M}}_x$ respectively.

Weight Distributions							
		KM (12,4,5)			KM (18,6,6)		
k	A_k	$A_k^{(1)}$	$A_k^{(2)}$	$A_k^{(3)}$	$\hat{A}_k^{(1)}$	$\hat{A}_k^{(2)}$	$\hat{A}_k^{(3)}$
0	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	2	0	3	2	2
5	0	5	13	6	13	11	7
6	0	37	31	26	35	35	34
7	0	89	82	101	78	77	97
8	759	179	172	201	149	177	169
9	0	333	316	310	325	336	311
10	0	469	485	454	521	463	476
11	0	601	614	589	614	595	601
12	2576	665	688	691	659	679	697
13	0	599	610	634	599	624	621
14	0	487	441	446	461	489	450
15	0	323	318	319	322	307	315
16	759	176	191	194	178	162	182
17	0	87	84	74	87	84	85
18	0	31	35	34	39	37	32
19	0	11	10	15	10	13	11
20	0	3	2	1	2	3	5
21	0	0	1	0	0	1	0
22	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0
24	1	0	0	0	0	0	0

Table 3.2: Weight Distributions of the Extended Golay (24,12) Linearly Altered Codes

It is seen from this table that the effect of the matrices $\boldsymbol{M}_1$ and $\boldsymbol{M}_3$ is to decrease the minimum distance of the Extended Golay (24,12) code from 8 to 5 and that of $\boldsymbol{M}_2$ decreases its minimum distance from 8 to 4. Further, $\widehat{\boldsymbol{M}}_1$, $\widehat{\boldsymbol{M}}_2$ and $\widehat{\boldsymbol{M}}_3$ decrease the minimum distance of the Extended Golay (24,12) from 8 to 4. The probability of undetected error is plotted versus BSC probability of error in Figures 3.4 and 3.5. Further, $\boldsymbol{M}_x$ appears to alter the performance of the error detection code less than $\widehat{\boldsymbol{M}}_x$. This suggests that, when selecting an error correction code to be used in such an error control scheme, the error correction capability of the code $\mathcal{C}_1$ should not be the sole criterion. Its effect upon the performance of the error detection code also needs to be considered.

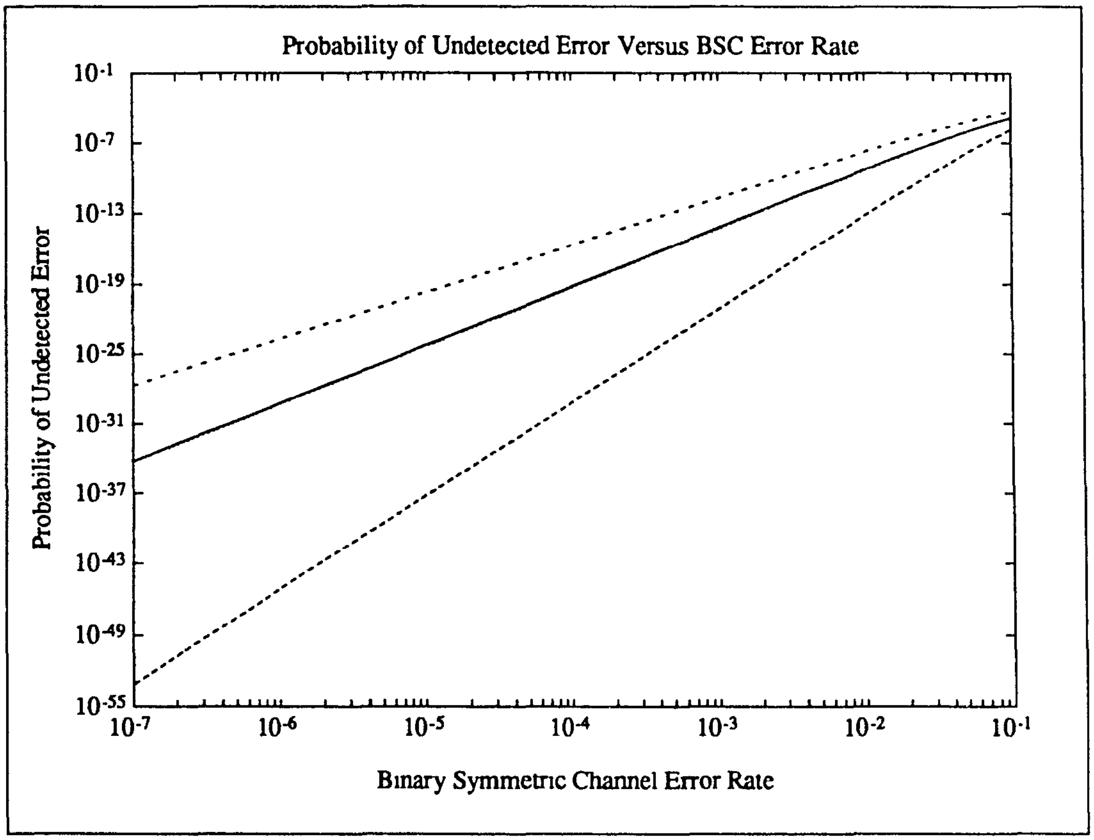

Figure 3.4: Reliability of the Extended Golay (24,12) Code Linearly Altered by $\boldsymbol{M}_x$. Legend: — — — $\mathcal{C}_0$ — $\mathcal{C}_0^{(1)}$ and $\mathcal{C}_0^{(3)}$ — · — $\mathcal{C}_0^{(2)}$

Again, the effect of the linear alteration is more significant at lower error rates; yet it remains more significant at high rates than it was in the previous case of the BCH (15,11). This suggests that some "matching" between the error detection and correction codes is required so as to minimize the effect of the latter upon the former while maintaining an acceptable error correction capability.

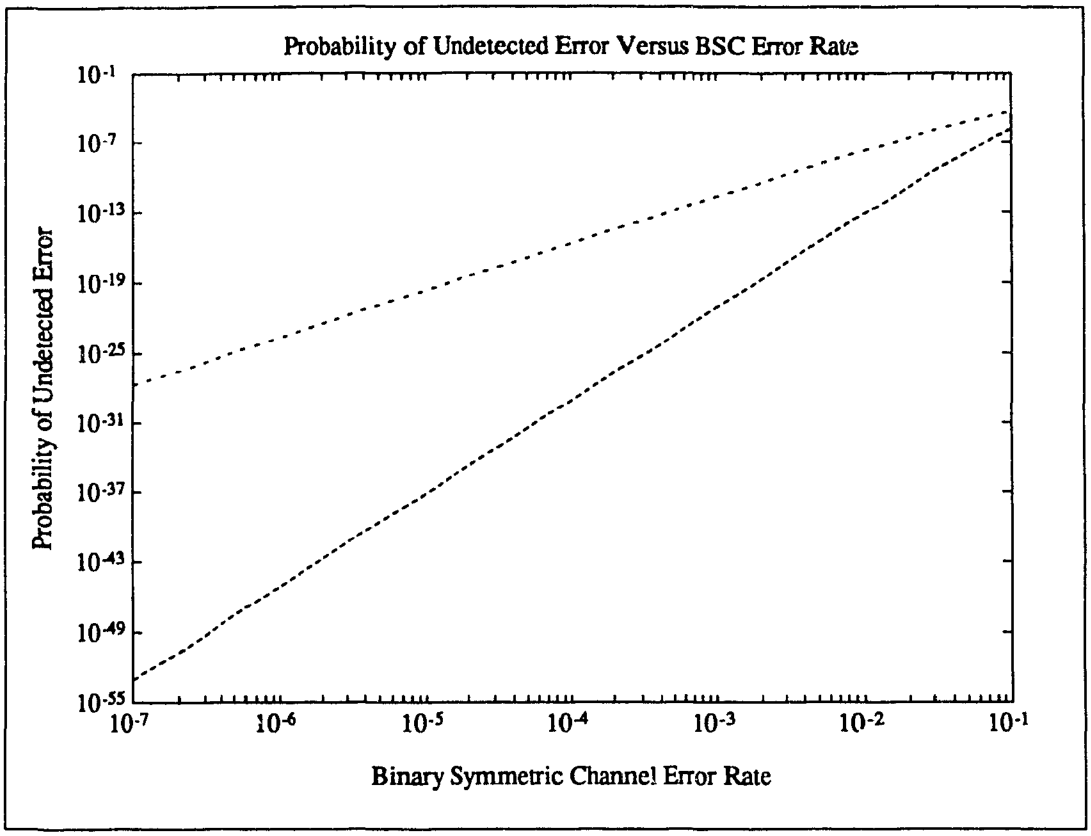

Figure 3.5: Reliability of the Extended Golay (24,12) Code Linearly Altered by $\widehat{\boldsymbol{M}_x}$. Legend: $---$ $\mathcal{C}_0$ $\quad - \cdot -$ $\hat{\mathcal{C}}_0^{(1)}$, $\hat{\mathcal{C}}_0^{(2)}$, and $\hat{\mathcal{C}}_0^{(3)}$

3.3 Computation of Weight Distributions

In order to analyze the performance of the GH-ARQ II error control scheme, it will be necessary to compute the weight distribution of the error detection and error correction codes. This does not pose a problem when k is small, but restricts the analysis to very low rate codes which are unpractical. Suppose it is required to compute the weight distribution of codes for which k is quite large (on the order of 500 bits for example). In this case, the MacWilliams Identity will prove to be very useful. This section will consider the cases of an unaltered code, a linearly altered code, and a shortened code. It is assumed that all codes under consideration are binary and linear.

3.3.1 Unaltered Code $\mathcal{C}$

For the case where no error correction coding has been performed, the MacWilliams identity will directly apply. Assume that it is desired to compute the weight distribution of the binary (N,k) linear code $\mathcal{C}$ with generator matrix $\boldsymbol{G}$. Further, suppose that the rate of the code, $R=\frac{k}{N}$, is close to unity and because k is large, it is not possible to directly compute the weight distribution of this code. The dual code $\mathcal{C}^{\perp}$ is generated by the parity check matrix $\boldsymbol{H}$ of the code $\mathcal{C}$. Note that $\boldsymbol{G}\boldsymbol{H}^T=\boldsymbol{0}$, where $\boldsymbol{0}$ is the $(k\times(N-k))$ zero matrix. The dual code is a very low rate code consisting of 2^{N-k} codewords. Since $N-k$ is small, the weight distribution of the dual code is easily computed by enumeration of all its codewords. Let the weight distribution of the dual code $\mathcal{C}^{\perp}$ be denoted by $\{A'_k\}$ and that of the code $\mathcal{C}$ by $\{A_k\}$. The MacWilliams Identity relates the two weight distributions as follows [8]:

$$\sum_{k=0}^{N} A_k x^{N-k} y^k = \frac{1}{\mid \mathcal{C}^{\perp} \mid} \sum_{k=0}^{N} A'_k (x+y)^{N-k}(x-y)^k,$$

where $\mid \mathcal{C}^{\perp} \mid = 2^{N-k}$, which is the number of codewords in the dual code. Equivalently, it is possible to set $x=1$ to obtain:

$$\sum_{k=0}^{N} A_k y^k = \frac{1}{\mid \mathcal{C}^{\perp} \mid} \sum_{k=0}^{N} A'_k (y+1)^{N-k}(1-y)^k.$$

Therefore, after computing the weight distribution of the low rate dual code $\mathcal{C}^{\perp}$, it is then a simple matter to compute the weight distribution of the high rate code $\mathcal{C}$.

3.3.2 Linearly Altered Code $\mathcal{C}_0^{(x)}$

Now consider the case of a long, high rate linearly altered code, say $\mathcal{C}_0^{(x)}$. Suppose that the error detection code is linearly altered by $\boldsymbol{M}_x$, which leads to the generator matrix $\boldsymbol{G}_x = \boldsymbol{M}_x \otimes \boldsymbol{I}_{\frac{N}{n}}$. The problem then is to compute the dual code $\mathcal{C}_0^{(x)\perp}$, given that $\mathcal{C}_0^{(x)}$ is generated by $\boldsymbol{G}\boldsymbol{G}_x$. Assuming that $\boldsymbol{G}_x^{-1}$ exists, then the dual of $\mathcal{C}_0^{(x)}$ is generated by $\boldsymbol{H}(\boldsymbol{G}_x^{-1})^T$. It may be verified that the condition for $\mathcal{C}_0^{(x)}$ and $\mathcal{C}_0^{(x)\perp}$ to be dual,

$$\boldsymbol{G}\boldsymbol{G}_x(\boldsymbol{H}(\boldsymbol{G}_x^{-1})^T)^T = \boldsymbol{0},$$

is indeed true. Therefore, to compute the weight distribution of the linearly altered error detection code $\mathcal{C}_0^{(x)}$ generated by $\boldsymbol{G}\boldsymbol{G}_x$, it suffices to compute the weight distribution of the dual code $\mathcal{C}_0^{(x)\perp}$ generated by $\boldsymbol{H}(\boldsymbol{G}_x^{-1})^T$ and then to use the MacWilliams Identity.

3.3.3 Shortened Code $\mathcal{C}_s$

Often a suitable code cannot be found to match the desired length of a packet. In this case, a code $\mathcal{C}$ with natural length longer than the desired packet length may be shortened to accomodate the constraints. Let the natural length of the code be N bits and the desired length of the packet be $N-\Delta$ bits, so that it is required to shorten the code $\mathcal{C}$ by Δ bits. Also, let the shortened code be denoted by $\mathcal{C}_s$. The generator matrix of the code $\mathcal{C}$ is $\boldsymbol{G}$, a $(k \times N)$ matrix. Similarly, the generator matrix of the shortened code $\mathcal{C}_s$ is $\boldsymbol{G}_s$, a $((k-\Delta)\times(N-\Delta))$ matrix. The dual matrix of $\boldsymbol{G}_s$, denoted by $\boldsymbol{H}_s$, is a $((N-k)\times(N-\Delta))$ matrix. To compute the weight distribution of the code $\mathcal{C}_s$, it suffcies to compute the weight distribution of the code generated by $\boldsymbol{H}_s$ and then to use the MacWilliams Identity.

$$\sum_{k=0}^{N} A_k y^k = \frac{1}{2^{N-k}} \sum_{k=0}^{N} A'_k (y+1)^{N-k}(1-y)^k,$$

where $\{A_k\}$ and $\{A'_k\}$ are the weight distributions of the codes $\mathcal{C}_s$ and $\mathcal{C}_s^{\perp}$ respectively.

Example 3.3

Suppose it is desired to compute the weight distribution of the BCH (31,26) code which is shortened by 1 ($\Delta = 1$) bit and linearly altered by each of the three (5×5) matrices in Example 3.1. The generator polynomial of the BCH (31,26) code is $g(x) = x^5 + x^2 + 1$. Direct computation of the weight distribution of this code would require the generation of 2^{25} codewords; clearly, this is not a practical approach. This is an ideal application for the MacWilliams Identity since $N-k$ is small and the dual code is a very low rate code. The generator polynomial of the dual code, $h(x)$, is [8]

$$\begin{aligned} h(x) &= \frac{x^N - 1}{g(x)} = \frac{x^{31}-1}{x^5+x^2+1} \\ &= x^{26}+x^{23}+x^{21}+x^{20}+x^{17}+x^{16}+x^{15}+x^{14} \\ &\quad +x^{13}+x^9+x^8+x^6+x^5+x^4+x^2+1 \end{aligned}$$

Note that the shortened (30,25) code is not necessarily cyclic. To avoid confusion regarding the various codes that are generated by the BCH (31,26) code $\mathcal{C}$, the shortened (30,25) code will be denoted by $\mathcal{C}_s$ and the shortened (30,25) code which is linearly altered by $\boldsymbol{M}_x$ of the KM (15,5,5) code will be denoted by $\mathcal{C}_s^{(x)}$. Then, using the MacWilliams Identity, the weight distribution of these shortened codes may be easily computed. They are summarized in Table 3.3. In this table, A_k refers to the number of codewords in $\mathcal{C}_s$ with weight k, while $A_k^{(x)}$ refers to the number of codewords in $\mathcal{C}_s^{(x)}$ with Hamming weight k.

Weight Distributions				
k	A_k	$A_k^{(1)}$	$A_k^{(2)}$	$A_k^{(3)}$
0	1	1	1	1
1	0	0	0	0
2	0	13	9	12
3	140	121	136	111
4	945	839	859	875
5	4368	4520	4413	4577
6	18200	18606	18582	18356
7	63960	63414	63728	63334
8	183885	182857	182817	183661
9	446160	447376	446886	447178
10	936936	938975	939091	937544
11	1708980	1706931	1707440	1707857
12	2705885	2702543	2702507	2703959
13	3739680	3742704	3742023	3741135
14	4541040	4545396	4545252	4544808
15	4850640	4846692	4848000	4848084
16	4547475	4543227	4543371	4543011
17	3739680	3743808	3741924	3743244
18	2700880	2703835	2703871	2704156
19	1708980	1705839	1707528	1705761
20	939939	938525	938409	938481
21	446160	447784	446875	447983
22	182520	182974	183014	182884
23	63960	63430	63696	63334
24	18655	18555	18579	18607
25	4368	4464	4438	4490
26	840	857	837	848
27	140	133	128	127
28	15	13	17	13
29	0	0	1	1
30	0	0	0	0

Table 3.3: Weight Distributions of the Shortened BCH (30,25) Codes

Chapter 4

Analysis of the GH-ARQ II Scheme

This chapter presents the performance analysis of a GH-ARQ II error control scheme on a burst-noise channel described by the Gilbert-Elliott model. In order to efficiently analyze a system that may use a depth-2, depth-3, or even depth-4 error correction code, a systematic approach is required. This approach begins by describing the receiver's state transition diagram which is a simple first-order Markov chain. From this Markov chain, it is possible to compute expressions for the system's throughput and reliability in terms of its transition probabilities. The problem then becomes the computation of these transition probabilities.

This approach will be seen to be general in the sense that it is applicable to the analysis of depth-2, depth-3, and depth-4 systems and is easily extended to other channels modelled by first-order Markov chains.

4.1 Analysis Assumptions

This section briefly describes the pertinent assumptions that will be made in the analysis of the GH-ARQ II error control scheme. The feedback channel is assumed to be *noiseless*. This is a common assumption in the literature and is achievable in practice by coding the receiver's message with a very low rate code.

It is also assumed that the error correction code $\mathcal{C}_1$ has a non-negligible effect on the performance of the error detection code $\mathcal{C}_0$. Wang and Lin [20] had assumed in their analysis that $\mathcal{C}_1$ was chosen so that it had no effect on $\mathcal{C}_0$.

The transitions in the Gilbert-Elliott model are assumed to occur between transmissions of the packets. This ensures that during the transmission of a packet, the error rate is constant. This assumption is common and was made, for example, by Lugand *et al.* [4] and Deng *et al.* [6] in their analyses.

The retransmission strategy is a selective repeat request approach with an infinite receiver buffer; the details regarding its operation are not considered. Various selective repeat request strategies are considered in detail in [2, 45, 50]. Also, the stop-and-wait retransmission strategy will be treated as a special case of the analysis with a selective repeat request retransmission strategy.

Since a depth-2 system is analyzed, the generator matrix of the error correction code $\mathcal{C}_1^{(2)}$ will be $\boldsymbol{M} = [\boldsymbol{M}_1 \mid \boldsymbol{M}_2]$, and it will be assumed that both $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ are invertible. As described in Appendix A.2, the analysis may be easily modified to account for non-invertible $\boldsymbol{M}_i$.

Finally, the error detection and error correction decoders are not assumed to be *bounded distance* decoders. In the case of the error detection decoder, this means that it is capable of detecting some error patterns with Hamming weight greater than or equal to the minimum distance of the error detection code $\mathcal{C}_0$. In the case of the error correction decoder, this means that it is capable of correcting some error patterns with Hamming weight greater than its error correction capability t_x, where $t_x = \lfloor \frac{d_x - 1}{2} \rfloor$ and d_x is the minimum distance of the code $\mathcal{C}_1^{(x)}$. For convenience let τ_x be the maximum number of errors in a combined block $[\mathcal{B}_j^{(i+1-x)} \mid \ldots \mid \mathcal{B}_j^{(i)}]$ which is possibly correctable. Consequently, this implies that there does not exist any error sequence $\underline{e}$ containing more than τ_x which is correctable. In general, $\tau_x > t_x$ unless $\mathcal{C}_1^{(x)}$ is a *perfect code.* Recall that a perfect code is one which is capable of correcting all sequences with up to and including t_x errors, but cannot correct any sequences with more than t_x errors; therefore, for such a code, $\tau_x = t_x$.

4.2 Receiver State Transition Diagram

This section describes the receiver state transition diagram for a depth-2 system which, in effect, uses a half-rate code for error correction. This state transition diagram is shown in Figure 4.1.

As seen in this figure, the depth-2 receiver state transition diagram consists of five states: two absorbing states, **c** and **u**, and three transient states, **1**, **T1**, and **T2**. The state **c** corresponds to the receiver delivering an error-free message to the data sink, while the state **u** corresponds to it delivering a message containing errors. In its initial state, state **1**, the receivers inverts the received packet (multiplying by $\boldsymbol{G}_x^{-1}$) and attempts to detect errors. After this inversion, the receiver will deliver

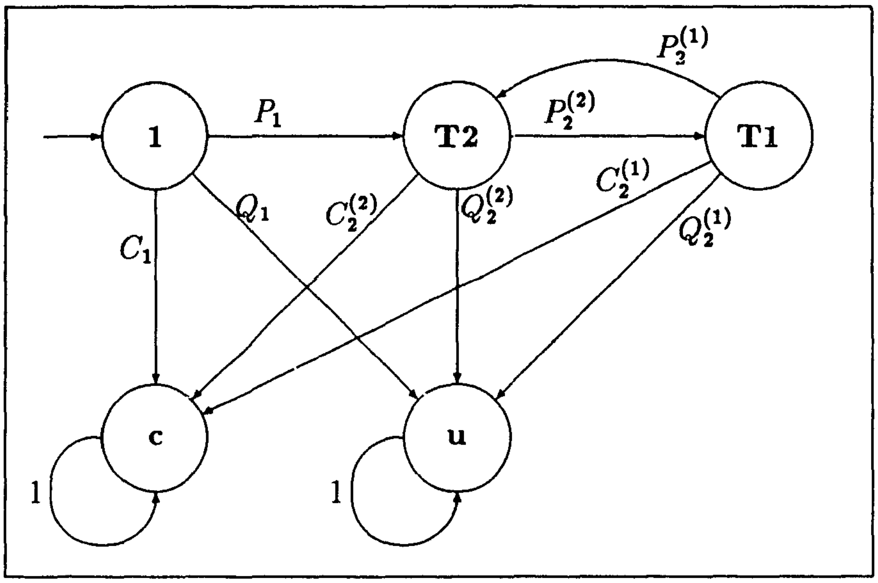

Figure 4.1: Depth-2 System Receiver State Transition Diagram

an accurate message to the data sink with probabilty C_1, and will deliver a message containing undetected errors with probability Q_1. The probability that the receiver will request a retransmission, after this initial inversion, is P_1.

After the receiver has requested a retransmission, it enters state **T2** in which the transmitter will encode the packet $(\underline{I}, \underline{Q})$ with $\boldsymbol{M_2}$ of the error correction code $C_1^{(2)}$ as described in the previous section. Similar to state **1**, the receiver delivers an error-free message (by successfully inverting or decoding the received packet) to the sink with probability $C_2^{(2)}$ and delivers a message containing undetected errors with probability $Q_2^{(2)}$. It will request another retransmission with probability $P_2^{(2)}$ and then moves to state **T1**. This state is identical to state **T2** except that the transmitter codes the packet $(\underline{I}, \underline{Q})$ with $\boldsymbol{M_1}$ of the error correction code $C_1^{(2)}$. In this case, the transition probabilities are superscripted by (1) instead of (2). Note that the receiver never returns to state **1**. This reflects the fact that after the initial unsuccessful transmission of a message, the receiver will always attempt to correct the errors in a received packet if the inversion is not possible or unsuccessful. Thus, the receiver moves between states **T2** and **T1** as the transmitter alternates encoding the message with $\boldsymbol{M_2}$ and $\boldsymbol{M_1}$ of the error correction code $C_1^{(2)}$. Finally, the receiver returns `NAK` to the transmitter when it moves from one transient state to another and returns `ACK` when it moves from a transient state to an absorbing state.

4.3 Throughput and Reliability

In this section, the throughput and reliability of the the GH-ARQ II (Depth-2) scheme are expressed in terms of the transition probabilities in the receiver's state transition diagram. These may be easily computed by using the *first-step analysis* described in [19]. In the present context, the throughput is defined to be the mean number of transmissions of a message before the receiver returns a positive acknowledgement to the transmitter. This is the expected number of transitions in the receiver state transition diagram before absorption into either states **c** or **u**. The reliability (or probability of undetected error) is defined to be the probability of absorption into state **u**. To compute the throughput and reliability of this system, the transition probability matrix $\boldsymbol{P}$ of the receiver must be constructed. That is,

$$\begin{array}{c} \\ \boldsymbol{P} = \end{array} \begin{array}{c} \begin{array}{ccccc} \mathbf{c} & \mathbf{1} & \mathbf{T1} & \mathbf{T2} & \mathbf{u} \end{array} \\ \left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ C_1 & 0 & 0 & P_1 & Q_1 \\ C_2^{(1)} & 0 & 0 & P_2^{(1)} & Q_2^{(1)} \\ C_2^{(2)} & 0 & 0 & P_2^{(2)} & Q_2^{(2)} \\ 0 & 0 & 0 & 0 & 1 \end{array}\right] \end{array} \begin{array}{c} \mathbf{c} \\ \mathbf{1} \\ \mathbf{T1} \\ \mathbf{T2} \\ \mathbf{u} \end{array}$$

This matrix may be more conveniently expressed by relabelling the states and the transition probabilities from states **1**, **T1**, and **T2** so that they may be more easily indexed. Note that P_{ij} is the probability of transition from state i to state j.

$$\begin{array}{c} \\ \boldsymbol{P} = \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ P_{10} & P_{11} & P_{12} & P_{13} & P_{14} \\ P_{20} & P_{21} & P_{22} & P_{23} & P_{24} \\ P_{30} & P_{31} & P_{32} & P_{33} & P_{34} \\ 0 & 0 & 0 & 0 & 1 \end{array}\right] \end{array} \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array}$$

Let X_t be the state of the Markov chain at time t and assume that T is the number of steps until absorption so that $T = \min_{n \geq 0}\{X_n = 0 \text{ or } X_n = 4\}$.

The reliability of the GH-ARQ II scheme P_{ud} is $\Pr(X_T = 4 \mid X_0 = 1)$, the probability of absorption into state 4 (**u**)To compute this, begin by defining u_i as follows:

$$u_i = \Pr(X_T = 4 \mid X_0 = i), \quad i = 1, 2, 3.$$

The key to the first-step analysis is to compute $\Pr(X_T = 4 \mid X_1 = i), i = 1, \ldots, 5$. First consider $X_1 = 0$; in this case $\Pr(X_T = 4 \mid X_1 = i) = 0$ because $X_1 = 0$ corresponds to the absorbing state **c**. Conversely, if $X_1 = 4$, then $\Pr(X_T = 4 \mid X_1 =$

$i) = 1$. Finally, if $X_1 = i$ for $i = 1, 2, 3$, then $\Pr(X_T = 4 \mid X_1 = i) = u_i$ because the transition probabilties of the chain are assumed stationary. In summary, then,

$$\begin{aligned}
\Pr(X_T = 4 \mid X_1 = 0) &= 0 \\
\Pr(X_T = 4 \mid X_1 = 1) &= u_1 \\
\Pr(X_T = 4 \mid X_1 = 2) &= u_2 \\
\Pr(X_T = 4 \mid X_1 = 3) &= u_3 \\
\Pr(X_T = 4 \mid X_1 = 4) &= 1.
\end{aligned}$$

Recall that $u_i = \Pr(X_T = 4 \mid X_0 = i)$ for $i = 1, 2, 3$. By total probability, then

$$\begin{aligned}
u_i &= \sum_{k=0}^{4} \Pr(X_T = 4, X_1 = k \mid X_0 = i) \\
&= \sum_{k=0}^{4} \Pr(X_T = 4, \mid X_1 = k, X_0 = i) \Pr(X_1 = k \mid X_0 = i) \\
&= \sum_{k=0}^{4} \Pr(X_T = 4, \mid X_1 = k) \Pr(X_1 = k \mid X_0 = i)
\end{aligned}$$

The final step is a consequence of the fact that the Markov chain is first-order so that the next state depends only on the present state and none of the previous ones. Expand the above equations for $i = 1, 2, 3$ to obtain

$$\begin{aligned}
u_i &= \Pr(X_T = 4, \mid X_1 = 0) \Pr(X_1 = 0 \mid X_0 = i) \\
&\quad + \Pr(X_T = 4, \mid X_1 = 1) \Pr(X_1 = 1 \mid X_0 = i) \\
&\quad + \Pr(X_T = 4, \mid X_1 = 2) \Pr(X_1 = 2 \mid X_0 = i) \\
&\quad + \Pr(X_T = 4, \mid X_1 = 3) \Pr(X_1 = 3 \mid X_0 = i) \\
&\quad + \Pr(X_T = 4, \mid X_1 = 4) \Pr(X_1 = 4 \mid X_0 = i) \\
&= u_1 P_{i1} + u_2 P_{i2} + u_3 P_{i3} + P_{i4}
\end{aligned}$$

These equations may be expressed as the following system:

$$\begin{bmatrix} P_{11} - 1 & P_{12} & P_{13} \\ P_{21} & P_{22} - 1 & P_{23} \\ P_{31} & P_{32} & P_{33} - 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} -P_{14} \\ -P_{24} \\ -P_{34} \end{bmatrix} \tag{4.1}$$

Substitute the values of the transition probabilities in the receiver's state transition diagram of Figure 4.1 for the P_{ij} in Equation 4.1 to obtain:

$$\begin{bmatrix} -1 & 0 & P_1 \\ 0 & -1 & P_2^{(1)} \\ 0 & P_2^{(2)} & -1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} -Q_1 \\ -Q_2^{(1)} \\ -Q_2^{(2)} \end{bmatrix}$$

Finally, solve for $P_{ud} = u_1$ (by using Cramer's rule, for example) to obtain the expression for the reliability of the GH-ARQ II scheme as a function of the transition probabilities of the receiver state transition diagram.

$$P_{ud} = Q_1 + P_1 \left[\frac{Q_2^{(2)} + P_2^{(2)} Q_2^{(1)}}{1 - P_2^{(1)} P_2^{(2)}} \right]$$

The next problem is to compute the throughput of the GH-ARQ II system. Let T be defined as previously and $v_i = \mathrm{E}[T \mid X_0 = i]$, where $\mathrm{E}[\cdot]$ denotes the expectation operation. The quantity v_i is interpreted to be the average number of transitions until absorption into state **c** or state **u** given that the initial state is i, so that the throughput of the GH-ARQ II system is simply v_1. The approach to solve for v_1 is similar to that previously taken to solve for u_1. To compute v_i, use the fact that the Markov chain is first-order and total probability to obtain:

$$\begin{aligned} v_i &= \mathrm{E}[T \mid X_0 = i] \\ &= \sum_{k=0}^{4} \mathrm{E}[T \mid X_1 = k] \Pr(X_1 = k \mid X_0 = i) \end{aligned}$$

Now, the quantity $\mathrm{E}[T \mid X_1 = k]$ must be computed. If $k = 0$ or $k = 4$, then the process has reached an absorbing state after one transition, so $\mathrm{E}[T \mid X_1 = 0] = \mathrm{E}[T \mid X_1 = 4] = 1$. If $1 \le i \le 3$, then $\mathrm{E}[T \mid X_1 = k] = 1 + v_i$. Therefore, for $1 \le i \le 3$,

$$v_i = 1 + v_1 P_{i1} + v_2 P_{i2} + v_3 P_{i3}.$$

This equation may then be expanded into a system of three linear equations:

$$\begin{bmatrix} P_{11} - 1 & P_{12} & P_{13} \\ P_{21} & P_{22} - 1 & P_{23} \\ P_{31} & P_{32} & P_{33} - 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix} \tag{4.2}$$

Again, substitute the values of the transition probabilities in the receiver's state transition diagram of Figure 4.1 for the P_{ij} in Equation 4.2 to obtain:

$$\begin{bmatrix} -1 & 0 & P_1 \\ 0 & -1 & P_2^{(1)} \\ 0 & P_2^{(2)} & -1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$

Solving for v_1 the throughput of the GH-ARQ II system η is

$$\eta = \frac{1}{1 + P_1 \left[\frac{1 + P_2^{(2)}}{1 - P_2^{(1)} P_2^{(2)}} \right]}$$

4.4 Computation of P_1, Q_1, and C_1

The computation of the transition probabilities P_1, Q_1, and C_1 is very similar to the computation of the probabilities P, Q, and C of the simple ARQ error control scheme described in Section 2.2. The effect of the error correction code on the error detection code is easily integrated into these expressions.

The probability of correct reception on the first transmission, C_1, is simply the probability that zero errors occurred during transmission. Let Ω_i be the current state of the channel. Then, as in Section 2.2

$$\begin{aligned} C_1 &= \Pr(0 \text{ Transmission Errors}) \\ &= \Pr(0 \text{ Transmission Errors} \mid \Omega_i = \text{G}) \Pr(\Omega_i = \text{G}) \\ &\quad + \Pr(0 \text{ Transmission Errors} \mid \Omega_i = \text{B}) \Pr(\Omega_i = \text{B}) \end{aligned}$$

$$\boxed{C_1 = \frac{1}{P_{gb} + P_{bg}} \left[P_{bg}(1-\epsilon_g)^N + P_{gb}(1-\epsilon_b)^N \right]}$$

The probability of undetected error on the first transmission, Q_1, is the probability that the error sequence $\underline{e}$ is a codeword in the error detection code that is linearly altered by $\boldsymbol{M}_i$ of the error correction code.

$$Q_1 = \Pr(\text{Undetected Errors}, \Omega_i = \text{G}) + \Pr(\text{Undetected Errors}, \Omega_i = \text{B})$$

$$\boxed{Q_1 = \frac{1}{P_{gb} + P_{bg}} \sum_{k=d_1}^{N} A_k^{(1)} \left[P_{bg}\epsilon_g^k (1-\epsilon_g)^{N-k} + P_{gb}\epsilon_b^k (1-\epsilon_b)^{N-k} \right]}$$

where $A_k^{(1)}$ and d_1 are the number of codewords with Hamming weight k and minimum distance in the error detection code that has been linearly altered by $\boldsymbol{M}_i$ of the error correction code. The computation of the weight distribution $\{A_k^{(1)}\}$ is described in Section 3.3.

Finally, the probability of a retransmission after the first transmission, P_1, is simply

$$\boxed{P_1 = 1 - Q_1 - C_1}$$

4.5 Computation of $P_2^{(x)}$, $Q_2^{(x)}$, and $C_2^{(x)}$

In order to compute the transition probabilities $P_2^{(x)}$, $Q_2^{(x)}$, and $C_2^{(x)}$, the flowchart shown in Figure 4.2 will be useful. Note that x refers to the submatrix $\boldsymbol{M}_x$ of the error correction code $C_1^{(2)}$ with which the currently transmitted packet is encoded.

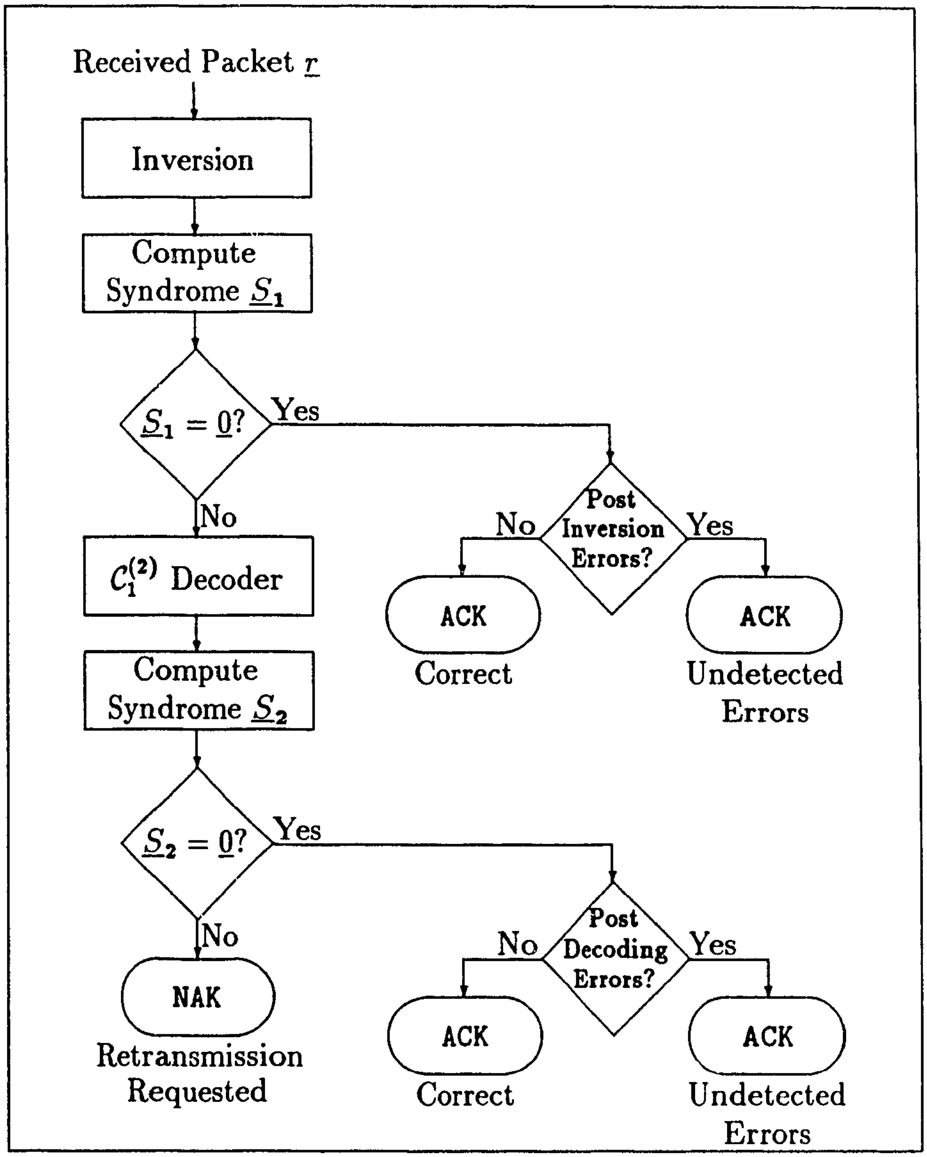

Figure 4.2: Depth-2 System Receiver Flowchart

In this flowchart, the `Inversion` is performed if the matrix $\boldsymbol{M}_x$ that was used to code this packet is invertible. If it is not, then the receiver begins immediately with the `Decoder`.

From Figure 4.2, the transition probabilities $C_2^{(x)}$, $Q_2^{(x)}$, and $P_2^{(x)}$ may be expressed as

$$\begin{aligned} C_2^{(x)} &= \Pr(\underline{S}_1 = \underline{0}, \text{No PI Errors}) + \Pr(\underline{S}_1 \neq \underline{0}, \underline{S}_2 = \underline{0}, \text{No PD Errors}) \\ Q_2^{(x)} &= \Pr(\underline{S}_1 = \underline{0}, \text{PI Errors}) + \Pr(\underline{S}_1 \neq \underline{0}, \underline{S}_2 = \underline{0}, \text{PD Errors}) \\ P_2^{(x)} &= \Pr(\underline{S}_1 \neq \underline{0}, \underline{S}_2 \neq \underline{0}) \end{aligned}$$

where "PI Errors" refers to the presence of post-inversion errors (if the inversion is possible) and "PD Errors" refers to the presence of post-decoding errors at the output of the decoder.

4.5.1 Computation of $C_2^{(x)}$

Recall that $C_2^{(x)} = \Pr(\underline{S}_1 = \underline{0}, \text{No PI Errors}) + \Pr(\underline{S}_1 \neq \underline{0}, \underline{S}_2 = \underline{0}, \text{No PD Errors})$. This is interpreted as the probability that the receiver delivers an error-free message to the data sink on the second and succeeding transmissions.

The $\Pr(\underline{S}_1 = \underline{0}, \text{No PI Errors})$ is simply the probability of zero transmission errors which is equivalent to the transition probability C_1 computed in Section 4.4; therefore,

$$\Pr(\underline{S}_1 = \underline{0}, \text{No PI Errors}) = \frac{1}{P_{gb} + P_{bg}} \left[P_{bg}(1 - \epsilon_g)^N + P_{gb}(1 - \epsilon_b)^N \right].$$

The $\Pr(\underline{S}_1 \neq \underline{0}, \underline{S}_2 = \underline{0}, \text{No PD Errors})$ may be interpreted as the probability that the error sequence added to the packet during transmission is not a codeword in the error detection code $\mathcal{C}_0^{(x)}$ (linearly altered by $\boldsymbol{M}_x$) and that this sequence is correctable by the decoder. In this context, "correctable" means that the error sequence is decoded to $\underline{0}$ so that the decoded packet is error-free. Let

$$X = \Pr(\underline{S}_1 \neq \underline{0}, \underline{S}_2 = \underline{0}, \text{No PD Errors}).$$

There are two approaches which may be taken to compute this probability. The first approach begins by generating every possible error sequence which is not a codeword in the code $\mathcal{C}_0^{(x)}$. Then, X is computed by summing the probability of successfully correcting each of these error sequences. When the length of the packet N is small (say $N < 15$) this exhaustive approach is feasible. However, when the packet length is greater than this, a more efficient approach is required.

To develop such an approach, X may be equivalently expressed, as

$$X = \sum_{j=1}^{2^N-2^k} \Pr(\underline{\epsilon}_j \notin C_0^{(x)}, \underline{\epsilon}_{pd}^{(j)} = \underline{0}),$$

where the expression $\underline{\epsilon}_j \notin C_0^{(x)}$ is interpreted to be an N-bit error sequence $\underline{\epsilon}_j$ which is not a codeword in the code $C_0^{(x)}$, and $\underline{\epsilon}_{pd}^{(j)}$ is the decoded error sequence at the output of the error correction code decoder when the currently received error sequence is $\underline{\epsilon}_j$. Note that there are 2^N-2^k words are not codewords. The second approach to compute X decreases the number of error sequences which need to be considered by grouping together those sequences $\underline{\epsilon}_j$ with "similar structure". Moreover, the requirement that $\underline{\epsilon}_j$ be correctable, $\underline{\epsilon}_{pd}^{(j)} = \underline{0}$, further decreases the number of sequences that needs to be considered.

The notion of correctable error sequences with similar structure is the key to making the computation of X manageable. Suppose that the received packet $\underline{r}$ is comprised of four blocks so that

$$\underline{r} = [\mathcal{B}_0 \mid \mathcal{B}_1 \mid \mathcal{B}_2 \mid \mathcal{B}_3].$$

Now suppose that the Hamming weight of blocks $\mathcal{B}_1$, $\mathcal{B}_2$, and $\mathcal{B}_3$ is zero, and that of block $\mathcal{B}_0$ is two. Let the Hamming weight of this packet $w(\underline{r})$ be denoted as

$$w(\underline{r}) = [2 \mid 0 \mid 0 \mid 0].$$

Two packets are said to be *equivalent* if the weight partition[1] of one of the packets may be obtained from the other by permuting the order of its blocks. For example, packets $\underline{r}_1$ and $\underline{r}_2$ with weight partitions $w(\underline{r}_1) = [2 \mid 1 \mid 0 \mid 1]$ and $w(\underline{r}_2) = [0 \mid 1 \mid 1 \mid 2]$ are equivalent in the sense that their weight partitions differ only by their order. Further, a received packet $\underline{r}$ is *potentially correctable* if the Hamming weight of each of its blocks is less than or equal to τ_2, where τ_2 is as defined in Section 4.1. Whether or not this packet is actually correctable depends also upon the previously received packet. Conversely, if the currently received packet contains a block with Hamming weight greater than τ_2, then it is not correctable with certainty. Consequently, the approach that will be taken to compute X considers only those error sequences which are potentially correctable and not equivalent to any other error sequence under consideration.

Before X may be computed, it is required to calculate $P_c(w)$, the probability that a combined block $[\mathcal{B}_j^{(i-1)} \mid \mathcal{B}_j^{(i)}]$ containing w errors is correctable. This quantity depends upon the error correction code $C_1^{(2)}$ as well as its decoder implementation. For $w \leq t_2, P_c(w) = 1$ and for $w > \tau_2, P_c(w) = 0$. In the range $t_2 < w \leq \tau_2$, $P_c(w)$ is

[1] **Note that the weight partition is assumed to refer to the Hamming weight partition of a packet. These terms tend to be used interchangeably.**

computed by exhaustive enumeration of the combined blocks containing w errors. In this case,

$$P_c(w) = \frac{\mathcal{N}(\mathcal{B} \text{ correctable})}{\binom{2n}{w}}, \quad t_2 < w \leq \tau_2.$$

where the expression $\mathcal{N}(\mathcal{B}$ correctable) denotes the number of combined blocks $\mathcal{B} = [\mathcal{B}^{(i-1)} \mid \mathcal{B}^{(i-1)}]$ with Hamming weight w that are correctable (decoded to $\underline{0}$).

Recall that the computation of $X = \Pr(\underline{S}_1 \neq \underline{0}, \underline{S}_2 = \underline{0}, \text{No PD Errors})$ may be expressed as

$$\begin{aligned} X &= \sum_{j=1}^{2^N-2^k} \Pr(\underline{\epsilon}_j \notin \mathcal{C}_0^{(x)}, \underline{\epsilon}_{pd}^{(j)} = \underline{0}) \\ &= \sum_{j=1}^{2^N-2^k} \Pr(\underline{\epsilon}_j \notin \mathcal{C}_0^{(x)}, \underline{\epsilon}_j \text{ is correctable}), \end{aligned}$$

where $\underline{\epsilon}_j$ is the j-th N-bit sequence which is not a codeword in the code $\mathcal{C}_0^{(x)}$. Let Λ be the set of all words which are not codewords in the code $\mathcal{C}_0^{(x)}$ and, let Λ_i be the subset of Λ containing those with Hamming weight i. Clearly, the Λ_i form disjoint subsets which cover Λ. That is, $\Lambda_i \cap \Lambda_j = \emptyset$ for $i \neq j$, and

$$\bigcup_{i=1}^{N} \Lambda_i = \Lambda.$$

Since the code $\mathcal{C}_0^{(x)}$ is linear, it must contain the zero-sequence; consequently, $\Lambda_0 = \emptyset$ and X may be computed as follows:

$$\begin{aligned} X &= \sum_{j=1}^{2^N-2^k} \Pr(\underline{\epsilon}_j \notin \mathcal{C}_0^{(x)}, \underline{\epsilon}_j \text{ is correctable}) \\ &= \sum_{\underline{e} \in \Lambda} \Pr(\underline{e} \text{ is correctable}) \\ &= \sum_{k=1}^{N} \sum_{\underline{e} \in \Lambda_k} \Pr(\underline{e} \text{ is correctable}) \\ &= \sum_{k=1}^{N} \mid \Lambda_k \mid \Pr(\underline{e} \text{ is correctable}, w(\underline{e}) = k), \end{aligned}$$

where $\mid \Lambda_k \mid$ is the cardinality of the set Λ_k and $w(\underline{e})$ is the Hamming weight of the error sequence $\underline{e}$. Also, let $B_k^{(x)} = \mid \Lambda_k \mid$ so that, if $A_k^{(x)}$ is the number of codewords with weight k in the code $\mathcal{C}_0^{(x)}$, then $A_k^{(x)} + B_k^{(x)} = \binom{N}{k}$, the total number of error sequences with Hamming weight k. Now,

$$X = \sum_{k=1}^{N} B_k^{(x)} \Pr(\underline{e} \text{ is correctable}, w(\underline{e}) = k).$$

The $\Pr(\underline{e}$ is correctable, $w(\underline{e}) = k)$ imposes two constraints on the error sequences $\underline{e}$ which need to be considered in the computation of X.

1. The Hamming weight of the error sequence $\underline{e}$ must be k.
2. The requirement that $\underline{e}$ be correctable restricts consideration to only those error sequences which are potentially correctable. Suppose that an error sequence $\underline{e}'$ contains a block with more than τ_2 errors. Then the $\Pr(\underline{e}'$ is correctable, $w(\underline{e}') = k)$ is zero because $\underline{e}'$ is uncorrectable with certainty. Therefore, there is no point in considering error sequences which are not potentially correctable.

These two constraints are reminiscent of the integer partitioning problem described in Section 2.3. In particular, suppose that the set of potentially correctable partitions of $\underline{e}$ with Hamming weight k is denoted by $\pi^{(k)}$. Let the cardinality of such a set be denoted, once again, by $\mid \pi^{(k)} \mid$. Further, let the j-th partition of the set $\pi^{(k)}$ be $\pi_j^{(k)}$. Finally, the expression $\underline{e} \sim \pi_j^{(k)}$ is interpreted to mean that the Hamming weight partition among the n-bit blocks comprising the error sequence $\underline{e}$ is equivalent to the partition $\pi_j^{(k)}$. The computation of X may then be expressed as

$$X = \sum_{k=1}^{N\tau_2/n} B_k^{(x)} \sum_{j=1}^{|\pi^{(k)}|} \Pr(\underline{e} \text{ is correctable}, \underline{e} \sim \pi_j^{(k)}). \tag{4.3}$$

Equation 4.3 is the key to the computation of X. As previously described, the computation of the $\Pr(\underline{e}$ is correctable, $w(\underline{e}) = k)$ becomes a summation of the probability that each potentially correctable error sequence is successfully corrected. If $\pi_j^{(k)}$ contains fewer parts than blocks in $\underline{e}$, then it is padded with a sufficient number of zero-parts. Note that the upper limit on the summation indexed by k is $N\tau_2/n$, instead of N as might have been expected. This is because there exists no potentially correctable error sequence with Hamming weight greater than $N\tau_2/n$. Now, the effect of the channel states and the selective repeat retransmission strategy are easily integrated into this expression for X.

$$X = \sum_{k=1}^{N\tau_2/n} B_k^{(x)} \sum_{j=1}^{|\pi^{(k)}|} \sum_{\Omega_i} \sum_{\Omega_{i-\delta}} \Pr(\underline{e} \text{ is correctable}, \underline{e} \sim \pi_j^{(k)}, \Omega_{i-\delta}, \Omega_i),$$

where, as before, the expression $\underline{e} \sim \pi_j^{(k)}$ is interpreted to mean that the Hamming weight k of the error sequence $\underline{e}$ is partitioned like $\pi_j^{(k)}$. The current state of the channel is denoted by Ω_i and $\Omega_{i-\delta}$ is its state δ transitions in the past. The parameter δ refers to the roundtrip retransmission delay in the selective repeat request retransmission strategy. The stop-and-wait retransmission strategy corresponds to

$\delta = 1$. The problem then becomes the computation of

$$\begin{aligned}&\Pr(\underline{e} \text{ is correctable}, \underline{e} \sim \pi_j^{(k)}, \Omega_{i-\delta}, \Omega_i)\\ &= \Pr(\underline{e} \text{ is correctable} \mid \underline{e} \sim \pi_j^{(k)}, \Omega_{i-\delta}, \Omega_i)\\ &\quad \cdot \Pr(\underline{e} \sim \pi_j^{(k)} \mid \Omega_{i-\delta}, \Omega_i) \Pr(\Omega_{i-\delta}, \Omega_i)\end{aligned}$$

The quantity $\Pr(\Omega_{i-\delta}, \Omega_i)$ is the δ-step transition probability from state $\Omega_{i-\delta}$ to Ω_i and is computed in Appendix B.3. The other quantities may be easily computed as shown below.

The $\Pr(\underline{e} \sim \pi_j^{(k)} \mid \Omega_{i-\delta}, \Omega_i)$ is independent of the state $\Omega_{i-\delta}$, so that

$$\Pr(\underline{e} \sim \pi_j^{(k)} \mid \Omega_{i-\delta}, \Omega_i) = \Pr(\underline{e} \sim \pi_j^{(k)} \mid \Omega_i).$$

Let $\pi_{ji}^{(k)}$ be the number of parts equal to i in the j-th partition of the integer k. Then,

$$\Pr(\underline{e} \sim \pi_j^{(k)} \mid \Omega_i) = \frac{\mathcal{N}(\underline{e} \sim \pi_j^{(k)})}{\mathcal{N}(w(\underline{e}) = k)} \varepsilon_i^k (1 - \varepsilon_i)^{N-k},$$

where $\mathcal{N}(\cdot)$ is the number of error sequences satisfying the indicated requirement, and ε_i is the probability of error in the current state of the channel, Ω_i. Then,

$$\begin{aligned}\mathcal{N}(w(\underline{e}) = k) &= \binom{N}{k}\\ \mathcal{N}(\underline{e} \sim \pi_j^{(k)}) &= \frac{\left(\frac{N}{n}\right)!}{\prod_{l=0}^{\tau_2} \pi_{jl}^{(k)}!} \prod_{l=0}^{\tau_2} \left[\binom{n}{l}\right]^{\pi_{jl}^{(k)}};\end{aligned}$$

therefore,

$$\Pr(\underline{e} \sim \pi_j^{(k)} \mid \Omega_i) = \frac{\left(\frac{N}{n}\right)!}{\binom{N}{k}} \prod_{l=0}^{\tau_2} \frac{\left[\binom{n}{l}\right]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!} \varepsilon_i^k (1 - \varepsilon_i)^{N-k}.$$

Now, it is required to compute $\Pr(\underline{e} \text{ is correctable} \mid \underline{e} \sim \pi_j^{(k)}, \Omega_{i-\delta}, \Omega_i)$. Let p be the number of errors in the previously received block $\mathcal{B}^{(i-1)}$ of the combined block $[\mathcal{B}^{(i-1)} \mid \mathcal{B}^{(i)}]$. Then,

$$\begin{aligned}&\Pr(\underline{e} \text{ is correctable} \mid \underline{e} \sim \pi_j^{(k)}, \Omega_{i-\delta}, \Omega_i)\\ &= \Pr(\underline{e} \text{ is correctable} \mid \underline{e} \sim \pi_j^{(k)}, \Omega_{i-\delta})\\ &= \prod_{l=0}^{\tau_2} [\Pr(p \le \tau_2 - l)]^{\pi_{jl}^{(k)}}\\ &= \prod_{l=0}^{\tau_2} \left[\sum_{\alpha=0}^{\tau_2 - l} P_c(\alpha + l) \Pr(p = \alpha)\right]^{\pi_{jl}^{(k)}}\end{aligned}$$

$$= \prod_{l=0}^{\tau_2} \left[\sum_{\alpha=0}^{\tau_2 - l} \binom{n}{\alpha} P_c(\alpha + l)\varepsilon_{i-\delta}^{\alpha}(1 - \varepsilon_{i-\delta})^{n-\alpha} \right]^{\pi_{jl}^{(k)}},$$

where $\varepsilon_{i-\delta}$ is the probability of error in the state of the channel δ transitions in the past and $P_c(w)$, as described earlier in this section, reflects the error correction capability of the code $C_1^{(2)}$. The computation of $\Pr(\underline{S}_1 \neq \underline{0}, \underline{S}_2 = \underline{0}, \text{No PD Errors})$ is now complete:

$$\Pr(\underline{S}_1 \neq \underline{0}, \underline{S}_2 = \underline{0}, \text{No PD Errors})$$
$$= \left(\frac{N}{n}\right)! \sum_{k=1}^{N\tau_2/n} \frac{B_k^{(x)}}{\binom{N}{k}} \sum_{j=1}^{|\pi^{(k)}|} \sum_{\Omega_i} \varepsilon_i^k (1 - \varepsilon_i)^{N-k}$$
$$\cdot \sum_{\Omega_{i-\delta}} \Pr(\Omega_{i-\delta}, \Omega_i) \prod_{l=0}^{\tau_2} \frac{\left[\binom{n}{l} \sum_{\alpha=0}^{\tau_2 - l} P_c(\alpha + l)\varepsilon_{i-\delta}^{\alpha}(1 - \varepsilon_{i-\delta})^{n-\alpha}\right]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!}.$$

The transition probability $C_2^{(x)}$ is then,

$$C_2^{(x)} = \frac{1}{P_{gb} + P_{bg}} \left[P_{bg}(1 - \epsilon_g)^N + P_{gb}(1 - \epsilon_b)^N \right]$$
$$+ \left(\frac{N}{n}\right)! \sum_{k=1}^{N\tau_2/n} \frac{B_k^{(x)}}{\binom{N}{k}} \sum_{j=1}^{|\pi^{(k)}|} \sum_{\Omega_i} \varepsilon_i^k (1 - \varepsilon_i)^{N-k}$$
$$\cdot \sum_{\Omega_{i-\delta}} \Pr(\Omega_{i-\delta}, \Omega_i) \prod_{l=0}^{\tau_2} \frac{\left[\binom{n}{l} \sum_{\alpha=0}^{\tau_2 - l} P_c(\alpha + l)\varepsilon_{i-\delta}^{\alpha}(1 - \varepsilon_{i-\delta})^{n-\alpha}\right]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!}$$

4.5.2 Computation of $Q_2^{(x)}$

Recall that $Q_2^{(x)} = \Pr(\underline{S}_1 = \underline{0}, \text{PI Errors}) + \Pr(\underline{S}_1 \neq \underline{0}, \underline{S}_2 = \underline{0}, \text{PD Errors})$. This is interpreted to be the probability that the receiver delivers a message containing errors to the data sink on the second and succeeding transmissions.

The $\Pr(\underline{S}_1 = \underline{0}, \text{PI Errors})$ is simply the probability of undetected errors after inversion. It is similar to the transition probability Q_1 which was computed in Section 4.4, except that the error detection code is linearly altered by $\boldsymbol{M}_x$ of the error correction code; thus,

$$\Pr(\underline{S}_1 = \underline{0}, \text{PI Errors}) = \frac{1}{P_{gb} + P_{bg}} \sum_{k=d_x}^{N} A_k^{(x)} \left[P_{bg}\epsilon_g^k (1 - \epsilon_g)^{N-k} + P_{gb}\epsilon_b^k (1 - \epsilon_b)^{N-k} \right].$$

The $\Pr(\underline{S}_1 \neq \underline{0}, \underline{S}_2 = \underline{0}, \text{PD Errors})$ is interpreted to be the probability that an error sequence which is not a codeword in the code $\mathcal{C}_0^{(x)}$ is decoded into a nonzero codeword in the unaltered error detection code $\mathcal{C}_0$. As before, at least two approaches may be taken to compute this probability. The first consists in generating every possible error sequence which is not a codeword in the code $\mathcal{C}_0^{(x)}$ and then computing the probability that this sequence is decoded into a nonzero codeword of the code $\mathcal{C}_0$. Again, this approach is practical only for situations where $N < 15$. The second approach consists in grouping together equivalent decoded error sequences. Let the expression $\underline{\epsilon}_j \notin \mathcal{C}_0^{(x)}$ be interpreted to be the j-th N-bit error sequence $\underline{\epsilon}_j$ which is not a codeword in the code $\mathcal{C}_0^{(x)}$, and let $\underline{\epsilon}_{pd}^{(j)}$ be the decoded error sequence at the output of the error correction code decoder when the current error sequence is $\underline{\epsilon}_j$. Then,

$$\begin{aligned} Y &= \Pr(\underline{S}_1 \neq \underline{0}, \underline{S}_2 = \underline{0}, \text{PD Errors}) \\ &= \sum_{j=1}^{2^N-2^k} \Pr(\underline{\epsilon}_j \notin \mathcal{C}_0^{(x)}, \underline{\epsilon}_{pd}^{(j)} \in \tilde{\mathcal{C}}_0), \end{aligned}$$

where $\tilde{\mathcal{C}}_0$ is the unaltered error detection code without the zero codeword. To compute Y, begin by removing the restriction that $\underline{\epsilon}_j \notin \mathcal{C}_0^{(x)}$. Now, let

$$Y' = \sum_{j=1}^{2^N} \Pr(\underline{\epsilon}_{pd}^{(j)} \in \tilde{\mathcal{C}}_0),$$

where $\underline{\epsilon}_j$ is now any N-bit error sequence. So

$$\begin{aligned} Y &= \sum_{j=1}^{2^N-2^k} \Pr(\underline{\epsilon}_j \notin \mathcal{C}_0^{(x)}, \underline{\epsilon}_{pd}^{(j)} \in \tilde{\mathcal{C}}_0) \\ &= \sum_{j=1}^{2^N} \Pr(\underline{\epsilon}_{pd}^{(j)} \in \tilde{\mathcal{C}}_0) - \sum_{j=0}^{2^N-2^k-1} \Pr(\underline{\epsilon}_j \in \mathcal{C}_0^{(x)}, \underline{\epsilon}_{pd}^{(j)} \in \tilde{\mathcal{C}}_0) \\ &= Y' - \sum_{j=0}^{2^N-2^k-1} \Pr(\underline{\epsilon}_j \in \mathcal{C}_0^{(x)}, \underline{\epsilon}_{pd}^{(j)} \in \tilde{\mathcal{C}}_0) \\ Y &= Y' - \Pr(\underline{\epsilon}_{pd}^{(0)} \in \tilde{\mathcal{C}}_0) - \sum_{j=0}^{2^N-2^k-1} \Pr(\underline{\epsilon}_j \in \tilde{\mathcal{C}}_0^{(x)}, \underline{\epsilon}_{pd}^{(j)} \in \tilde{\mathcal{C}}_0), \end{aligned}$$

where $\tilde{\mathcal{C}}_0^{(x)}$ is the code $\mathcal{C}_0^{(x)}$ without the $\underline{0}$ codeword and $\underline{\epsilon}_{pd}^{(0)}$ is the output of the error correction code decoder when the currently received error sequence is $\underline{0}$. It is usually the case that the zero error sequence is much more likely to be received than an error sequence equal to a codeword in the code $\tilde{\mathcal{C}}_0^{(x)}$. That is,

$$\Pr(\underline{\epsilon}_0 \in \tilde{\mathcal{C}}_0^{(x)}, \underline{\epsilon}_{pd}^{(0)} \in \tilde{\mathcal{C}}_0) \gg \sum_{j=1}^{2^N-2^k-1} \Pr(\underline{\epsilon}_j \in \tilde{\mathcal{C}}_0^{(x)}, \underline{\epsilon}_{pd}^{(j)} \in \tilde{\mathcal{C}}_0),$$

where $\underline{\epsilon}_0$ is the zero error sequence and $\underline{\epsilon}_{pd}^{(0)}$ is the output of the decoder when the currently received error sequence is $\underline{0}$. Thus, the expression for Y may be approximated by

$$Y \approx Y' - \Pr(\underline{\epsilon}_{pd}^{(0)} \in \tilde{\mathcal{C}}_0).$$

To complete the approximate computation of Y, it is required to compute two quantities: Y' and $\Pr(\underline{\epsilon}_{pd}^{(0)} \in \tilde{\mathcal{C}}_0)$. Recall,

$$\begin{aligned} Y' &= \sum_{j=1}^{2^N-2^k} \Pr(\underline{\epsilon}_{pd}^{(j)} \in \hat{\mathcal{C}}_0) \\ &= \sum_{k=d}^{N} \frac{A_k}{\binom{N}{k}} \Pr(k \text{ PD Errors}), \end{aligned}$$

where A_k is the number of codewords in the error detection $\mathcal{C}_0$ with Hamming weight k, d is the minimum distance of the code $\mathcal{C}_0$, and $\Pr(k$ PD Errors) is the probability of k errors at the output of the decoder. The problem then becomes the computation of $\Pr(k$ PD Errors).

Suppose that k errors ($d \leq k \leq N$) remain uncorrected in the decoded packet. These errors may be partitioned in a manner similar to the way in which the errors in the error sequence $\underline{e}$ were partitioned in the computation of $C_2^{(x)}$. So, instead of partitioning an packet input to the decoder, the errors in the packet at the output of the decoder $[\mathcal{B}_1^{(o)} \mid \ldots \mid \mathcal{B}_{N/n}^{(o)}]$ will be partitioned. It is then possible to define the $\Pr(k$ PD Errors) in terms of the valid partitions of the k errors at the output of the decoder. In this context, a valid partition consists of at most N/n parts with no part greater than n. Let the set of valid partitions of the integer k (as defined in this context) be denoted by $\pi^{(k)}$. Further, let the cardinality of this set be denoted, once again, by $\mid \pi^{(k)} \mid$ and the j-th element of the set $\pi^{(k)}$ be denoted $\pi_j^{(k)}$. Then it is possible to compute $\Pr(k$ PD Errors) as

$$\Pr(k \text{ PD Errors}) = \sum_{j=1}^{|\pi^{(k)}|} \mathcal{N}(\pi_j^{(k)}) \Pr(\pi_j^{(k)}),$$

where $\mathcal{N}(\pi_j^{(k)})$ is the number of sequences with partition $\pi_j^{(k)}$ and $\Pr(\pi_j^{(k)})$ is the probability that the sequence at the output of the decoder is partitioned like $\pi_j^{(k)}$. Similar to the computation of $C_2^{(x)}$, the effect of the channel and the retransmission strategy are easily integrated into this expression:

$$\Pr(k \text{ PD Errors}) = \sum_{\Omega_{i-\delta}} \sum_{\Omega_i} \Pr(\Omega_{i-\delta}, \Omega_i) \sum_{j=1}^{|\pi^{(k)}|} \mathcal{N}(\pi_j^{(k)}) \Pr(\pi_j^{(k)}),$$

where δ, as before, represents the roundtrip delay of the selected retransmission strategy. The quantity $\mathcal{N}(\pi_j^{(k)})$ is easily computed to be

$$\mathcal{N}(\pi_j^{(k)}) = \frac{\left(\frac{N}{n}\right)!}{\prod_{l=0}^{n} \pi_{jl}^{(k)}!}.$$

The probability of the partition $\pi_j^{(k)}$ is

$$\Pr(\pi_j^{(k)}) = \prod_{l=0}^{n} [P_{pd}(l)]^{\pi_{jl}^{(k)}},$$

where $P_{pd}(l)$ is the probability that the block $\mathcal{B}^{(o)}$ at the output of the decoder contains l errors. The approach that will be taken to compute this probability will also be useful when $\Pr(\underline{\epsilon}_{pd}^{(0)} \in \tilde{\mathcal{C}}_0)$ is calculated. To compute the probability distribution $P_{pd}(l)$ it is required to exhaustively enumerate all possible $2n$-bit combined blocks $[\mathcal{B}^{(i-\delta)} \mid \mathcal{B}^{(i)}]$ and then to determine the output of the decoder. Then, a table as shown in Table 4.1 may be constructed. In this table, $\mathcal{N}(w_{i-\delta}, w_i, w_o)$ is the number of n-bit blocks at the output of the decoder containing w_o errors such that the number of errors in the block $\mathcal{B}^{(i-\delta)}$ at the input of the decoder is $w_{i-\delta}$ and the number of errors in the block $\mathcal{B}^{(i)}$ is w_i.

$w_{i-\delta}$	w_i	w_o	$\mathcal{N}(w_{i-\delta}, w_i, w_o)$
0	0	0	$\mathcal{N}(0,0,0)$
0	0	1	$\mathcal{N}(0,0,1)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
0	0	n	$\mathcal{N}(0,0,n)$
0	1	0	$\mathcal{N}(0,1,0)$
0	1	1	$\mathcal{N}(0,1,1)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
0	1	n	$\mathcal{N}(0,1,n)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	n	0	$\mathcal{N}(n,n,0)$
n	n	1	$\mathcal{N}(n,n,1)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	n	n	$\mathcal{N}(n,n,n)$

Table 4.1: Error Correction Code $\mathcal{C}_1^{(2)}$ Decoder Output

Using this table, it is a simple matter to compute $P_{pd}(l)$:

$$P_{pd}(l) = \sum_{j=0}^{n}\sum_{k=0}^{n} \frac{\mathcal{N}(j,k,l)}{\binom{n}{j}\binom{n}{k}} \Pr(w_{i-\delta}=j, w_i=k)$$

$$P_{pd}(l) = \sum_{j=0}^{n}\sum_{k=0}^{n} \frac{\mathcal{N}(j,k,l)}{\binom{n}{j}\binom{n}{k}} \Pr(w_{i-\delta}=j)\Pr(w_i=k)$$

$$P_{pd}(l) = \sum_{j=0}^{n}\sum_{k=0}^{n} \mathcal{N}(j,k,l)\varepsilon_{i-\delta}^{j}(1-\varepsilon_{i-\delta})^{n-j}\varepsilon_i^{k}(1-\varepsilon_i)^{n-k};$$

therefore

$$\Pr(k \text{ PD Errors}) = \left(\frac{N}{n}\right)! \sum_{\Omega_{i-\delta}}\sum_{\Omega_i} \Pr(\Omega_{i-\delta},\Omega_i) \sum_{j=1}^{|\pi^{(k)}|} \frac{\prod_{l=0}^{n}[P_{pd}(l)]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!},$$

and

$$Y' = \left(\frac{N}{n}\right)! \sum_{\Omega_{i-\delta}}\sum_{\Omega_i} \Pr(\Omega_{i-\delta},\Omega_i) \sum_{k=d}^{N} \frac{A_k}{\binom{N}{n}} \sum_{j=1}^{|\pi^{(k)}|} \frac{\prod_{l=0}^{n}[P_{pd}(l)]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!}.$$

Having computed Y', there remains the computation of $\Pr(\underline{\epsilon}_{pd}^{(0)} \in \tilde{\mathcal{C}}_0)$. This is the same as Y' except that the number of errors in the current input block to the decoder $\mathcal{B}^{(i)}$ is known to be 0 (since the error sequence is the all- zero sequence). This requires the following simple modifications:

$$\Pr(\underline{\epsilon}_{pd}^{(0)} \in \tilde{\mathcal{C}}_0) = \sum_{k=d}^{N} \frac{A_k}{\binom{N}{k}} \Pr(k \text{ PD Errors}, \underline{\epsilon}_j = \underline{0}),$$

where $\Pr(k \text{ PD Errors}, \underline{\epsilon}_j = \underline{0})$ is the probability of k post-decoding errors when the current input error sequence is $\underline{0}$. As before

$$\Pr(k \text{ PD Errors}, \underline{\epsilon}_j = \underline{0}) = \sum_{\Omega_{i-\delta}}\sum_{\Omega_i} \Pr(\Omega_{i-\delta},\Omega_i) \sum_{j=1}^{|\pi^{(k)}|} \mathcal{N}(\pi_j^{(k)})\Pr(\pi_j^{(k)}),$$

and

$$\mathcal{N}(\pi_j^{(k)}) = \frac{\left(\frac{N}{n}\right)!}{\prod_{l=0}^{n} \pi_{jl}^{(k)}!}.$$

The difference between this case and the previous one is in the quantity $\Pr(\pi_j^{(k)})$ which must reflect that fact that the currently received error sequence is $\underline{0}$. Consequently,

$$\Pr(\pi_j^{(k)}) = \prod_{l=0}^{n} [P_{pd0}(l)]^{\pi_{jl}^{(k)}},$$

where $P_{pd0}(l)$ is the probability that the block at the output of the decoder $\mathcal{B}^{(o)}$ contains l errors when the currently received error sequence is equal to $\underline{0}$. This quantity is easily computed by considering, in Table 4.1, only those rows with $w_i = 0$. Then,

$$\begin{aligned} P_{pd0}(l) &= \sum_{j=0}^{n} \frac{\mathcal{N}(j,0,l)}{\binom{n}{j}} \Pr(w_{i-\delta} = j) \\ &= \sum_{j=0}^{n} \mathcal{N}(j,0,l)\varepsilon_{i-\delta}^{j}(1-\varepsilon_{i-\delta})^{n-j}. \end{aligned}$$

The $\Pr(\underline{\epsilon}_{pd}^{(0)} \in \tilde{\mathcal{C}}_0)$ is, therefore,

$$\Pr(\underline{\epsilon}_{pd}^{(0)} \in \tilde{\mathcal{C}}_0) = \left(\frac{N}{n}\right)! \sum_{\Omega_{i-\delta}} \sum_{\Omega_i} \Pr(\Omega_{i-\delta}, \Omega_i) \sum_{k=d}^{N} \frac{A_k}{\binom{N}{k}} \sum_{j=1}^{|\pi^{(k)}|} \prod_{l=0}^{n} \frac{[P_{pd0}(l)]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!}.$$

The approximation for $\Pr(\underline{S}_1 \neq \underline{0}, \underline{S}_2 = \underline{0}, \text{PD Errors})$ is complete:

$$\begin{aligned} &\Pr(\underline{S}_1 \neq \underline{0}, \underline{S}_2 = \underline{0}, \text{PD Errors}) \\ &\approx \left(\frac{N}{n}\right)! \sum_{\Omega_{i-\delta}} \sum_{\Omega_i} \Pr(\Omega_{i-\delta}, \Omega_i) \sum_{k=d}^{N} \frac{A_k}{\binom{N}{k}} \sum_{j=1}^{|\pi^{(k)}|} \left[\prod_{l=0}^{n} \frac{[P_{pd}(l)]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!} - \prod_{l=0}^{n} \frac{[P_{pd0}(l)]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!} \right]. \end{aligned}$$

The transition probability $Q_2^{(x)}$ is then,

$$\begin{aligned} Q_2^{(x)} = &\frac{1}{P_{gb} + P_{bg}} \sum_{k=d_x}^{N} A_k^{(x)} \left[P_{bg}\epsilon_g^k(1-\epsilon_g)^{N-k} + P_{gb}\epsilon_b^k(1-\epsilon_b)^{N-k} \right] \\ &+ \left(\frac{N}{n}\right)! \sum_{\Omega_{i-\delta}} \sum_{\Omega_i} \Pr(\Omega_{i-\delta}, \Omega_i) \sum_{k=d}^{N} \frac{A_k}{\binom{N}{k}} \sum_{j=1}^{|\pi^{(k)}|} \left[\prod_{l=0}^{n} \frac{[P_{pd}(l)]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!} - \prod_{l=0}^{n} \frac{[P_{pd0}(l)]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!} \right] \end{aligned}$$

4.5.3 Computation of $P_2^{(x)}$

This is the probability of a retransmission on the first and following retransmissions. It is simply

$$P_2^{(x)} = 1 - C_2^{(x)} - Q_2^{(x)}$$

4.6 An Example – Throughput and Reliability Computation

This example illustrates the computation of the throughput and reliability of a simple GH-ARQ II error control scheme. The error detection code $\mathcal{C}_0$ is the BCH (15,11) code and the error correction code is $\boldsymbol{M} = [\boldsymbol{M}_1 \mid \boldsymbol{M}_2]$, where $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ are as defined in Example 3.1. The weight distributions of $\mathcal{C}_0$ and $\mathcal{C}_0^{(x)}$ $(x = 1, 2)$ are also computed in that section. Since the submatrix $\boldsymbol{M}_1$ of the code $\mathcal{C}_1^{(2)}$ is seen to have no effect upon the performance of the code $\mathcal{C}_0$, there is a fairly good match between the codes in this error control scheme. The error correction code decoder simply selects the codeword in $\mathcal{C}_1^{(2)}$ which is closest to the combined block $\mathcal{B}^{(i-\delta)} \mid \mathcal{B}^{(i)}$. This is feasible as there are only $2^5 = 32$ codewords in this particular error correction code. The error correction capability of the code $\mathcal{C}_1^{(2)}$ is easily computed by exhaustive enumeration of all possible 10-bit sequences; Table 4.2 summarizes $P_c(w)$ for this code. From this table it is seen that $\tau_2 = 2$

w	$P_c(w)$
0	1.0
1	1.0
2	0.06667
$3 \le w \le 10$	0.0

Table 4.2: Error Correction Capability of $\mathcal{C}_1^{(2)}$

Having specified the codes in this example, there remains the specification of the burst-noise channel model parameters. Suppose that fairly dense bursts occur in the channel, so that $P_b = 0.05$. Also suppose the average error rate is $\bar{\epsilon} = 10^{-2}$ and the transition probability from the bad to the good state, $P_{bg} = 0.025$. Using the expressions from Section 2.1, the parameters of this channel model are easily computed:

$$\begin{aligned}
P_{gb} &= \frac{P_{bg} P_b}{1 - P_b} = 1.316 \times 10^{-3} \\
P_{bg} &= 0.025 \\
\epsilon_g &= \bar{\epsilon} P_b = 5 \times 10^{-4} \\
\epsilon_b &= \bar{\epsilon} \left[\frac{P_b^2 - P_b + 1}{P_b} \right] = 0.1905
\end{aligned}$$

Also, let the retransmission strategy be a simple stop-and-wait scheme so that $\delta = 1$.

The transition probabilities C_1, Q_1, and P_1 are easily computed as shown below (refer to Section 4.4 for details).

$$
\begin{aligned}
C_1 &= \frac{1}{P_{gb}+P_{bg}}\left[P_{bg}(1-\epsilon_g)^N + P_{gb}(1-\epsilon_b)^N\right] \\
&= 0.50927 \\
Q_1 &= \frac{1}{P_{gb}+P_{bg}}\sum_{k=d_1}^{N} A_k^{(1)}\left[P_{bg}\epsilon_g^k(1-\epsilon_g)^{N-k} + P_{gb}\epsilon_b^k(1-\epsilon_b)^{N-k}\right] \\
&= 1.1135\times 10^{-2} \\
P_1 &= 1 - Q_1 - C_1 \\
&= 0.47959
\end{aligned}
$$

To compute the transition probability $C_2^{(x)}$ as described in Section 4.5 it is required to enumerate the potentially correctable partitions for the integers up to and including $N\tau_2/n = 6$. This enumeration may be achieved by using the integer partition trees described in Section 2.3. The valid partitions are summarized in Table 4.3.

k	Partitions	π
1	$\{1\}$	$\pi_1^{(1)} = (2,1,0)$
2	$\{1,1\}$ $\{2\}$	$\pi_1^{(2)} = (1,2,0)$ $\pi_2^{(2)} = (2,0,1)$
3	$\{1,2\}$ $\{1,1,1\}$	$\pi_1^{(3)} = (1,1,1)$ $\pi_2^{(3)} = (0,3,0)$
4	$\{1,1,2\}$	$\pi_1^{(4)} = (0,2,1)$
5	$\{1,2,2\}$	$\pi_1^{(5)} = (0,1,2)$
6	$\{2,2,2\}$	$\pi_1^{(6)} = (0,0,3)$

Table 4.3: Valid Partitions for the Computation of $C_2^{(x)}$

From this table, it is also seen that the number of sequences that need to be considered in order to compute $C_2^{(x)}$ decreases from $2^{15} - 2^{11} = 30720$ to 8 sequences. This represents a significant reduction in complexity. The computation of the transition probability $Q_2^{(x)}$ requires the consideration of more sequences than does that of $C_2^{(x)}$; 53 sequences must be considered as enumerated in Table 4.4.

Using the expressions which were derived in Section 4.5, the transition probabilities $C_2^{(x)}$, $Q_2^{(x)}$, and $P_2^{(x)}$ for $x = 1,2$ are found to be:

$$
\begin{aligned}
C_2^{(1)} &= 0.74138 \\
C_2^{(2)} &= 0.73973 \\
Q_2^{(1)} &= 1.7987\times 10^{-2} \\
Q_2^{(2)} &= 2.1450\times 10^{-2} \\
P_2^{(1)} &= 0.24063 \\
P_2^{(2)} &= 0.23882
\end{aligned}
$$

k	Partitions
1	{1}
2	{1,1} {2}
3	{1,1,1} {1,2} {3}
4	{1,1,2} {1,3} {2,2} {4}
5	{1,1,3} {1,2,2} {1,4} {2,3} {5}
6	{1,1,4} {1,2,3} {2,2,2} {2,4} {3,3}
7	{1,1,5} {1,2,4} {1,3,3} {2,2,3} {2,5}
8	{1,2,5} {1,3,4} {2,2,4} {2,3,3} {3,5} {4,4}
9	{1,3,5} {1,4,4} {2,2,5} {2,3,4} {3,3,3} {4,5}
10	{1,4,5} {2,3,5} {2,4,4} {3,3,4} {5,5}
11	{1,5,5} {2,4,5} {3,3,5} {3,4,4}
12	{2,5,5} {3,4,5} {4,4,4}
13	{3,5,5} {4,4,5}
14	{4,5,5}
15	{5,5,5}

Table 4.4: Valid Partitions for the Computation of $Q_2^{(x)}$

The computation of the throughput and reliability of this error control scheme is then completed by substituting the values for the transition probabilities into the expressions for η and P_{ud} found in Section 4.3. Then,

$$\begin{aligned} \eta &= 0.6134 \\ P_{ud} &= 2.424 \times 10^{-2}. \end{aligned}$$

For purposes of comparison, the throughput and reliability of the simple ARQ scheme described in Section 2.2 are computed given these channel parameters.

$$\begin{aligned} \eta(\text{ARQ}) &= 0.5204 \\ P_{ud}(\text{ARQ}) &= 2.140 \times 10^{-2} \end{aligned}$$

These results demonstrate an appreciable increase in throughput efficiency of the GH-ARQ II scheme over the simple ARQ scheme with no error correction. As expected, the reliability of the GH-ARQ II scheme is slightly worse than that of the simple ARQ scheme. This is a consequence of two factors.

1. The error correction code $C_1^{(2)}$ (in particular M_2) has a non-negligible effect upon the error detection code C_0.
2. On the first and following retransmissions, there are two "opportunities" for the receiver to make an undetected error: after the inversion and, possibly, after error correction decoding, if the inversion fails or is not possible.

The effect of the error correction coding in the GH-ARQ II scheme is manifested in the values of the transition probabilities. On the first transmission, the probability that a retransmission will be requested, P_1, is 0.48. On the second and following transmissions this probability, $P_2^{(x)}$, decreases to approximately 0.24. Most of this difference is attributable to the error correction capability of the GH-ARQ II scheme.

4.7 Summary

This section summarizes the expressions for the transition probabilities of the depth-2 GH-ARQ II error control scheme. Recall that the throughput and reliability of this scheme were described in terms of these transition probabilities in Section 4.3.

$$
\begin{aligned}
C_1 &= \frac{1}{P_{gb}+P_{bg}}\left[P_{bg}(1-\epsilon_g)^N + P_{gb}(1-\epsilon_b)^N\right] \\
Q_1 &= \frac{1}{P_{gb}+P_{bg}}\sum_{k=d_1}^{N} A_k^{(1)}\left[P_{bg}\epsilon_g^k(1-\epsilon_g)^{N-k} + P_{gb}\epsilon_b^k(1-\epsilon_b)^{N-k}\right] \\
P_1 &= 1 - Q_1 - C_1 \\
C_2^{(x)} &= \frac{1}{P_{gb}+P_{bg}}\left[P_{bg}(1-\epsilon_g)^N + P_{gb}(1-\epsilon_b)^N\right] \\
&\quad + \left(\frac{N}{n}\right)! \sum_{k=1}^{N\tau_2/n} \frac{B_k^{(x)}}{\binom{N}{k}} \sum_{j=1}^{|\pi^{(k)}|} \sum_{\Omega_i} \varepsilon_i^k (1-\varepsilon_i)^{N-k} \\
&\qquad \cdot \sum_{\Omega_{i-\delta}} \Pr(\Omega_{i-\delta},\Omega_i) \prod_{l=0}^{\tau_2} \frac{\left[\binom{n}{l}\sum_{\alpha=0}^{\tau_2-l} P_c(\alpha+l)\varepsilon_{i-\delta}^{\alpha}(1-\varepsilon_{i-\delta})^{n-\alpha}\right]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!} \\
Q_2^{(x)} &= \frac{1}{P_{gb}+P_{bg}}\sum_{k=d_x}^{N} A_k^{(x)}\left[P_{bg}\epsilon_g^k(1-\epsilon_g)^{N-k} + P_{gb}\epsilon_b^k(1-\epsilon_b)^{N-k}\right] \\
&\quad + \left(\frac{N}{n}\right)! \sum_{\Omega_{i-\delta}} \sum_{\Omega_i} \Pr(\Omega_{i-\delta},\Omega_i) \sum_{k=d}^{N} \frac{A_k}{\binom{N}{k}} \sum_{j=1}^{|\pi^{(k)}|} \left[\prod_{l=0}^{n} \frac{[P_{pd}(l)]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!} - \prod_{l=0}^{n} \frac{[P_{pd0}(l)]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!}\right] \\
P_2^{(x)} &= 1 - C_2^{(x)} - Q_2^{(x)} \\
\eta &= \frac{1}{1 + P_1\left[\frac{1+P_2^{(2)}}{1-P_2^{(1)}P_2^{(2)}}\right]} \\
P_{ud} &= Q_1 + P_1\left[\frac{Q_2^{(2)} + P_2^{(2)}Q_2^{(1)}}{1-P_2^{(1)}P_2^{(2)}}\right]
\end{aligned}
$$

Chapter 5

Simulation and Analysis Results

This chapter begins by describing an error correction code with properties that make it well suited to the GH-ARQ II scheme. Then, an overview of the simulation procedure is presented. Finally, this chapter concludes by presenting the results of several experiments.

5.1 The KM Error Correction Code

This code was initially proposed and described by Krishna and Morgera [1]. Recall that the complexity of the GH-ARQ II error control scheme is strongly dependent upon the error correction code $\mathcal{C}_1^{(x)}$; if a separate decoder were required for each error correction code $\mathcal{C}_1^{(2)}, \ldots, \mathcal{C}_1^{(m)}$, the complexity of such a system would be prohibitive. Fortunately, the distinguishing property of the KM codes is that a fixed decoder configuration is capable of decoding the codes $\mathcal{C}_1^{(2)}, \ldots, \mathcal{C}_1^{(m)}$. This makes these codes ideal candidates for the GH-ARQ II error control scheme under consideration. It should be noted though that it is probably possible to find other codes with such a property; however, only KM codes will considered. Since these codes have a short block length ($n = 4, \ldots, 8$) they are well suited to soft decision decoding [3].

The error correction capabilities of the code $\mathcal{C}_1^{(x)}$, $P_c(w)$, as described in Section 4.5.1 was computed for most of the codes described in [1]. These are summarized in Appendix C. In general, KM codes are referred to as (mn, n, d), where m is the maximum depth of the code, n is the block length and d is the design minimum distance of the code. For example, the $(21, 7, 6)$ KM code encodes 7 bits into 21 bits and has a design minimum distance equal to 6. Note that this code also generates the $(14, 7, 3)$ KM code by deleting the submatrix $\boldsymbol{M}_3$ from its generator matrix $\boldsymbol{M}$.

5.2 Overview of the Simulation Procedure

A flowchart of the simulation procedure is shown in Figure 5.1. The k message bits are randomly generated with 0 and 1 being equiprobable. The message is then encoded with the cyclic[1] error detection code $g(x)$. Let $m(x)$, the message polynomial, be defined as

$$m(x) = \sum_{i=0}^{k-1} m_i x^i,$$

where m_i is the i-th bit in the message. The generator polynomial $g(x)$ of the error detection code is similarly defined as

$$g(x) = \sum_{j=0}^{N-k} g_j x^j,$$

where g_j is the j-th coefficient in the generator polynomial. The codeword resulting from the error detection coding, $v(x)$ is

$$v(x) = m(x)g(x) = \sum_{i=0}^{k-1} \sum_{j=0}^{N-k} m_i g_j x^{i+j}.$$

Note that the summation is performed over $GF(2)$ so that the coefficients of $v(x)$ are 0 or 1. The resulting codeword in the error detection code is then partitioned into n-bit blocks, whereupon each block is encoded with the appropriate submatrix $\boldsymbol{M}_x$ in the KM error correction code to form the transmitted vector $\underline{t}^{(x)}$ for the GH-ARQ II scheme.

An error sequence is then generated using the Gilbert-Elliott model and added to $\underline{v}$ and $\underline{t}^{(x)}$. The codeword from the error detection code $\underline{v}$ and the transmitted packet $\underline{t}^{(x)}$ are kept separate so that the GH-ARQ II and the simple ARQ schemes each operate under identical conditions.

Then, the decoding of the received packets is performed. The error detection decoding in the simple ARQ scheme is performed by using a polynomial division circuit. The received polynomial, $r'(x) = v(x) + e(x)$, is divided by $g(x)$. If the remainder of this division is 0, then it is assumed that no errors have occurred and the simple ARQ scheme positively acknowledges the transmission. The simulator then checks the accepted message for undetected errors. Otherwise, errors have been detected and a retransmission is requested. An undetected error is made if the error sequence $e(x)$ is a codeword in the error detection code, so that $e(x) \bmod g(x) = 0$. This

[1]In these experiments, only a cyclic error detection code is used. In principle, the only restriction required is that the code be linear.

implementation of the decoder is capable of detecting some patterns containing more errors than its error detection capability.

The decoding process in the GH-ARQ II scheme is similar to that of the simple ARQ scheme except that a provision must be made for inversion and error correction. The receiver in the GH-ARQ II scheme begins by inverting the error correction coding (if the corresponding submatrix $\boldsymbol{M}_x$ is invertible) and then attempts to detect errors with the same error detection decoder used by the simple ARQ receiver. On the second and following transmissions of a message, the receiver will attempt to correct the errors in the received packet if the inversion is unsuccessful or not possible. In this simulator, the error correction decoding is performed by selecting the codeword in the KM code which is closest to the combined block. This decoding approach is possible because n is small (which makes the number of KM codewords 2^n manageable). The output of the error correction code decoder is then checked for errors by using the error detection code. Again, this is accomplished by dividing by $g(x)$. If the syndrome is zero, the simulator checks for undetected errors, otherwise a retransmission is requested.

If the channel conditions are very poor, it is possible that a large number of transmissions would be required before an error-free packet is received. In order to safeguard against this possibility, the simulator will abandon a packet that requires more than a preset number of retransmissions. Usually, this limit is on the order of 80 retransmissions.

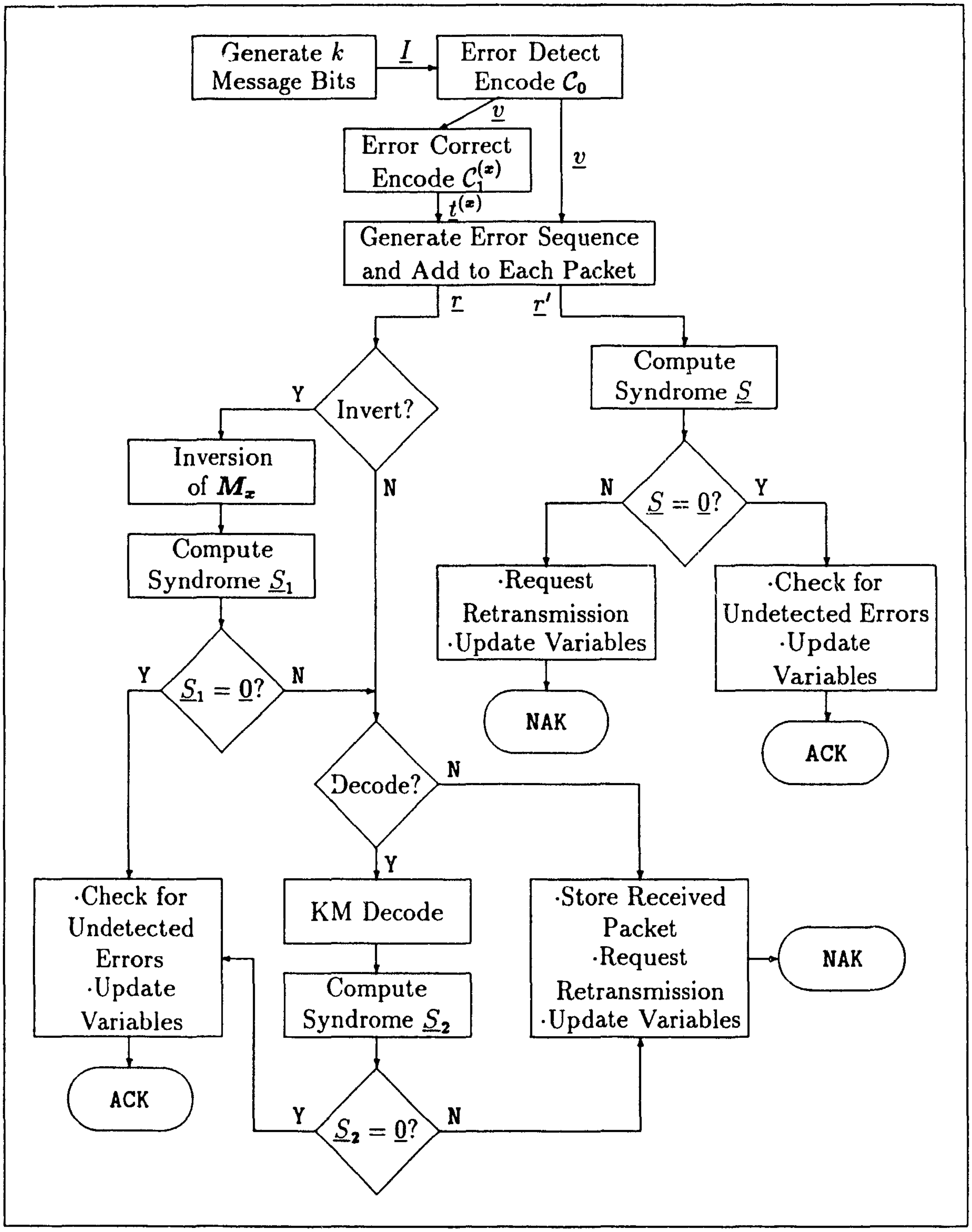

Figure 5.1: Flowchart of the Simulator

5.3 Comparison of Analysis and Simulation Results

This set of experiments estimates the throughput efficiency by simulation of several GH-ARQ II schemes. A comparison is made with the results of the previous analyses. The experiments in this section use a simple stop-and-wait scheme ($\delta = 1$) for their retransmission strategy and the simulations are all performed over 10000 packets.

Experiment A

The GH-ARQ II error control scheme in this experiment uses the BCH (127,120) code shortened by $\Delta = 47$ bits for error detection. The generator polynomial of this code is $g(x) = x^7 + x^3 + 1$. The KM (12,4,5) depth-3 code is used for error correction. The performance of this scheme was simulated and analyzed on three channels as summarized in Table 5.1.

Channel	Burst	P_{gb}	P_{bg}	P_b
A	Dense	2.04×10^{-4}	0.01	0.02
B	Diffuse	3.33×10^{-3}	0.01	0.25
C	BSC	0	0.01	1.0

Table 5.1: Channel Parameters for Experiment 5.3.A

The results of the simulations and analyses are presented in Figure 5.2. These are plotted on the range of average error rate from 10^{-7} to 10^{-2}.

The simulation and analysis results are seen to be in very close agreement. The maximum error (in absolute value) is approximately 12%. This error could be decreased by increasing the number of packets used in the simulation. The results suggest that the throughput of this error control scheme increases as the errors in the channel become progressively "bursty" in nature (as P_b decreases). Note that the results of the reliability analysis and simulations are not presented because, since the probability of undetected error tends to be quite small, its estimation by simulation is very difficult. For example, to estimate a probability of undetected error on the order of 10^{-5}, the simulation would require at least 10^6 packets, which is not feasible.

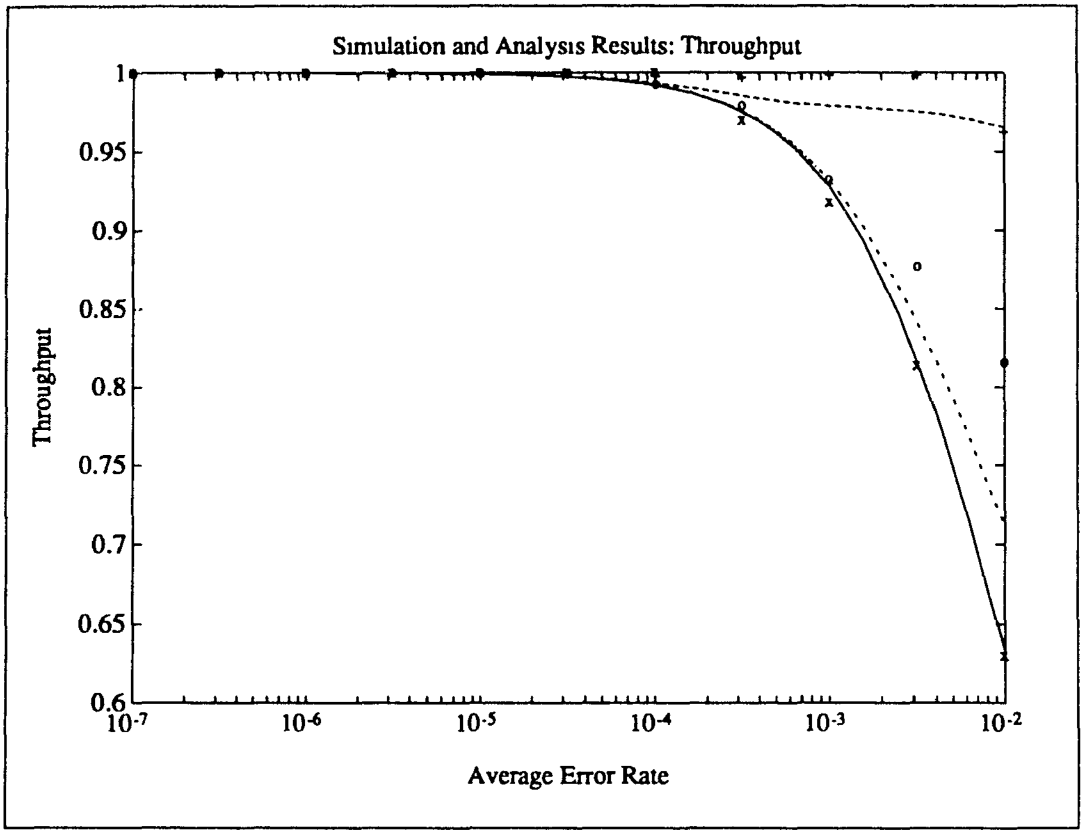

Figure 5.2: Throughput Analysis and Simulation of Experiment 5.3.A. Legend: – – – Analysis-Channel A – · – Analysis-Channel B — Analysis-Channel C + Simulation-Channel A o Simulation-Channel B x Simulation-Channel C

Experiment B

In this experiment, the GH-ARQ II error control scheme uses the (15,5,5) KM code for error correction and the BCH (127,120) code shortened by $\Delta = 47$ bits for error detection. As before, the retransmission strategy is a simple stop-and-wait scheme. The channel parameters differ from the previous experiment in that these do not satisfy the conditions described by Lugand *et al.* in [4]. Instead, the channel parameters for this experiment, summarized in Table 5.2, are similar to those described by Deng *et al.* [6] in their analysis. Note that $\rho = 1$ corresponds to the binary symmetric channel and that these parameters do not necessarily satisfy the conditions outlined by Lugand *et al.* [4].

Channel	P_{gb}	P_{bg}	$\rho = \frac{\epsilon_b}{\epsilon_g}$
A	0.1	0.05	1
B	0.1	0.05	10
C	0.1	0.05	1000

Table 5.2: Channel Parameters for Experiment 5.3.B

The results of the simulations and analyses are plotted in Figure 5.3 on the range of average error rate from 10^{-7} to 10^{-1}.

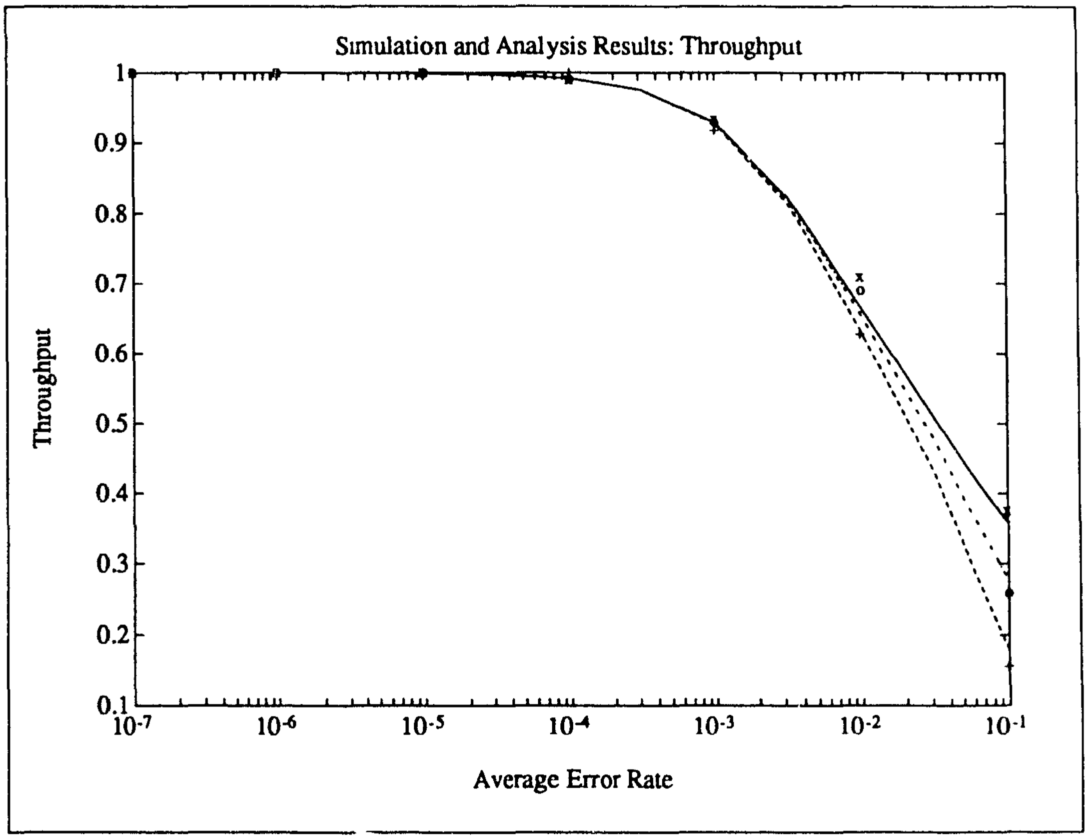

Figure 5.3: Throughput Analysis and Simulation of Experiment 5.3.B. Legend: — — — Analysis-$\rho = 1$ — · — Analysis-$\rho = 10$ — Analysis-$\rho = 1000$ + Simulation-$\rho = 1$ o Simulation-$\rho = 10$ x Simulation-$\rho = 1000$

Once again, the analysis and simulation results are seen to correspond closely; the maximum difference in absolute value being approximately 10.4%.

Experiment C

The goal of this experiment is to compare the results of the approximate throughput analysis described in Appendix A.1 with those obtained by simulation. To this end, the GH-ARQ II error control scheme uses the depth-4 (24,6,9) KM code for error correction and the BCH (127,120) code, shortened by $\Delta = 47$ bits, for error detection. The channel parameters are summarized in Table 5.3.

Channel	P_{gb}	P_{bg}	$\rho = \frac{\epsilon_b}{\epsilon_g}$
A	0.011	0.01	1
B	0.011	0.01	10
C	0.011	0.01	1000

Table 5.3: Channel Parameters for Experiment 5.3.C

The results of the analysis and simulations are plotted in Figure 5.4.

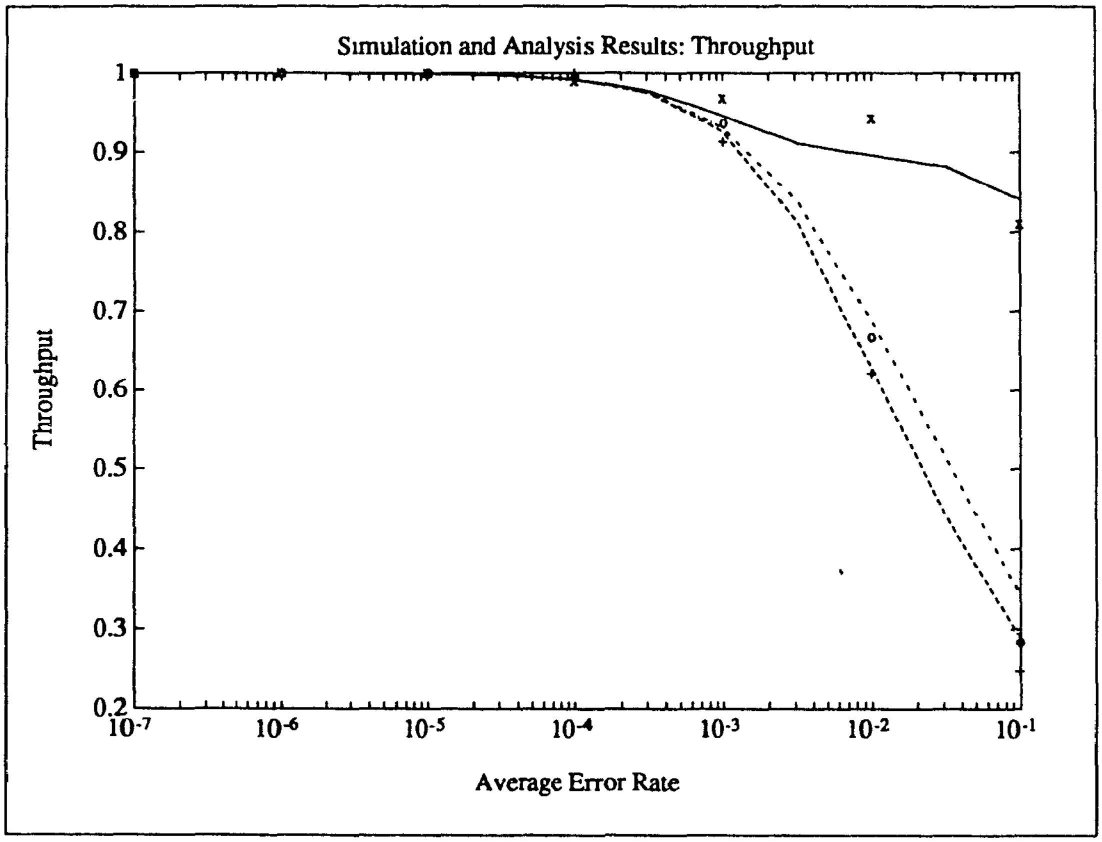

Figure 5.4: Throughput Analysis and Simulation of Experiment 5.3.C. Legend: – – – Analysis-$\rho = 1$ – · – Analysis-$\rho = 10$ — Analysis-$\rho = 1000$ + Simulation-$\rho = 1$ o Simulation-$\rho = 10$ x Simulation-$\rho = 1000$

It is seen that there is a very good correspondence between the approximate throughput analysis and the simulations. In general, the effect of the approximation upon the throughput analysis appears to be very slight. However, in the case of the reliability analysis it has been found that the between it and the simulation can be quite significant, depending upon the channel parameters.

5.4 Effect of the Roundtrip Delay

This goal of this experiment is to investigate the effect of the roundtrip delay upon the performance of the GH-ARQ II error control scheme. This is an important consideration because the roundtrip delay is an integral characteristic of the channel and retransmission strategy of real communications systems. The error detection code $\mathcal{C}_0$ is the BCH (127,120) code shortened by $\Delta = 47$ bits and the error correction code $\mathcal{C}_1$ is the (12,4,5) KM code. The performance of this error control scheme was analyzed on the four channels summarized in Table 5.4.

Channel	P_{gb}	P_{bg}	ϵ_g	ϵ_b	P_b	$\bar{\epsilon}$
A	0.0011	0.01	5×10^{-3}	0.455	0.099	0.050
B	0.01	0.01	0.025	0.075	0.5	0.050
C	0.5	0.01	10^{-5}	0.1	0.98	0.098
D	2×10^{-4}	3×10^{-4}	10^{-5}	0.1	0.4	0.040

Table 5.4: Channel Paramters for the Investigation of the Effect of δ

The throughput and reliability of this GH-ARQ II scheme were computed as a function of roundtrip delay δ. The percentage improvement for the throughput and reliability are plotted in Figures 5.5 and 5.6. In this context, the percent is relative to the case when $\delta = 1$. That is, if $\eta(\delta)$ and $P_{ud}(\delta)$ are the throughput and reliability of the GH-ARQ II scheme when the roundtrip delay is δ, then

$$\% \text{ Improvement}(\eta) = \frac{\eta(\delta) - \eta(1)}{\eta(1)} \times 100,$$

and

$$\% \text{ Improvement}(P_{ud}) = \frac{P_{ud}(1) - P_{ud}(\delta)}{P_{ud}(1)} \times 100.$$

Note that the % Improvement (P_{ud}) is defined so that an improved reliability (lower probability of undetected error) results in a positive percentage.

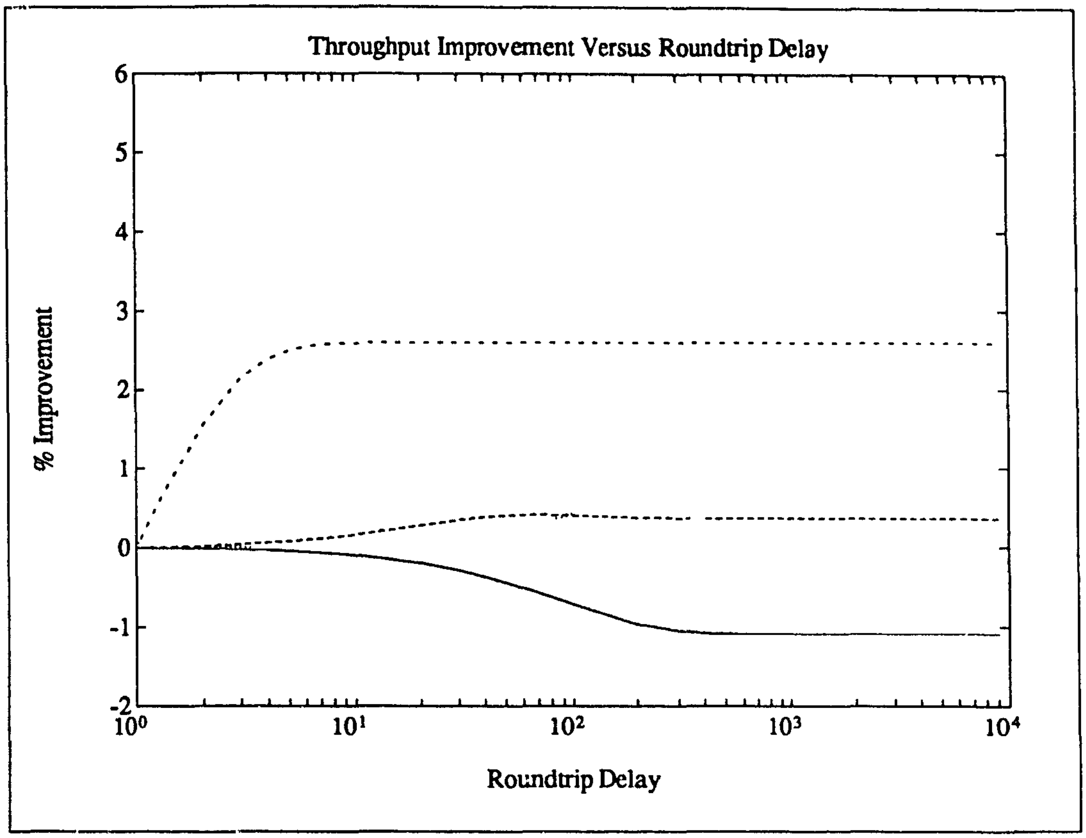

Figure 5.5: Throughput Improvement Versus Roundtrip Delay. Legend: — Channel A — — — Channel B — · — Channel C · · · Channel D

The results are interesting as they suggest that it is difficult to determine *a priori* the effect of the roundtrip delay on the performance of the GH-ARQ II scheme with a given burst-noise channel. The throughput of this error control scheme is seen to decrease monotonically on channel A, and increases monotonically on channels C and D. On channel B, the throughput increases to a maximum of improvement of 0.4% at $\delta \approx 50$ and then decreases slightly to level off at % Improvement $(\eta) = 0.37\%$. In all cases, though, the effect of the roundtrip delay is seen to be fairly small.

The effect of the roundtrip delay upon this error control scheme's reliability is more significant than it was upon its throughput. On channels B and C, the effect of the roundtrip delay is to improve the reliability by only 2 to 3% . However, on channel A, the reliability deteriorates by as much as 22% and, on channel D, improves by up to 10%.

Interestingly, the effect of the roundtrip delay upon the throughput is comparable to that upon the reliability. In particular, the performance deteriorates on channel D, improves very slightly on channels B and C, and improves moderately on channel A. The only generalization possible at this time is to say that the roundtrip delay has

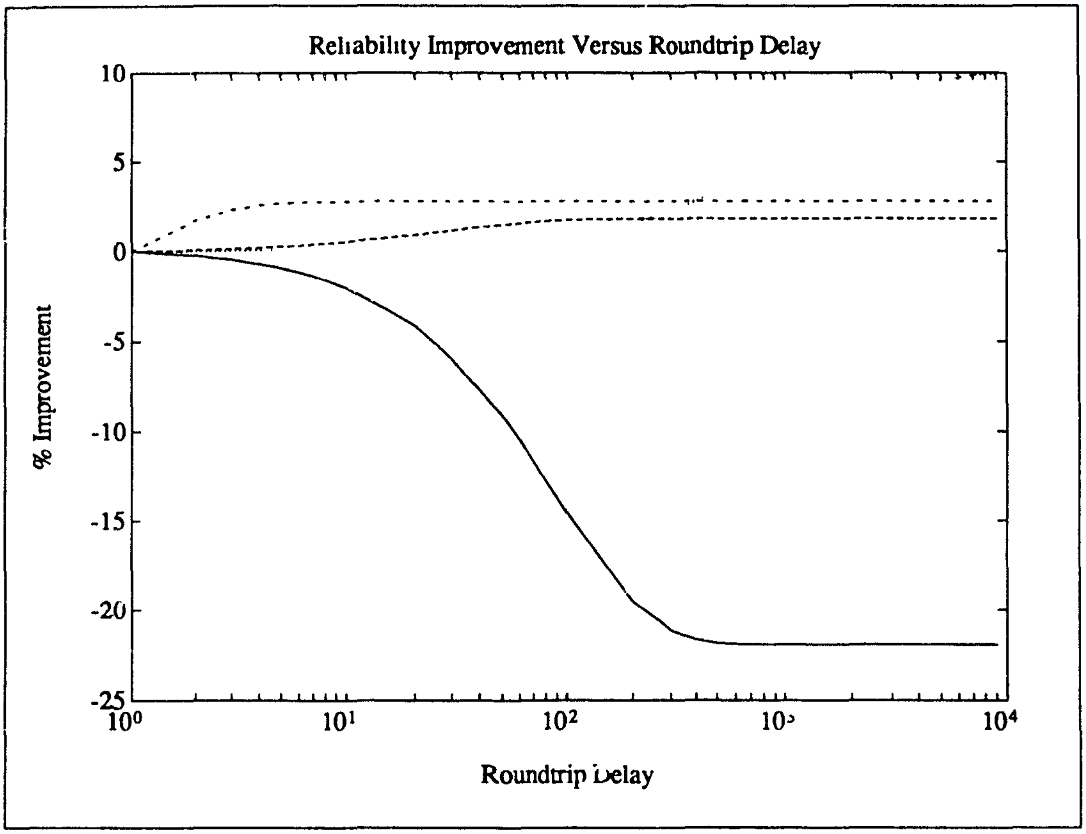

Figure 5.6: Reliability Improvement Versus Roundtrip Delay. Legend: — Channel A – – – Channel B – · – Channel C · · · Channel D

no effect upon the performance of the GH-ARQ II error control scheme when the channel is modelled as a BSC. On burst-noise channels it appears to be difficult to determine the effect of roundtrip delay on the performance of the GH-ARQ II error control scheme.

5.5 Performance Improvement Over Simple ARQ

This section explores the improvement in performance of a GH-ARQ II error control scheme over a simple ARQ scheme. Also, it establishes some conditions under which there is no significant improvement in performance of the GH-ARQ II scheme over the simple ARQ scheme.

Experiment A

The GH-ARQ II error control scheme considered in this experiment uses the BCH (127,120) code shortened by $\Delta = 47$ bits for error detection and the KM (15,5,5) depth-3 code for error correction. The roundtrip delay is $\delta = 1$ so that the retransmission strategy is a simple stop-and-wait scheme. The channel paramteres are summarized in Table 5.5.

Channel	Burst	P_{bg}	P_b
A	Dense	0.025	0.02
B	Diffuse	0.025	0.25
C	BSC	0.025	1.0

Table 5.5: Channel Parameters for Experiment 5.5.A

The throughput and reliability analyses of this experiment are plotted in Figures 5.7 and 5.8.

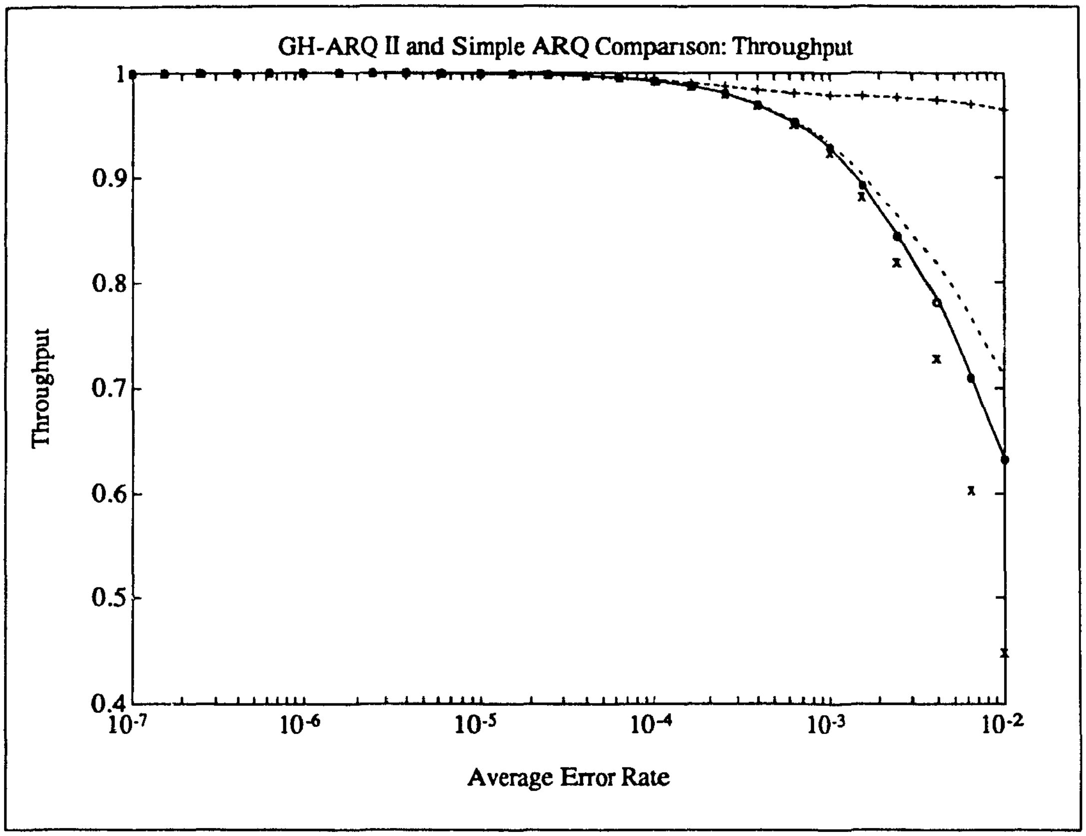

Figure 5.7: Throughput Analysis of Experiment 5.5.A. Legend: – – – GH-ARQ II-Channel A – · – GH ARQ II-Channel B — GH-ARQ II-Channel C + Simple ARQ-Channel A o Simple ARQ-Channel B x Simple ARQ-Channel C

As expected, the throughput efficiency of the GH-ARQ II error control scheme improves as the channel becomes progressively burstier, however, that of the simple ARQ scheme also increases. Further as the errors in the channel become burstier in nature, the difference in performance between the two schemes becomes negligible. This result may seem surprising, but can be understood by considering the relationship between this channel and this error control scheme. A dense burst channel is one in which the bursts are very short in duration yet cause many errors. Since the GH-ARQ II scheme has no error correction capability on the initial transmission of a message, an error burst during this transmission will force a retransmission (unless an undetected error occurs). On the second transmission of the message, the channel has either entered its "good" state (because the bursts tend to be short in duration) in which case the error correction capability is probably not required (as no errors have occurred), or the channel is still in its very "bad" state and the error correction code is incapable of correcting the numerous errors that have occurred. Note that the reliability of the simple ARQ scheme is greater than that of the GH-ARQ II scheme up to average error rate equal to 10^{-3} at which point they are approximately equal.

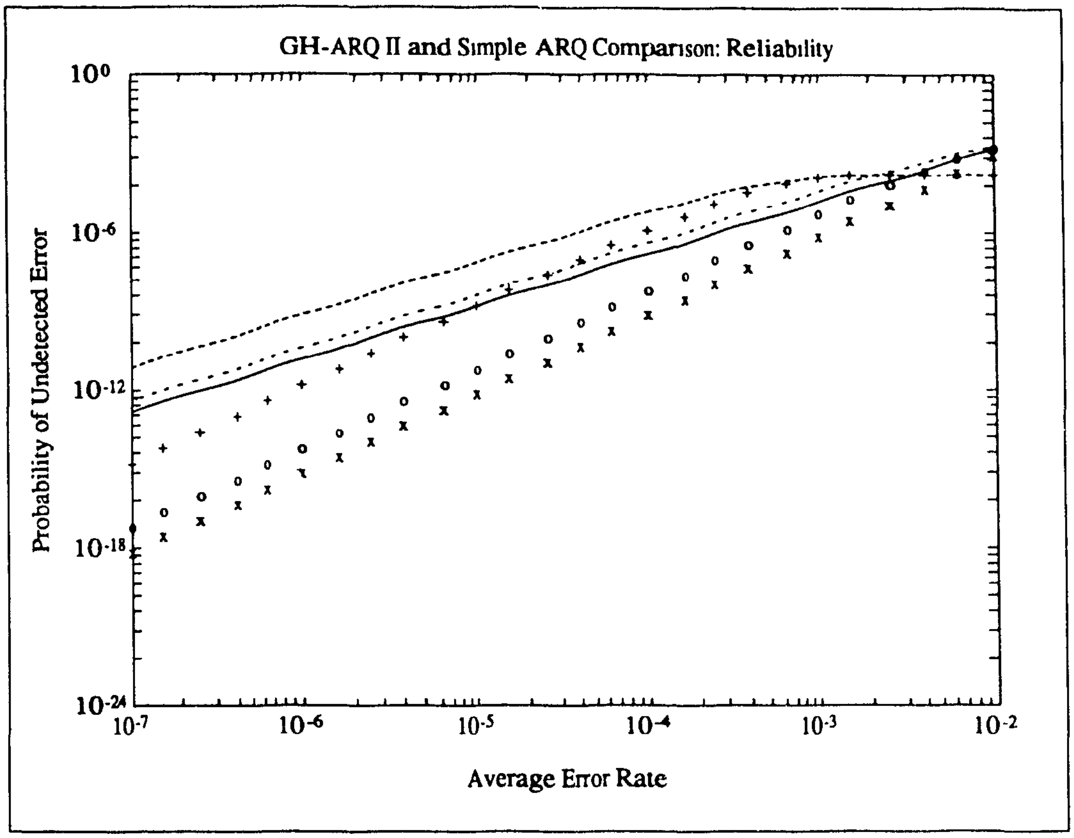

Figure 5.8: Reliability Analysis of Experiment 5.5.A. Legend: — — — GH-ARQ II-Channel A — · — GH-ARQ II-Channel B — GH-ARQ II-Channel C + Simple ARQ-Channel A o Simple ARQ-Channel B x Simple ARQ-Channel C

For such a channel, where the bursts are very short in duration and occur frequently, there is no siginificant performance advantage in selecting a GH-ARQ II error control scheme over a simple ARQ scheme.

There is, however, a significant improvement in efficiency on the other two channels when the average error rate is greater than $\bar{\epsilon} = 10^{-3}$. Below this rate, the throughput efficiency of the two schemes is virtually identical but the simple ARQ scheme is seen to be far more reliable. However, as the average error rate increases, the difference in reliability between the two schemes decreases progressively until $\bar{\epsilon} = 10^{-3}$, at which point they are comparable.

Experiment B

This experiment compares the performance of the GH-ARQ II and simple ARQ error control schemes when the channel is described in a manner similar to that in Experiment 5.3.B. Both of these error control schemes use the BCH (127,120) code shortened by $\Delta = 22$ bits for error detection. Further, the GH-ARQ II scheme uses the KM (21,7,6) code for error correction. The roundtrip delay, δ, is one and the channel parameters are summarized in Table 5.6.

Channel	P_{gb}	P_{bg}	$\rho = \frac{\epsilon_b}{\epsilon_g}$
A	5×10^{-2}	2×10^{-3}	1
B	5×10^{-2}	2×10^{-3}	10
C	5×10^{-2}	2×10^{-3}	1000

Table 5.6: Channel Parameters for Experiment 5.5.B

The throughput efficiency and reliability of these error control schemes are plotted in Figure 5.9 and 5.10.

On all three channels, the improvement in throughput efficiency of the GH-ARQ II scheme over the simple ARQ scheme is very significant. For example, when $\bar{\epsilon} = 10^{-2}$, the throughput of the GH-ARQ II scheme is approximately 80% greater than that of the simple ARQ scheme with equal reliability. This represents a tremendous improvement. Moreover, when the average error rate exceeds $\bar{\epsilon} = 10^{-2}$, the throughput and reliability of this GH-ARQ II scheme are greater than that of the simple ARQ scheme.

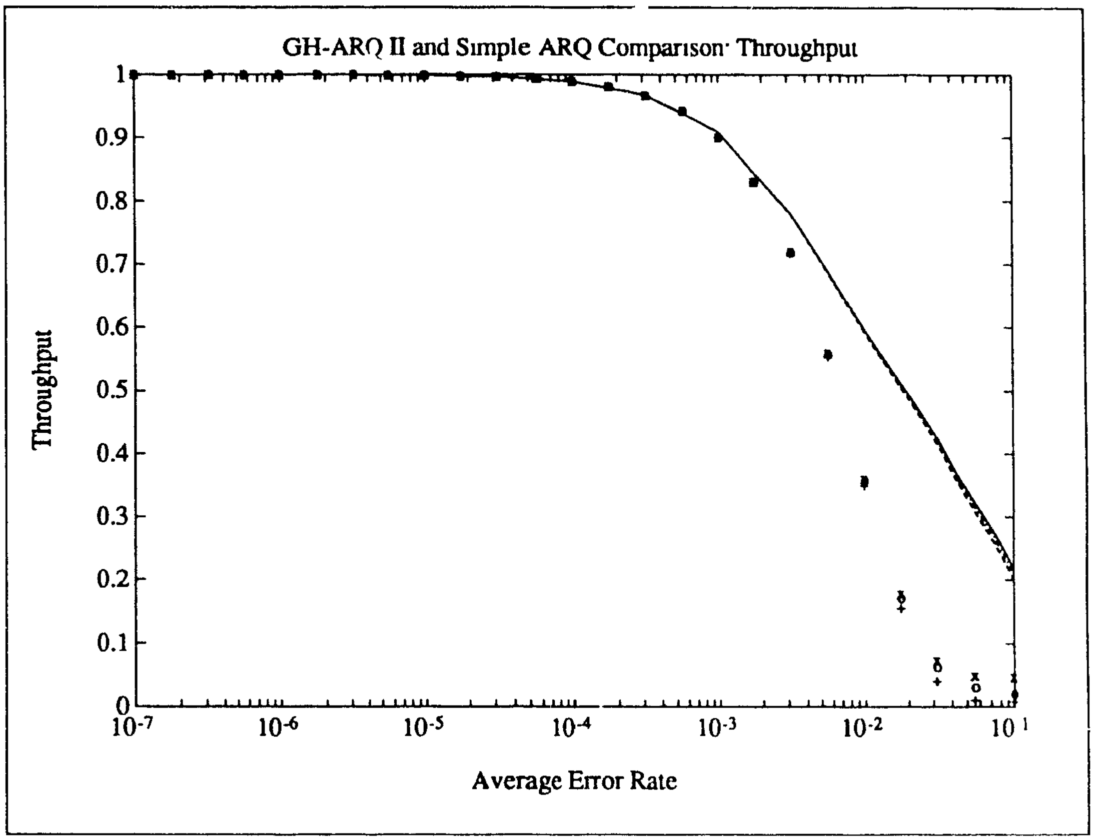

Figure 5.9: Throughput Analysis of Experiment 5.5.B. Legend: – – – GH-ARQ II-Channel A – · – GH-ARQ II-Channel B — GH-ARQ II-Channel C + Simple ARQ-Channel A o Simple ARQ-Channel B x Simple ARQ-Channel C

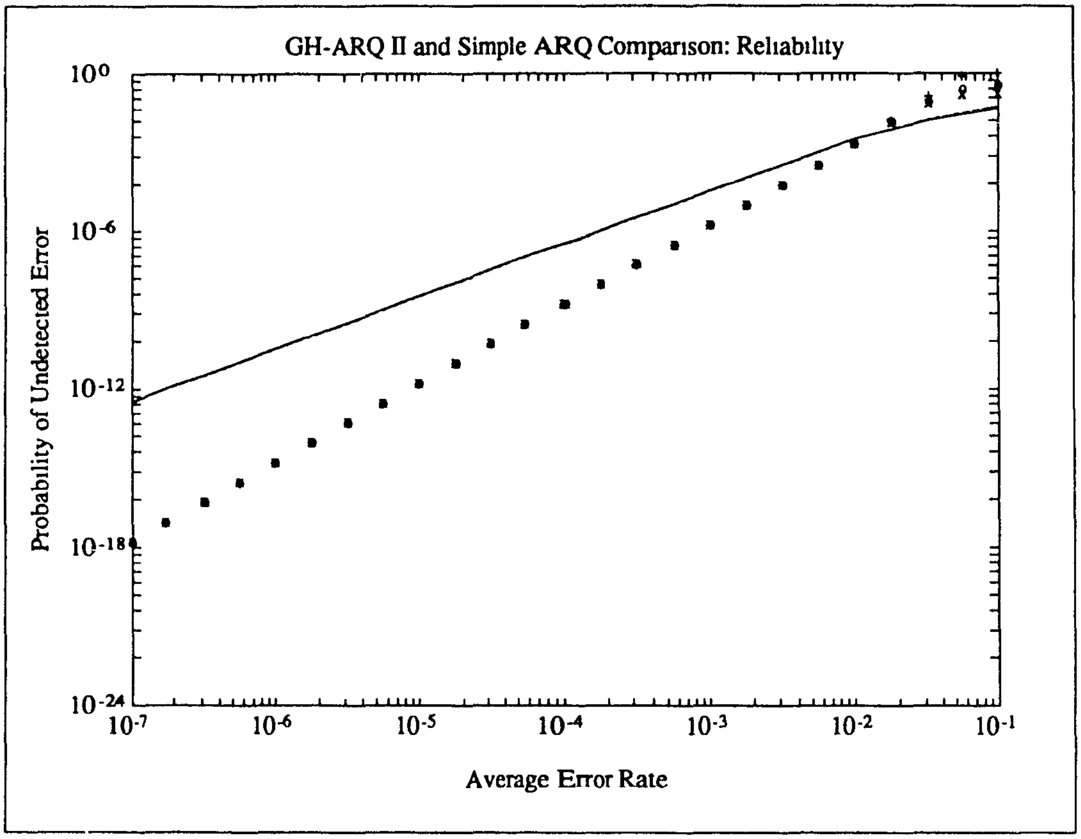

Figure 5.10: Reliability Analysis of Experiment 5.5.B. Legend: – – – GH-ARQ II-Channel A – · -- GH-ARQ II-Channel B — GH-ARQ II-Channel C + Simple ARQ-Channel A o Simple ARQ-Channel B x Simple ARQ-Channel C

5.6 Performance Of GH-ARQ II On Mobile Radio Channels

Cygan *et al.* [5] measured the characteristics of three mobile radio channels in Germany and computed the corresponding parameters of a Gilbert-Elliott model. These channel parameters are summarized in Table 5.7.

Channel	Description	P_{gb}	P_{bg}	ϵ_g	ϵ_b	$\bar{\epsilon}$
A	Interstate Munich $\bar{v} = 40$km/h	3.95×10^{-4}	1.05×10^{-4}	2.1×10^{-4}	0.317	0.2505
B	Secondary Highway Hamburg $\bar{v} = 40$km/h	2.1×10^{-4}	1.54×10^{-4}	3.4×10^{-4}	0.298	0 1721
C	Autobahn Munich–Stuttgart $\bar{v} = 90$km/h	2.96×10^{-5}	1.29×10^{-4}	1.1×10^{-4}	0.194	0.0363

Table 5.7: Gilbert-Elliott Model Parameters of Three Mobile Radio Channels

The peformance of a GH-ARQ II error control scheme was analyzed on these three channels modelled by the respective Gilbert-Elliott models The GH-ARQ II scheme in question uses a BCH (127,120) code shortened by $\Delta = 47$ bits for error detection and the KM (15,5,5) depth-3 code for error correction For comparison, the simple ARQ error control scheme also uses the BCH (127,120) code shortened by $\Delta = 47$ bits for error detection The retransmission strategy is the simple stop-and-wait scheme corresponding to $\delta = 1$. The results of this analysis are summarized in Table 5.8.

These results suggest that there is little improvement (approximately 1.5% on the average) in throughput and reliability of the GH-ARQ II error control scheme over the simple ARQ scheme. On these channels, the GH-ARQ II scheme fails to show any significant improvement in throughput because the error rate in the "bad" state

	Throughput		Reliability	
Channel	GH-ARQ II	Simple ARQ	GH-ARQ II	Simple ARQ
A	0.215439	0.212672	0.028682	0.029020
B	0.422795	0.416229	0.010682	0.010828
C	0.809086	0.807698	0.001830	0.001805

Table 5.8: Performance Analysis of GH-ARQ II and Simple ARQ Error Control Schemes on Three Mobile Radio Channels

	δ	Throughput			Reliability
Channel	(Approx)	GH-ARQ II	Simple ARQ	% Improvement	GH-ARQ II
A	2000	0.224412	0.212672	5.52	0.030830
B	4000	0.432827	0.416229	3.99	0.012406
C	$\geq 10^5$	0.811899	0.807698	0.52	0.001988

Table 5.9: Maximum Throughput of GH-ARQ II on Three Mobile Channels

of these channel models is so high that it renders the error correction code ineffective; it is unable to correct all the errors that have occurred.

A slightly greater improvement in the throughput of the GH-ARQ II scheme over the simple ARQ scheme is possible if a selective-repeat request retransmission strategy is employed instead of the stop-and-wait scheme. The throughput of the GH-ARQ II scheme was computed as a function of retransmission roundtrip delay δ. It was found that, for a given value of δ, the throughput of the GH-ARQ II scheme could be maximized on two of the three channels. On channel C, the throughput improvement appears to increase monotonically up to $\delta \approx 10000$ and then becomes constant. The maximized throughput, the corresponding (approximate) δ and reliability for each of these channels are summarized in Table 5.9

On these channels, the selective repeat request retransmission strategy has a beneficial effect upon the throughput efficiency of the GH-ARQ II error control scheme. In particular, the throughput of the GH-ARQ II scheme improves by 5.52% over the Simple ARQ scheme when $\delta = 2000$, yet only improves by 1.3% when a stop-and-wait retransmission strategy is used. This phenomenon would not be observed on binary symmetric channels.

5.7 Effect of Maximum Depth

This experiment investigates the effect of the error correction code's depth upon the throughput of a GH-ARQ II error control scheme. This scheme uses the (24,6,9) KM code for error correction and the BCH (127,120) code shortened by $\Delta = 47$ bits. The throughput efficiency of the GH-ARQ II error control scheme was analyzed in the case of the depth-4 system with the approximate expressions of Appendix A.1. For the case of $\rho = 1000$, the throughput analysis is plotted in Figures 5.11 and for that of $\rho = 1$ (BSC), it is plotted in Figure 5.12. In all the plots, the range of average error rate is from 10^{-2} to 0.25.

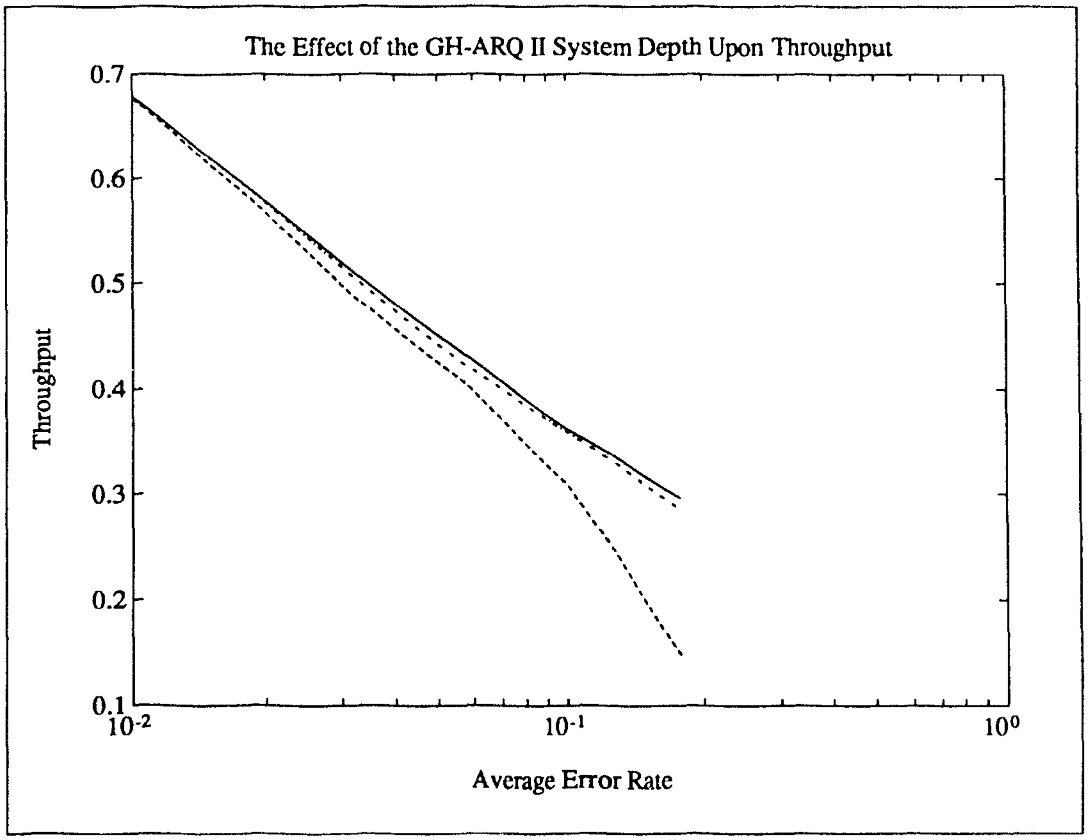

Figure 5.11: Throughput Analysis of a GH-ARQ II Scheme. $\rho = 1000$. Legend. – – – Depth-2 System – · – Depth-3 System — Depth-4 System

Regarding the case of $\rho = 1000$, the improvement in throughput efficiency of the depth-3 and depth-4 systems over the depth-2 system is significant at average error rates greater than 5×10^{-2}. The difference in throughput between the depth-3 and depth-4 systems is insignificant over the range under consideration. When the channel is a BSC ($\rho = 1$) the improvement in throughput becomes significant at average error rates greater than 2×10^{-2}. But, as the average error rate continues to increase, the

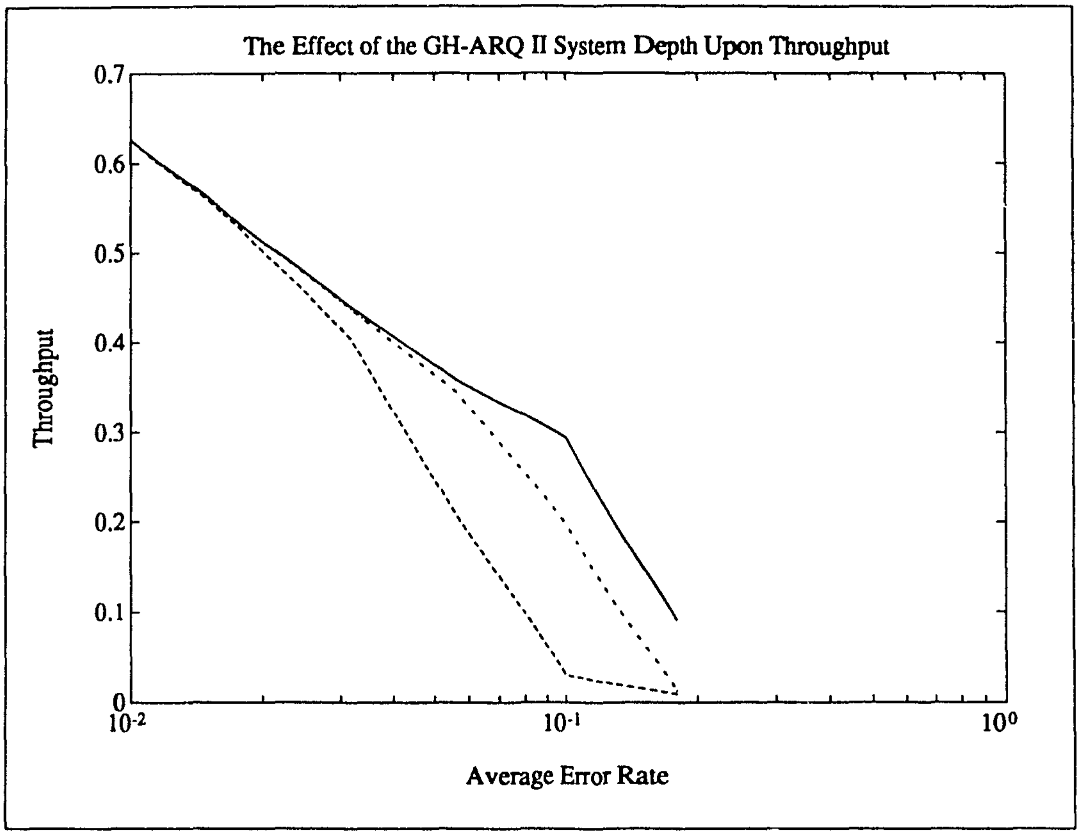

Figure 5.12: Throughput Analysis of a GH-ARQ II Scheme. $\rho = 1$. Legend: – – – Depth-2 System – · – Depth-3 System — Depth- 4 System

improvement in throughput of the depth-4 system over the depth-3 becomes much more significant than it was in the case when $\rho = 1000$. Interestingly, when the average rate approaches 0.125, the throughput of the depth-3 system approaches that of the depth-2 system. This is because the error correction codes in these systems are equally incapable of correcting the errors caused by the channel.

It was found that, in general, the depth-2 system exhibits a lower reliability than the other two systems and that the difference is significant, once again, at average error rates above 5×10^{-2}.

Chapter 6

Summary

This chapter describes some other interesting topics related to the work in this thesis and concludes by presenting a general summary and some conclusions.

6.1 Other Interesting Topics

The analysis of the GH-ARQ II and simple ARQ error control schemes assumed that the receiver buffer was infinite; in practice this is not achievable. The next logical step in this analysis would be to extend it to cases where the receiver buffer is finite and to study its effect upon the system's throughput efficiency and reliability.

The performance of GH-ARQ II error control schemes using soft-decision decoding has been studied when the channel is modelled by a Binary Symmetric Channel. It would be interesting to extend this analysis to analog fading channel models so that the effect of multipath and frequency selective fading upon the performance of GH-ARQ II schemes in mobile communications systems could be studied.

As Section 3.2 demonstrated, the error correction code can have a significant effect upon the performance of the error detection code. This effect merits a closer investigation. As previously mentionned, this problem has been considered in the context of a concatenated coding scheme; it should now be considered in the context of the GH-ARQ II error control scheme. Such an investigation should determine the conditions, if any, for which the error correction encoding improves the performance of the error detection code. Then, if it is found that the error correction encoding cannot improve the performance of the error detection code, the conditions for which the former has no effect upon the latter should be determied.

The experiments in Chapter 5 demonstrated that the retransmission delay can have a moderate effect upon the performance of GH-ARQ II error control schemes on burst-noise channels. It might be worthwhile to determine *a priori* the effect of δ as a function of the channel model's parameters.

Finally, KM codes were chosen for error correction in the GH-ARQ II scheme. It would be a worthwhile endeavour to search for other classes of codes with the property that their decoder implementation is fixed, yet may have greater minimum distance o. a lesser effect upon error detection codes of interest.

6.2 Conclusions

A general approach to analyze the performance of GH-ARQ II error control schemes on channels modelled by first-order Markov chains is presented. This approach is easily extended to depth-3 and depth-4 systems and is best suited to schemes which use a short block code for error correction. Several issues related to this performance analysis were presented: the effect of the error correction encoding upon the error detection code, the integer partitioning problem, the computation of the weight distribution of various codes and the analysis of a simple ARQ scheme on a Gilbert-Elliott burst-noise channel model.

The approach taken to analyze the GH-ARQ II error control scheme begins by constructing the receiver's state transition diagram. The throughput and reliability of this scheme may then be expressed in terms of the transition probabilities in this diagram. The problem then becomes the computation of these transition probabilities. The complexity of this computation may be decreased by grouping together those error sequences with "similar" properties. The integer partitioning problem was found to be useful in this computation of the transition probabilities.

The results of several experiments were presented. It was found that the analysis and simulation results were very consistent and that the approximation for the throughput computation is very good. Further, the roundtrip delay associated with the various retransmission strategies can have a beneficial or detrimental effect upon the performance of the GH-ARQ II scheme. It is difficult to determine, *a priori*, the exact effect of the roundtrip delay. Also, it was found that the throughput efficiency and reliability of the GH-ARQ II scheme improve as the errors in the channel become progressively bursty in nature. However, it was also observed that this phenomenon is equally true for simple ARQ systems and consequently the difference in performance between the two types of schemes appears to decrease as the channel becomes burstier. In general, it appears that GH-ARQ II schemes may be better suited to slowly varying channel conditions than they are to quickly varying conditions. When the channel is fairly diffuse and the average error rate is high, the high-order GH-ARQ II systems were found to offer significantly improved performance over second order (depth-2) systems. Further, under these conditions, the GH-ARQ II systems were found to yield significantly higher throughput efficiencies than simple ARQ schemes while maintaining comparable reliability.

Bibliography

[1] H. Krishna and S. D. Morgera, "A New Error Control Scheme for Hybrid ARQ Systems," *IEEE Transactions on Communications,* Vol. COM-35, pp. 981-990, October 1987.

[2] P. S. Yu and S. Lin, "An Efficient Selective-Repeat ARQ Scheme for Satellite Channels and Its Throughput Analysis," *IEEE Transactions on Communications,* Vol. COM-29, pp. 353-363, March 1981.

[3] S. D. Morgera and V. K. Oduol , "Soft-Decision Decoding Applied to the Generalized Type-II Hybrid ARQ Scheme," *IEEE Transactions on Communications,* Vol. COM-37, pp. 393-396, April 1989.

[4] 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 Transactions on Communications,* Vol. COM-37, pp. 755-765, July 1989.

[5] D. Cygan, M. Dippold and J. Finkenzeller, "Models for the Land Mobile Satellite Channel," *Archiv Für Elektronik Und Übertragungstechnik,* Vol. 42, pp. 329-338, November/December 1988.

[6] R. H. Deng and D. J. Costello, Jr., "Reliability and Throughput Analysis of a Concatenated Coding Scheme," *IEEE Transactions on Communications,* Vol. COM-35, pp. 698-705, July 1987.

[7] G. Ungerboeck, "Trellis-Coded Modulation with Redundant Signal Sets. Part 1: Introduction," *IEEE Communications Magazine,* Vol. 25, pp. 5-11, February 1987.

[8] F. J. MacWilliams and N. J. A. Sloane, *Theory of Error-Correcting Codes,* North-Holland, Amsterdam, The Netherlands, 1977.

[9] K. Hwang and F. A. Briggs, *Computer Architecture and Parallel Processing,* McGraw-Hill, New York 1984.

[10] E. N. Gilbert, "Capacity of a Burst-Noise Channel," *The Bell System Technical Journal,* Vol. 39, pp. 1253-1265, September 1960.

[11] E. O. Elliott, "Estimates of Error Rates for Codes on Burst- Noise Channels," *The Bell System Technical Journal,* Vol. 42, pp. 1977- 1997, September 1963.

[12] S. D. Morgera and H. Krishna, *Digital Signal Processing: Applications to Communications and Algebraic Coding Theories,* Academic Press, London 1989.

[13] L. N. Kanal and A. R. K. Sastry, "Models for Channels with Memory and Their Applications to Error Control," *Proceedings of the IEEE,* Vol. 66, pp. 724-744, July 1978.

[14] D. M. Mandelbaum, "An Adaptive-Feedback Coding Scheme Using Incremental Redundancy," *IEEE Transactions on Information Theory,* Vol. IT-20, pp. 388-389, May 1974.

[15] K. Brayer and S. Natarajan, "An Investigation of ARQ and Hybrid FEC-ARQ on an Experimental High Latitude Meteor Burst Channel," *IEEE Transactions on Communicaitons,* Vol. COM-37, pp. 1239-1242, November 1989.

[16] T. Kløve and M. Miller, "The Detection of Errors After Error-Correction Decoding," *IEEE Transactions on Communications,* Vol. COM-32, pp. 511-517, May 1984.

[17] E. Y. Rocher and R. L. Pickholtz, "An Analysis of the Effectiveness of Hybrid Transmission Schemes," *IBM Journal of Research and Development,* Vol. 14, pp. 426-433, July 1970.

[18] E. R. Berlekamp, R. E. Peile and S. P. Pope, "The Application of Error Control to Communications," *IEEE Communications Magazine,* Vol. 25, pp. 44-57, April 1987.

[19] H. M. Taylor and S. Karlin, *An Introduction to Stochastic Modelling,* Academic Press, London 1984.

[20] Y. Wang and S. Lin, "A Modified Selective-Repeat Type-II Hybrid ARQ System and Its Performance Analysis," *IEEE Transactions on Communications,* Vol. COM-31, pp. 593-607, May 1983.

[21] S. Lin and P. S. Yu, "A Hybrid ARQ Scheme with Parity Retransmission for Error Control of Satellite Channels," *IEEE Transactions on Communications,* Vol. COM-30, pp. 1701-1719, July 1982.

[22] J. G. Proakis and D. G. Manolakis, *Introduction to Digital Signal Processing,* Macmillan, New York 1988.

[23] I. P. Goulden and D. M. Jackson, *Combinatorial Enumeration,* John Wiley & Sons, New York 1983.

[24] S. Lin, D. J. Costello, Jr., and M. J. Miller, "Automatic-Repeat-Request Error-Control Schemes," *IEEE Communications Magazine,* Vol. 22, pp. 5-17, December 1984.

[25] Z. McC. Huntoon and A. M. Michelson, "On the Computation of the Probability of Post-Decoding Error Events for Block Codes," *IEEE Transactions on Information Theory,* Vol. IT-23, pp. 399-403, May 1977.

[26] T. Kasami, T. Kløve and S. Lin, "Linear Block Codes for Error Detection," *IEEE Transactions on Information Theory,* Vol. IT-29, pp. 131-137, January 1983.

[27] R. E. Blahut, *Theory and Practice of Error Control Codes,* Addison-Wesley, Reading, Massachusetts, 1983.

[28] S. K. Leung-Yan-Cheong, E. R. Barnes and D. U. Friedman, "On Some Properties of the Undetected Error Probability of Linear Codes," *IEEE Transactions on Information Theory,* Vol. IT-25, pp. 110-112, January 1979.

[29] R. V. Churchill and J. W. Brown, *Complex Variables and Applications,* McGraw-Hill, New York 1984.

[30] R. J. McEliece and W. E. Stark, "Channels with Block Interference," *IEEE Transactions on Information Theory,* Vol. IT-30, pp. 44-53, January 1984.

[31] R. A. Comroe and D. J. Costello, Jr., "ARQ Schemes for Data Transmission in Mobile Radio Systems," *IEEE Journal on Selected Areas in Communications,* Vol. 2, pp. 472-481, July 1984.

[32] H. O. Burton and D. D. Sullivan, "Errors and Error Control," *Proceedings of the IEEE,* Vol. 60, pp. 1293-1301, November 1972.

[33] J. K. Wolf, A. M. Michelson and A. H. Levesque, "On the Probability of Undetected Error for Linear Block Codes," *IEEE Transactions on Communications,* Vol. COM-30, pp. 317-324, February 1982.

[34] I. M. Jacobs, "Practical Applications of Coding," *IEEE Transactions on Information Theory,* Vol. IT-20, pp. 305-310, May 1974.

[35] P. J. Mabey, "Mobile Radio Data Transmission – Coding for Error Control," *IEEE Transactions on Vehicular Technology,* Vol. VT-27, pp. 99-109, August 1978.

[36] G. D. Forney, "Coding and its Application in Space Communications," *IEEE Spectrum,* Vol. 7, pp. 47-58, June 1970.

[37] T. Kasami, T. Fujiwara and S. Lin, "A Concatenated Coding Scheme for Error Control," *IEEE Transactions on Communications,* Vol. COM-34, pp 481-488, May 1986.

[38] A. I. Drukarev and K. P. Yiu, "Performance of Error-Correcting Codes on Channels with Memory," *IEEE Transactions on Communications,* Vol. COM-34, June 1986.

[39] A. Drukarev and D. J. Costello, Jr., "A Comparison of Block and Convolutional Codes in ARQ Error Control Schemes," *IEEE Transactions on Communications,* Vol. COM-30, pp. 2449-2455, November 1982.

[40] J. G. Proakis, *Digital Communications,* McGraw-Hill, New York, 1983.

[41] L. Lugand and D. J. Costello, Jr., "A Comparison of Three Hybrid ARQ Schemes Using Convolutional Codes on a Non-stationary Channel," *Conf. Rec. GLOBECOM '82*, Miami Florida, Nov. 29–Dec. 2, 1982, pp. C.8.4.1-C8.4.5.

[42] S. Tsai, "Markov Characterization of the HF Channel," *IEEE Transactions on Communication Technology,* Vol. COM-17, pp. 24-32, February 1969.

[43] J. J. Metzner and D. Chang, "Efficient Selective Repeat ARQ Strategies for Very Noisy and Fluctuating Channels," *IEEE Transactions on Communications,* Vol. COM-33, pp. 409-416, May 1985.

[44] A. R. K. Sastry, "Performance of Hybrid Error Control Schemes on Satellite Channels," *IEEE Transactions on Communications,* Vol. COM-23, pp. 689-694, July 1975.

[45] E. J. Weldon, "An Improved Selective-Repeat ARQ Strategy," *IEEE Transactions on Communications,* Vol. COM-30, pp. 480-486, March 1982.

[46] D. Drajić and B. Vučetić, "Evaluation of Hybrid Error Control Systems," *IEE Proceedings,* Vol. 131F, pp. 183-193, April 1984.

[47] C. Fujiwara, M. Kasahara, K. Yamashita and T. Namekawa, "Evaluations of Error Control Techniques in Both Independent-Error and Dependent-Error Channels," *IEEE Transactions on Communications,* Vol. COM-26, pp. 785-794, June 1978.

[48] J. J. Metzner, "Improvements in Block-Retransmission Schemes," *IEEE Transactions on Communications,* Vol. COM-27, pp. 524-532, February 1979.

[49] P. S. Sindhu, "Retransmission Error Control with Memory," *IEEE Transactions on Communications,* Vol. COM-25, pp. 473-479, May 1977.

[50] M. J. Miller and S. Lin, "The Analysis of Some Selective-Repeat ARQ Schemes with Finite Receiver Buffer," *IEEE Transactions on Communications,* Vol. COM-29, pp. 1307-1315, September 1981.

[51] A. M. Michelson and A. H. Levesque, *Error-Control Techniques for Digital Communications,* John Wiley & Sons, New York, 1985.

[52] J. MacWilliams, "A Theorem on the Distribution of Weights in a Systematic Code," *Bell System Technical Journal,* Vol. 42, pp. 79-95, January 1963.

[53] R. E. Blahut, "Algebraic Fields, Signal Processing, and Error Control," *Proceedings of the IEEE,* Vol. 73, pp. 874-893, May 1985.

[54] J. P. Stenbit, "Table of Generators for Bose-Chaudhuri Code," *IEEE Transactions on Information Theory,* Vol. IT-10, pp. 390-391, October 1964.

[55] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications,* Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[56] R. W. Lucky, J. Salz, E. J. Weldon, *Principles of Data Communication,* McGraw-Hill, New York 1968.

[57] C. E. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical Journal,* Vol. 27, pp. 379-423, July 1948.

[58] D. Haccoun, J. Conan and G. Golly, "Node to Node Protocols on a High Speed Full-Duplex Satellite Link," *IEEE National Telecommunications Conference,* pp. 28.1.1-28.1.4, December 1978.

[59] S. Kallel and D. Haccoun, "Sequential Decoding with ARQ and Code Combining: A Robust Hybrid FEC/ARQ System," *IEEE Transactions on Communications,* Vol. COM-36, pp. 773-780, July 1988.

[60] R. J. Benice and A. H. Frey, "Comparisons of Error Control Techniques," *IEEE Transactions on Communications Technology,* Vol. COM-12, pp. 146-154, December 1964.

[61] R. J. Benice and A. H. Frey, "An Analysis of Retransmission Schemes," *IEEE Transactions on Communications Technology,* Vol. COM-12, pp. 135-145, December 1964.

[62] J. M. Morris, "Optimal Blocklengths for ARQ Error Control Schemes," *IEEE Transactions on Communications,* Vol. COM-27, pp. 488-493, February 1979.

[63] C. S. K. Leung and A. Lam, "Forward Error Correction for an ARQ Scheme," *IEEE Transactions on Communications,* Vol. COM-29, pp. 1514-1519, October 1981.

[64] P. F. Turney, "An Improved Stop-and-Wait ARQ Logic for Data Transmission in Mobile Radio Systems," *IEEE Transactions on Communications,* Vol. COM-29, pp. 68-71, January 1981.

[65] B. Arazi, "Improving the Throughput of an ARQ Stop and Wait Scheme for Burst Noise Channels," *IEEE Transactions on Communications,* Vol. COM-24, pp. 661-663, June 1976.

[66] A. R. K. Sastry and L. N. Kanal, "Hybrid Error Control Using Retransmission and Generalized Burst-Trapping Codes," *IEEE Transactions on Communications,* Vol. COM-24, pp. 385-393, April 1976.

[67] D. Chase, "Code Combining – A Maximum-Likelihood Decoding Approach for Combining an Arbitrary Number of Noisy Packets," *IEEE Transactions on Communications,* Vol. COM-33, pp. 385-393, May 1985.

[68] B. D. Fritchman, "A Binary Channel Characterization Using Partitioned Markov Chains," *IEEE Transactions on Information Theory,* Vol. IT-13, pp. 221-227, April 1967.

[69] L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone, *Classification and Regression Trees,* Wadsworth Statistics/Probability Series, Belmont, California, 1984.

[70] A. Tucker, *Applied Combinatorics,* John Wiley & Sons, New York 1980.

[71] M. Hall, Jr., *Combinatorial Theory,* John Wiley & Sons, New York 1986.

[72] S. Karlin and H. M. Taylor, *A First Course in Stochastic Processes,* Academic Press, London 1975.

Appendix A

Analysis of Other GH-ARQ II Systems

This appendix describes an approximation that can significantly decrease the complexity of the throughput analysis of GH-ARQ II schemes. Then the analysis of systems using non-invertible submatrices $\boldsymbol{M}_x$, $x = 2, \ldots, m$ is presented. This appendix concludes by extending the analysis of Chapter 4 to GH-ARQ II Depth-3 and Depth-4 systems.

A.1 Approximate Throughput Analysis

Suppose that it is only required to estimate the throughput efficiency of a GH-ARQ II scheme; the estimate of its reliability being less important. In this case, it is possible to significantly decrease the computational load of the analysis by approximating the post-decoding probability of undetected error to be zero. This is not an unreasonable assumption since it is usually the case that the post-inversion probability of undetected error is greater (and often much greater) than the post-decoding probability of undetected error. That is,

$$\Pr(\underline{S}_1 = \underline{0}, \text{PI Errors}) > \Pr(\underline{S}_1 \neq \underline{0}, \underline{S}_2 = \underline{0}, \text{PD Errors})$$

is usually true.

By approximating $\Pr(\underline{S}_1 \neq \underline{0}, \underline{S}_2 = \underline{0}, \text{PD Errors}) = 0$, the complexity of the analysis is decreased. The expressions for the transition probability $Q_2^{(x)}$ is thus

$$Q_2^{(x)} = \frac{1}{P_{gb} + P_{bg}} \sum_{k=d_x}^{N} A_k^{(x)} \left[P_{bg}\epsilon_g^k(1-\epsilon_g)^{N-k} + P_{gb}\epsilon_b^k(1-\epsilon_b)^{N-k}\right]$$

and all other transition probabilities may be computed as described in Section 4.7. Experimental results regarding the consistency between the simulation and the approximate throughput computation were presented in Experiment 5.3.C.

A.2 Noninvertible Generator Matrices M_x

If a given submatrix $\boldsymbol{M}_x$ of the error correction code $C_1^{(2)}$ is not invertible, the analysis of the GH-ARQ II will require a few simple modifications. Since the matrix $\boldsymbol{M}_x$ is not invertible, the transition probabilities $C_2^{(x)}$, $Q_2^{(x)}$ and $P_2^{(x)}$ may be defined as follows:

$$\begin{aligned} C_2^{(x)} &= \Pr(\underline{S}_1 \neq \underline{0}, \underline{S}_2 = \underline{0}, \text{No PD Errors}) \\ Q_2^{(x)} &= \Pr(\underline{S}_1 \neq \underline{0}, \underline{S}_2 = \underline{0}, \text{PD Errors}) \\ P_2^{(x)} &= \Pr(\underline{S}_1 \neq \underline{0}, \underline{S}_2 \neq \underline{0}) \end{aligned}$$

Since the error sequence $\underline{e}_i$ is any N-bit sequence, then

$$B_k^{(x)} = \binom{N}{k}, \text{ for } 0 \leq k \leq N \text{ and } x = 1, 2.$$

The expressions for these transition probabilities are simply

$$\begin{aligned} C_2^{(x)} &= \left(\frac{N}{n}\right)! \sum_{k=1}^{N\tau_2/n} \sum_{j=1}^{|\pi^{(k)}|} \sum_{\Omega_i} \varepsilon_i^k (1-\varepsilon_i)^{N-k} \\ &\quad \cdot \sum_{\Omega_{i-\delta}} \Pr(\Omega_{i-\delta}, \Omega_i) \prod_{l=0}^{\tau_2} \frac{\left[\binom{n}{l} \sum_{\alpha=0}^{\tau_2-l} P_c(\alpha+l) \varepsilon_{i-\delta}^{\alpha} (1-\varepsilon_{i-\delta})^{n-\alpha}\right]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!} \\ Q_2^{(x)} &= \left(\frac{N}{n}\right)! \sum_{\Omega_{i-\delta}} \sum_{\Omega_i} \Pr(\Omega_{i-\delta}, \Omega_i) \sum_{k=d}^{N} \frac{A_k}{\binom{N}{k}} \sum_{j=1}^{|\pi^{(k)}|} \left[\prod_{l=0}^{n} \frac{[P_{pd}(l)]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!} - \prod_{l=0}^{n} \frac{[P_{pd0}(l)]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!}\right] \\ P_2^{(x)} &= 1 - C_2^{(x)} - Q_2^{(x)} \end{aligned}$$

A.3 Depth-3 GH-ARQ II Systems

This depth-3 system uses up to a rate-1/3 code $C_1^{(3)}$ for error correction. The generator matrix of this code may be expressed as

$$\boldsymbol{M} = [\boldsymbol{M}_1 \mid \boldsymbol{M}_2 \mid \boldsymbol{M}_3]$$

and the generator matrix of a rate-1/2 error correction code $C_1^{(2)}$ may be obtained from $\boldsymbol{M}$ by deleting the submatrix $\boldsymbol{M}_3$ from it.

A.3.1 The Receiver State Transition Diagram

The receiver state transition diagram of a depth-3 GH-ARQ II system is shown in Figure A.1. Similarly to the definition of the states in Section 4.2, states **T3**, **T2** and **T1** correspond to the transmitter encoding the packet $(\underline{I}, \underline{Q})$ with $\boldsymbol{M}_3$, $\boldsymbol{M}_2$ and $\boldsymbol{M}_1$ respectively of the error correction code with the receiver performing error correction decoding with $C_1^{(3)}$ (if necessary).

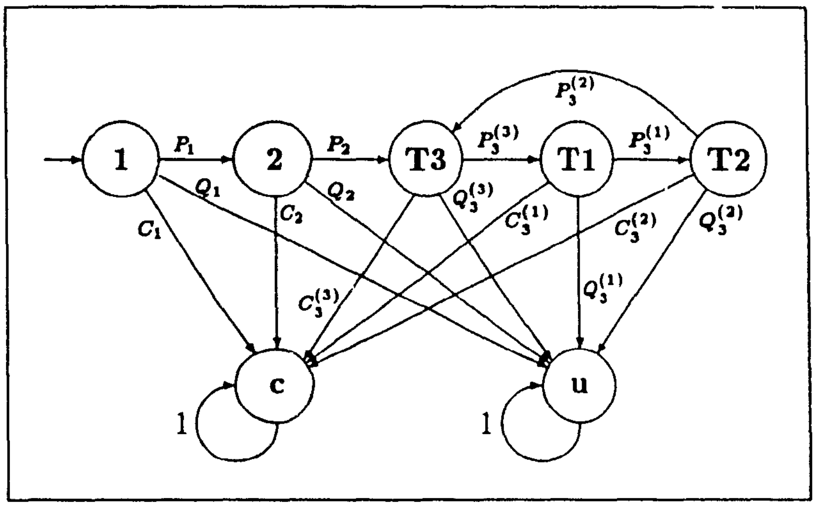

Figure A.1: Depth 3 System. Receiver State Transition Diagram

A.3.2 Its Throughput and Reliability

Using a procedure similar to the one described in Section 4.3, the throughput and reliability of the depth-3 system are easily computed. The throughput of this system was found to be

$$\eta = \frac{1}{1 + P_1 + P_1 P_2 \left[\frac{1+P_3^{(3)}+P_3^{(1)}P_3^{(3)}}{1-P_3^{(1)}P_3^{(2)}P_3^{(3)}}\right]},$$

and its reliability is

$$P_{ud} = Q_1 + P_1 Q_2 + P_1 P_2 \left[\frac{Q_3^{(3)} + P_3^{(3)}Q_3^{(1)} + P_3^{(1)}P_3^{(3)}Q_3^{(2)}}{1 - P_3^{(1)}P_3^{(2)}P_3^{(3)}}\right].$$

A.3.3 Transition Probabilities

This section presents the expressions for the transition probabilities. Most of these, except those related to the error correction decoding of the code $C_1^{(3)}$, are straightforward extensions of the transition probabilities computed in Chapter 4.

As before, $C_3^{(x)}$, $Q_3^{(x)}$ and $P_3^{(x)}$ may be expressed as

$$\begin{aligned} C_3^{(x)} &= \Pr(\underline{S}_1 = \underline{0}, \text{No PI Errors}) + \Pr(\underline{S}_1 \neq \underline{0}, \underline{S}_2 = \underline{0}, \text{No PD Errors}) \\ Q_3^{(x)} &= \Pr(\underline{S}_1 = \underline{0}, \text{PI Errors}) + \Pr(\underline{S}_1 \neq \underline{0}, \underline{S}_2 = \underline{0}, \text{PD Errors}) \\ P_3^{(x)} &= \Pr(\underline{S}_1 \neq \underline{0}, \underline{S}_2 \neq \underline{0}) \end{aligned}$$

The computation of $C_3^{(x)}$ is similar to that of $C_2^{(x)}$ except that the code $C_1^{(3)}$ is used to attempt to correct the errors that may have occurred during transmission. Let

$$X_3 = \Pr(\underline{S}_1 \neq \underline{0}, \underline{S}_2 = \underline{0}, \text{No PD Errors}),$$

so

$$C_3^{(x)} = \frac{1}{P_{gb} + P_{bg}} \left[P_{bg}(1 - \varepsilon_g)^N + P_{gb}(1 - \epsilon_b)^N \right] + X_3.$$

The problem, once again, is to compute X_3. Similarly to the analysis of the depth-2 system,

$$X_3 = \sum_{k=1}^{N\tau_3/n} B_k^{(x)} \sum_{j=1}^{|\pi^{(k)}|} \sum_{\Omega_i} \sum_{\Omega_{i-1}} \sum_{\Omega_{i-2}} \Pr(\underline{e} \text{ is correctable}, \underline{s} \sim \pi_j^{(k)}, \Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i),$$

where $\Omega_{i-2\delta}$ is the state of the channel 2δ transitions in the past and, as before, $\pi^{(k)}$ is the set of potentially correctable partitions of the integer k. To complete the computation of X_3, it is required to compute

$$\begin{aligned} &\Pr(\underline{e} \text{ is correctable}, \underline{e} \sim \pi_j^{(k)}, \Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i) \\ &= \Pr(\underline{e} \text{ is correctable} \mid \underline{e} \sim \pi_j^{(k)}, \Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i) \\ &\quad \cdot \Pr(\underline{e} \sim \pi_j^{(k)} \mid \Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i) \Pr(\Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i) \end{aligned} \tag{A.1}$$

As in Section 4.5.1,

$$\begin{aligned} &\Pr(\underline{e} \sim \pi_j^{(k)} \mid \Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i) \\ &= \Pr(\underline{e} \sim \pi_j^{(k)} \mid \Omega_i) \\ &= \frac{\left(\frac{N}{n}\right)!}{\binom{N}{k}} \prod_{l=0}^{\tau_3} \frac{\left[\binom{n}{l}\right]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!} \varepsilon_i^k (1 - \varepsilon_i)^{N-k}. \end{aligned} \tag{A.2}$$

There remains the computation of $\Pr(\underline{e} \text{ is correctable} \mid \underline{s} \sim \pi_j^{(k)}, \Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i)$:

$$\begin{aligned} &\Pr(\underline{e} \text{ is correctable} \mid \underline{s} \sim \pi_j^{(k)}, \Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i) \\ &= \Pr(\underline{e} \text{ is correctable} \mid \underline{e} \sim \pi_j^{(k)}, \Omega_{i-2\delta}, \Omega_{i-\delta}) \end{aligned}$$

Let p be the number of errors in the two previous blocks $\mathcal{B}^{(i-2\delta)}$ and $\mathcal{B}^{(i-\delta)}$ of the combined blocks $[\mathcal{B}^{(i-2\delta)} \mid \mathcal{B}^{(i-\delta)} \mid \mathcal{B}^{(i)}]$. Then,

$$\begin{aligned}
&\Pr(\underline{e} \text{ is correctable} \mid \underline{e} \sim \pi_j^{(k)}, \Omega_{i-2\delta}, \Omega_{i-\delta}) \\
&= \prod_{l=0}^{\tau_3} [\Pr(p \leq \tau_3 - l)]^{\pi_{jl}^{(k)}} \\
&= \prod_{l=0}^{\tau_3} \left[\sum_{\alpha=0}^{\tau_3 - l} P_c(\alpha + l) \Pr(p = \alpha) \right]^{\pi_{jl}^{(k)}} \\
&= \prod_{l=0}^{\tau_3} \left[\sum_{\alpha=0}^{\tau_3 - l} P_c(\alpha + l) \sum_{\beta=0}^{\alpha} \binom{n}{\beta} \binom{n}{\alpha-\beta} \epsilon_{i-2\delta}^{\beta} (1 - \epsilon_{i-2\delta})^{n-\beta} \epsilon_{i-\delta}^{\alpha-\beta} (1 - \epsilon_{i-\delta})^{n-\alpha+\beta} \right]^{\pi_{jl}^{(k)}} \quad \text{(A.3)}
\end{aligned}$$

By substituting Equations A.2 and A.3 into Equation A.1, the computation of X_3 is complete. The transition probability $C_3^{(x)}$ is thus

$$\begin{aligned}
C_3^{(x)} &= \frac{1}{P_{gb} + P_{bg}} \left[P_{bg}(1 - \epsilon_g)^N + P_{gb}(1 - \epsilon_b)^N \right] \\
&+ \left(\frac{N}{n}\right)! \sum_{k=1}^{N\tau_3/n} \frac{B_k^{(x)}}{\binom{N}{k}} \sum_{j=1}^{|\pi^{(k)}|} \sum_{\Omega_i} \epsilon_i^k (1 - \epsilon_i)^{N-k} \sum_{\Omega_{i-\delta}} \sum_{\Omega_{i-2\delta}} \Pr(\Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i) \\
&\cdot \prod_{l=0}^{\tau_3} \frac{\left[\binom{n}{l} \sum_{\alpha=0}^{\tau_3 - l} P_c(\alpha + l) \sum_{\beta=0}^{\alpha} \binom{n}{\beta} \binom{n}{\alpha-\beta} \epsilon_{i-2\delta}^{\beta} (1 - \epsilon_{i-2\delta})^{n-\beta} \epsilon_{i-\delta}^{\alpha-\beta} (1 - \epsilon_{i-\delta})^{n-\alpha+\beta} \right]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!}
\end{aligned}$$

The joint transition probability $\Pr(\Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i)$ is computed in Appendix B.3.

The computation, or rather approximation, of $Q_3^{(x)}$ is also similar to that of $Q_2^{(x)}$ in Chapter 4, with some modifications though. As before, let

$$Y_3 = \Pr(\underline{S}_1 \neq \underline{0}, \underline{S}_2 = \underline{0}, \text{PD Errors}).$$

Therefore,

$$Q_3^{(x)} = \frac{1}{P_{gb} + P_{bg}} \sum_{k=d_x}^{N} A_k^{(x)} \left[P_{bg} \epsilon_g^k (1 - \epsilon_g)^{N-k} + P_{gb} \epsilon_b^k (1 - \epsilon_b)^{N-k} \right] + Y_3.$$

Recall from Section 4.5.2 that

$$\begin{aligned}
Y_3' &= \sum_{j=1}^{2^N - 2^k} \Pr(\underline{\epsilon}_{pd}^{(j)} \in \tilde{\mathcal{C}}_0) \\
&= \sum_{k=d}^{N} \frac{A_k}{\binom{N}{k}} \Pr(k \text{ PD Errors}),
\end{aligned}$$

where A_k and d are the weight distribution and minimum distance of the unaltered error detection code respectively, and the Pr(k PD Errors) refers to the probability of k errors at the output of the error correction code $\mathcal{C}_1^{(3)}$ decoder. This probability may be expressed in terms of the current and previous channel states as

$$\Pr(k \text{ PD Errors}) = \sum_{\Omega_{i-2\delta}} \sum_{\Omega_{i-\delta}} \sum_{\Omega_i} \Pr(\Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i) \sum_{j=1}^{|\pi^{(k)}|} \mathcal{N}(\pi_j^{(k)}) \Pr(\pi_j^{(k)}),$$

where $\mathcal{N}(\pi_j^{(k)})$ is the number of decoded error sequences with post-decoding errors partitioned like $\pi_j^{(k)}$ and $\Pr(\pi_j^{(k)})$ is the probability of this partition. The quantity $\mathcal{N}(\pi_j^{(k)})$ is as computed in Section 4.5.2. To compute $\Pr(\pi_j^{(k)})$ Table 4.1 has to be slightly modified to account for $\Omega_{i-2\delta}$. This is shown in Table A.1 which is constructed like Table 4.1, except that it accounts for $\Omega_{i-2\delta}$ by adding the column labelled $w_{i-2\delta}$.

$w_{i-2\delta}$	$w_{i-\delta}$	w_i	w_o	$\mathcal{N}(w_{i-2\delta}, w_{i-\delta}, w_i, w_o)$
0	0	0	0	$\mathcal{N}(0,0,0,0)$
0	0	0	1	$\mathcal{N}(0,0,0,1)$
⋮	⋮	⋮	⋮	⋮
0	0	0	n	$\mathcal{N}(0,0,0,n)$
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
n	n	n	0	$\mathcal{N}(n,n,n,0)$
n	n	n	1	$\mathcal{N}(n,n,n,1)$
⋮	⋮	⋮	⋮	⋮
n	n	n	n	$\mathcal{N}(n,n,n,n)$

Table A.1: Error Correction Code $\mathcal{C}_1^{(3)}$ Decoder Output

Using Table A.1, it is a simple matter to compute $P_{pd}(l)$, the probability of l errors in a block $\mathcal{B}^{(o)}$ at the output of the error correction code $\mathcal{C}_1^{(3)}$ decoder. Then,

$$\begin{aligned} P_{pd}(l) &= \sum_{\Omega_{i-2\delta}} \sum_{\Omega_{i-\delta}} \sum_{\Omega_i} \Pr(\Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i) \\ &\quad \cdot \sum_{j=0}^{n} \sum_{k=0}^{n} \sum_{m=0}^{n} \frac{\mathcal{N}(j,k,m,l)}{\binom{n}{j}\binom{n}{k}\binom{n}{m}} \Pr(w_{i-\delta} = j, w_{i-\delta} = k, w_i = m) \\ &= \sum_{\Omega_{i-2\delta}} \sum_{\Omega_{i-\delta}} \sum_{\Omega_i} \Pr(\Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i) \\ &\quad \cdot \sum_{j=0}^{n} \sum_{k=0}^{n} \sum_{m=0}^{n} \frac{\mathcal{N}(j,k,m,l)}{\binom{n}{j}\binom{n}{k}\binom{n}{m}} \Pr(w_{i-2\delta} = j) \Pr(w_{i-\delta} = k) \Pr(w_i = m) \end{aligned}$$

$$P_{pd}(l) = \sum_{\Omega_{i-2\delta}} \sum_{\Omega_{i-\delta}} \sum_{\Omega_i} \Pr(\Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i)$$
$$\cdot \sum_{j=0}^{n} \sum_{k=0}^{n} \sum_{m=0}^{n} \mathcal{N}(j,k,m,l) \varepsilon_{i-2\delta}^{j} (1-\varepsilon_{i-2\delta})^{n-j} \varepsilon_{i-\delta}^{k} (1-\varepsilon_{i-\delta})^{n-k} \varepsilon_{i}^{m} (1-\varepsilon_{i})^{n-m}$$

The Pr(k PD Errors) is then

$$\Pr(k \text{ PD Errors}) = \left(\frac{N}{n}\right)! \sum_{\Omega_{i-2\delta}} \sum_{\Omega_{i-\delta}} \sum_{\Omega_i} \Pr(\Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i) \sum_{j=1}^{|\pi^{(k)}|} \prod_{l=0}^{n} \frac{[P_{pd}(l)]^{\pi_{jl}^{(k)}}}{\pi_{j.}^{(k)}!},$$

and

$$Y' = \left(\frac{N}{n}\right)! \sum_{\Omega_{i-2\delta}} \sum_{\Omega_{i-\delta}} \sum_{\Omega_i} \Pr(\Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i) \sum_{k=d}^{N} \frac{A_k}{\binom{N}{k}} \sum_{j=1}^{|\pi^{(k)}|} \prod_{l=0}^{n} \frac{[P_{pd}(l)]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!}.$$

The $\Pr(\underline{\epsilon}_{pd}^{(0)} \in \tilde{\mathcal{C}}_0)$ is

$$\Pr(\underline{\epsilon}_{pd}^{(0)} \in \tilde{\mathcal{C}}_0) = \sum_{k=d}^{N} \frac{A_k}{\binom{N}{k}} \Pr(k \text{ PD Errors}, \underline{e} = \underline{0}),$$

where $\underline{e}$ is the currently received error sequence and

$$\Pr(k \text{ PD Errors}, \underline{e} = \underline{0}) = \sum_{\Omega_{i-2\delta}} \sum_{\Omega_{i-\delta}} \sum_{\Omega_i} \Pr(\Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i) \sum_{j=1}^{|\pi^{(k)}|} \mathcal{N}(\pi_j^{(k)}) \Pr(\pi_j^{(k)})$$
$$\mathcal{N}(\pi_j^{(k)}) = \frac{\left(\frac{N}{n}\right)!}{\prod_{l=0}^{n} \pi_{jl}^{(k)}!}$$
$$\Pr(\pi_j^{(k)}) = \prod_{l=0}^{n} [P_{pd0}(l)]^{\pi_{jl}^{(k)}}$$

In this case, $P_{pd0}(l)$ is the probability that the block $\mathcal{B}^{(o)}$ at the output of the error correction decoder of the code $\mathcal{C}_1^{(3)}$ contains l errors when the currently received error sequence is equal to $\underline{0}$. Again, this quantity is easily computed by considering in Table A.1 only those rows with $w_i = 0$. Then, the result is

$$P_{pd0}(l) = \sum_{j=0}^{n} \sum_{k=0}^{n} \mathcal{N}(j,k,0,l) \varepsilon_{i-2\delta}^{j} (1-\varepsilon_{i-2\delta})^{n-j} \varepsilon_{i-\delta}^{k} (1-\varepsilon_{i-\delta})^{n-k}.$$

Therefore,

$$\Pr(\underline{\epsilon}_{pd}^{(0)} \in \tilde{\mathcal{C}}_0) = \left(\frac{N}{n}\right)! \sum_{\Omega_{i-2\delta}} \sum_{\Omega_{i-\delta}} \sum_{\Omega_i} \Pr(\Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i) \sum_{k=d}^{N} \frac{A_k}{\binom{N}{k}} \sum_{j=1}^{|\pi^{(k)}|} \prod_{l=0}^{n} \frac{[P_{pd0}(l)]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!}$$

and Y may be approximated as

$$\begin{aligned} Y_3 \approx & \left(\frac{N}{n}\right)! \sum_{\Omega_{i-2\delta}} \sum_{\Omega_{i-\delta}} \sum_{\Omega_i} \Pr(\Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i) \\ & \cdot \sum_{k=d}^{N} \frac{A_k}{\binom{N}{k}} \sum_{j=1}^{|\pi^{(k)}|} \left[\prod_{l=0}^{n} \frac{[P_{pd}(l)]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!} - \prod_{l=0}^{n} \frac{[P_{pd0}(l)]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!} \right]. \end{aligned}$$

Finally,

$$\begin{aligned} Q_3^{(x)} = & \frac{1}{P_{gb}+P_{bg}} \sum_{k=d_x}^{N} A_k^{(x)} \left[P_{bg}\epsilon_g^k (1-\epsilon_g)^{N-k} + P_{gb}\epsilon_b^k (1-\epsilon_b)^{N-k} \right] \\ & + \left(\frac{N}{n}\right)! \sum_{\Omega_{i-2\delta}} \sum_{\Omega_{i-\delta}} \sum_{\Omega_i} \Pr(\Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i) \\ & \cdot \sum_{k=d}^{N} \frac{A_k}{\binom{N}{k}} \sum_{j=1}^{|\pi^{(k)}|} \left[\prod_{l=0}^{n} \frac{[P_{pd}(l)]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!} - \prod_{l=0}^{n} \frac{[P_{pd0}(l)]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!} \right] \end{aligned}$$

A.3.4 Summary

$$\begin{aligned} C_1 &= \frac{1}{P_{gb}+P_{bg}} \left[P_{bg}(1-\epsilon_g)^N + P_{gb}(1-\epsilon_b)^N \right] \\ Q_1 &= \frac{1}{P_{gb}+P_{bg}} \sum_{k=d_1}^{N} A_k^{(1)} \left[P_{bg}\epsilon_g^k (1-\epsilon_g)^{N-k} + P_{gb}\epsilon_b^k (1-\epsilon_b)^{N-k} \right] \\ P_1 &= 1 - Q_1 - C_1 \\ C_2 &= \frac{1}{P_{gb}+P_{bg}} \left[P_{bg}(1-\epsilon_g)^N + P_{gb}(1-\epsilon_b)^N \right] \\ & \quad + \left(\frac{N}{n}\right)! \sum_{k=1}^{N\tau_2/n} \frac{B_k^{(2)}}{\binom{N}{k}} \sum_{j=1}^{|\pi^{(k)}|} \sum_{\Omega_i} \varepsilon_i^k (1-\varepsilon_i)^{N-k} \\ & \quad \cdot \sum_{\Omega_{i-\delta}} \Pr(\Omega_{i-\delta}, \Omega_i) \prod_{l=0}^{\tau_2} \frac{\left[\binom{n}{l} \sum_{\alpha=0}^{\tau_2-l} P_c(\alpha+l) \varepsilon_{i-\delta}^{\alpha} (1-\varepsilon_{i-\delta})^{n-\alpha} \right]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!} \\ Q_2 &= \frac{1}{P_{gb}+P_{bg}} \sum_{k=d_2}^{N} A_k^{(2)} \left[P_{bg}\epsilon_g^k (1-\epsilon_g)^{N-k} + P_{gb}\epsilon_b^k (1-\epsilon_b)^{N-k} \right] \\ & \quad + \left(\frac{N}{n}\right)! \sum_{\Omega_{i-\delta}} \sum_{\Omega_i} \Pr(\Omega_{i-\delta}, \Omega_i) \sum_{k=d}^{N} \frac{A_k}{\binom{N}{k}} \sum_{j=1}^{|\pi^{(k)}|} \left[\prod_{l=0}^{n} \frac{[P_{pd}(l)]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!} - \prod_{l=0}^{n} \frac{[P_{pd0}(l)]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!} \right] \\ P_2 &= 1 - C_2 - Q_2 \end{aligned}$$

$$
\begin{aligned}
C_3^{(x)} &= \frac{1}{P_{gb}+P_{bg}}\left[P_{bg}(1-\epsilon_g)^N + P_{gb}(1-\epsilon_b)^N\right] \\
&+ \left(\frac{N}{n}\right)! \sum_{k=1}^{N\tau_3/n} \frac{B_k^{(x)}}{\binom{N}{k}} \sum_{j=1}^{|\pi^{(k)}|} \sum_{\Omega_i} \epsilon_i^k (1-\epsilon_i)^{N-k} \sum_{\Omega_{i-\delta}} \sum_{\Omega_{i-2\delta}} \Pr(\Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i) \\
&\cdot \prod_{l=0}^{\tau_3} \frac{\left[\binom{n}{l}\sum_{\alpha=0}^{\tau_3-l} P_c(\alpha+l)\sum_{\beta=0}^{\alpha}\binom{n}{\beta}\binom{n}{\alpha-\beta}\epsilon_{i-2\delta}^{\beta}(1-\epsilon_{i-2\delta})^{n-\beta}\epsilon_{i-\delta}^{\alpha-\beta}(1-\epsilon_{i-\delta})^{n-\alpha+\beta}\right]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!} \\
Q_3^{(x)} &= \frac{1}{P_{gb}+P_{bg}} \sum_{k=d_x}^{N} A_k^{(x)} \left[P_{bg}\epsilon_g^k(1-\epsilon_g)^{N-k} + P_{gb}\epsilon_b^k(1-\epsilon_b)^{N-k}\right] \\
&+ \left(\frac{N}{n}\right)! \sum_{\Omega_{i-2\delta}} \sum_{\Omega_{i-\delta}} \sum_{\Omega_i} \Pr(\Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i) \sum_{k=d}^{N} \frac{A_k}{\binom{N}{k}} \sum_{j=1}^{|\pi^{(k)}|} \left[\prod_{l=0}^{n} \frac{[P_{pd}(l)]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!} - \prod_{l=0}^{n} \frac{[P_{pd0}(l)]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!}\right] \\
P_3^{(x)} &= 1 - C_3^{(x)} - Q_3^{(x)} \\
\eta &= \frac{1}{1 + P_1 + P_1P_2\left[\frac{1+P_3^{(3)}+P_3^{(1)}P_3^{(3)}}{1-P_3^{(1)}P_3^{(2)}P_3^{(3)}}\right]} \\
P_{ud} &= Q_1 + P_1Q_2 + P_1P_2\left[\frac{Q_3^{(3)} + P_3^{(3)}Q_3^{(1)} + P_3^{(1)}P_3^{(3)}Q_3^{(2)}}{1 - P_3^{(1)}P_3^{(2)}P_3^{(3)}}\right]
\end{aligned}
$$

A.4 Depth-4 GH-ARQ II Systems

This section briefly describes the performance analysis of a depth-4 GH-ARQ II error control scheme. Since its analysis is very similar to that of the depth-2 and depth-3 systems, only the pertinent details are presented.

A.4.1 The Receiver State Transition Diagram

Figure A.2 illustrates the receiver state transition diagram of a depth-4 GH-ARQ II error control scheme.

A.4.2 Its Throughput and Reliability

Again, using a procedure like the one described in Section 4.3, the throughput and reliability of the depth-4 system are easily computed. The throughput of the depth-4

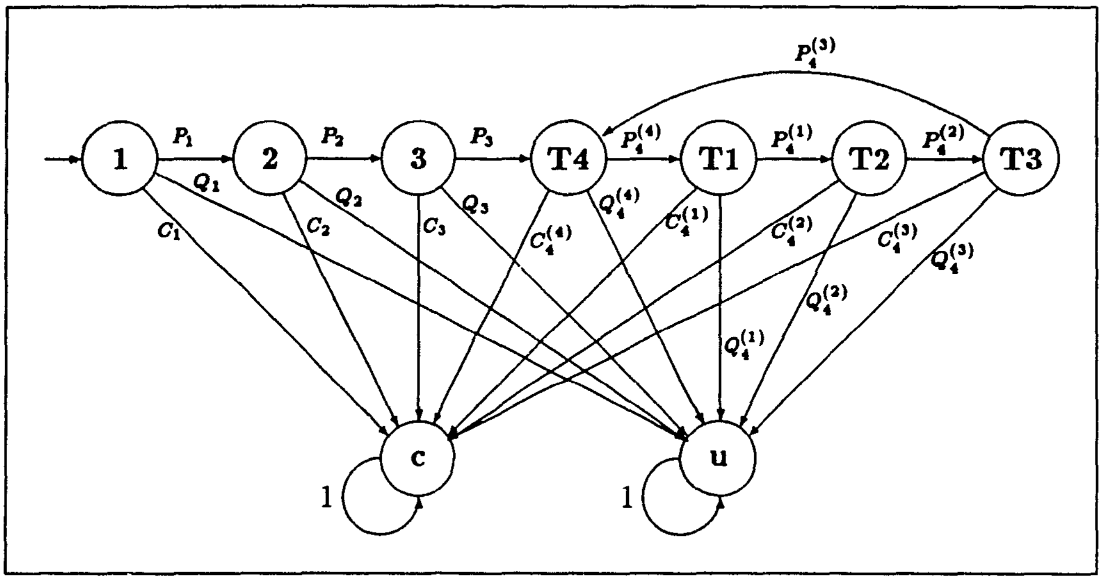

Figure A.2: Depth 4 System. Receiver State Transition Diagram

system was found to be

$$\eta = \frac{1}{1 + P_1 + P_1P_2 + P_1P_2P_3\left[\frac{1+P_4^{(4)}+P_4^{(1)}P_4^{(4)}+P_4^{(1)}P_4^{(2)}P_4^{(4)}}{1-P_4^{(1)}P_4^{(2)}P_4^{(3)}P_4^{(4)}}\right]},$$

and its reliability is

$$\begin{aligned} P_{ud} &= Q_1 + P_1Q_2 + P_1P_2Q_3 \\ &\quad + P_1P_2P_3\left[\frac{Q_4^{(3)}P_4^{(1)}P_4^{(2)}P_4^{(4)} + Q_4^{(4)} + Q_4^{(2)}P_4^{(1)}P_4^{(4)} + Q_4^{(1)}P_4^{(4)}}{1 - P_4^{(1)}P_4^{(2)}P_4^{(3)}P_4^{(4)}}\right]. \end{aligned}$$

A.4.3 Summary of Transition Probabilities

The transition probabilities for C_1, Q_1, P_1, C_2, Q_2 and P_2 are exactly as described in the summary in Section A.3.4. The transition probabilities C_3, Q_3 and P_3 may also computed from the $C_3^{(x)}$, $Q_3^{(x)}$ and $P_3^{(x)}$ as described in Section A.3.4 by setting $x = 3$. The other transition probabilities, namely $C_4^{(x)}$, $Q_4^{(x)}$ and $P_4^{(x)}$ are summarized below.

Let $\xi(\epsilon, q) = \binom{n}{q} \epsilon^q (1-\epsilon)^{n-q}$. Then,

$$
\begin{aligned}
C_4^{(x)} &= \frac{1}{P_{gb}+P_{bg}}\left[P_{bg}(1-\epsilon_g)^N + P_{gb}(1-\epsilon_b)^N\right] \\
&+ \left(\frac{N}{n}\right)! \sum_{k=1}^{N\tau_4/n} \frac{B_k^{(x)}}{\binom{N}{k}} \sum_{j=1}^{|\pi^{(k)}|} \sum_{\Omega_i} \epsilon_i^k (1-\epsilon_i)^{N-k} \sum_{\Omega_{i-\delta}} \sum_{\Omega_{i-2\delta}} \sum_{\Omega_{i-3\delta}} \Pr(\Omega_{i-3\delta}, \Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i) \\
&\cdot \prod_{l=0}^{\tau_4} \frac{\left[\binom{n}{l} \sum_{\alpha=0}^{\tau_4-l} P_c(\alpha+l) \sum_{\beta=0}^{\alpha} \sum_{\gamma=0}^{\alpha-\beta} \xi(\epsilon_{i-\delta},\beta)\xi(\epsilon_{i-2\delta},\gamma)\xi(\epsilon_{i-3\delta},\alpha-\beta-\gamma)\right]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!} \\
Q_4^{(x)} &= \frac{1}{P_{gb}+P_{bg}} \sum_{k=d_x}^{N} A_k^{(x)} \left[P_{bg}\epsilon_g^k(1-\epsilon_g)^{N-k} + P_{gb}\epsilon_b^k(1-\epsilon_b)^{N-k}\right] \\
&+ \left(\frac{N}{n}\right)! \sum_{\Omega_{i-3\delta}} \sum_{\Omega_{i-2\delta}} \sum_{\Omega_{i-\delta}} \sum_{\Omega_i} \Pr(\Omega_{i-3\delta}, \Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i) \\
&\cdot \sum_{k=d}^{N} \frac{A_k}{\binom{N}{k}} \sum_{j=1}^{|\pi^{(k)}|} \left[\prod_{l=0}^{n} \frac{[P_{pd}(l)]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!} - \prod_{l=0}^{n} \frac{[P_{pd0}(l)]^{\pi_{jl}^{(k)}}}{\pi_{jl}^{(k)}!}\right] \\
P_4^{(x)} &= 1 - C_4^{(x)} - Q_4^{(x)}
\end{aligned}
$$

Appendix B

Issues Regarding the Gilbert-Elliott Model

In this section the δ-step transition probabilities of the Gilbert-Elliott burst-noise channel model are derived. Then the derivation of the expressions for $\Pr(\Omega_i = \mathrm{G})$ and $\Pr(\Omega_i = \mathrm{B})$ is presented. This appendix concludes by deriving the joint δ-step transition probabilities of this channel model.

B.1 δ-step Transition Probabilities

Recall from Section 2.1 the transition probability matrix $\boldsymbol{P}$ of the Gilbert-Elliott model is,

$$\begin{array}{cc} & \begin{array}{cc} \mathrm{G} & \mathrm{B} \end{array} \\ \boldsymbol{P} = & \left[\begin{array}{cc} 1 - P_{gb} & P_{gb} \\ P_{bg} & 1 - P_{bg} \end{array} \right] \begin{array}{c} \mathrm{G} \\ \mathrm{B} \end{array} \end{array}$$

To compute the δ-step transition probabilities it suffices to compute

$$\boldsymbol{P}^\delta = \left[\begin{array}{cc} P_{11}^{(\delta)} & P_{12}^{(\delta)} \\ P_{21}^{(\delta)} & P_{22}^{(\delta)} \end{array} \right],$$

where $P_{ij}(\delta)$ is the δ-step transition probability from state i to state j. Note that this problem will be solved in general by using the transition probabilities P_{ij}. Upon its solution, the transition probabilities of the Gilbert-Elliott model will be substituted for the P_{ij}. The problem is then the computation of the transition probabilities $\delta_{ij}^{(\delta)}$; although this is tedious, it is not complicated. The computation begins by defining $\boldsymbol{P}^\delta$ recursively:

$$
\begin{aligned}
\boldsymbol{P}^{\delta} &= \boldsymbol{P}^{\delta-1}\boldsymbol{P} \\
&= \begin{bmatrix} P_{11}^{(\delta-1)} & P_{12}^{(\delta-1)} \\ P_{21}^{(\delta-1)} & P_{22}^{(\delta-1)} \end{bmatrix} \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \\
&= \begin{bmatrix} P_{11}P_{11}^{(\delta-1)} + P_{21}P_{12}^{(\delta-1)} & P_{12}P_{11}^{(\delta-1)} + P_{22}P_{12}^{(\delta-1)} \\ P_{11}P_{21}^{(\delta-1)} + P_{21}P_{22}^{(\delta-1)} & P_{12}P_{21}^{(\delta-1)} + P_{22}P_{22}^{(\delta-1)} \end{bmatrix}
\end{aligned} \tag{B.1}
$$

Equation B.1 generates two systems of equations as shown below

$$
S_1 : \quad \begin{cases} P_{11}^{(\delta)} = P_{11}P_{11}^{(\delta-1)} + P_{21}P_{12}^{(\delta-1)} \\ P_{12}^{(\delta)} = P_{12}P_{11}^{(\delta-1)} + P_{22}P_{12}^{(\delta-1)} \end{cases}
$$

$$
S_2 : \quad \begin{cases} P_{21}^{(\delta)} = P_{11}P_{21}^{(\delta-1)} + P_{21}P_{22}^{(\delta-1)} \\ P_{22}^{(\delta)} = P_{12}P_{21}^{(\delta-1)} + P_{22}P_{22}^{(\delta-1)} \end{cases}
$$

Since the two systems are similar in structure, S_1 will be explicitly solved by using z-transforms; the solution for S_2 is analogous. To compute the one-sided z-transform of $P_{ij}^{(\delta)}$, it is nessary to compute the initial conditions $P_{11}^{(-1)}$ and $P_{12}^{(-1)}$. Referring to S_1, let $\delta = 0$ to obtain

$$
\begin{aligned}
P_{11}^{(0)} &= P_{11}P_{11}^{(-1)} + P_{21}P_{12}^{(-1)} \\
P_{12}^{(0)} &= P_{12}P_{11}^{(-1)} + P_{22}P_{12}^{(-1)}
\end{aligned}
$$

Since, by definition, $P_{11}^{(0)} = 1$ and $P_{12}^{(0)} = 0$ then $P_{11}^{(-1)}$ and $P_{12}^{(-1)}$ may be found by solving the following system using Cramer's rule for instance.

$$
\begin{bmatrix} P_{11} & P_{21} \\ P_{12} & P_{22} \end{bmatrix} \begin{bmatrix} P_{11}^{(-1)} \\ P_{12}^{(-1)} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}
$$

Then,

$$
\begin{aligned}
P_{11}^{(-1)} &= \frac{P_{22}}{P_{11}P_{22} - P_{12}P_{21}} \\
P_{12}^{(-1)} &= \frac{-P_{12}}{P_{11}P_{22} - P_{12}P_{21}}
\end{aligned} \tag{B.2}
$$

The one-sided z-transform of $P_{ij}^{(\delta)}$ may then be computed. It is defined for a sequence $x(n), n \geq 0$ as

$$
X^{+}(z) \equiv \sum_{n=0}^{\infty} x(n) z^{-n},
$$

and its time delay property is [22]

$$\begin{aligned} x(n) &\overset{z^+}{\longleftrightarrow} X^+(z) \\ x(n-k) &\overset{z^+}{\longleftrightarrow} z^{-k}[X^+(z) + \sum_{n=1}^{k} x(-n)z^n]. \end{aligned}$$

Let $P_{ij}^{(\delta)} \overset{z^+}{\longleftrightarrow} \Pi_{ij}^+(z)$. The one-sided z-transform of the system of equations S_1 is

$$\Pi_{11}^+(z) = P_{11}z^{-1}[\Pi_{11}^+(z) + P_{11}^{(-1)}z] + P_{21}z^{-1}[\Pi_{12}^+(z) + P_{12}^{(-1)}z] \quad \text{(B.3)}$$

$$\Pi_{12}^+(z) = P_{12}z^{-1}[\Pi_{11}^+(z) + P_{11}^{(-1)}z] + P_{22}z^{-1}[\Pi_{12}^+(z) + P_{12}^{(-1)}z] \quad \text{(B.4)}$$

Substitute the values for $P_{11}^{(-1)}$ and $P_{12}^{(-1)}$ computed in Equation B.2 into Equations B.3 and B.4 to obtain:

$$\begin{aligned} \Pi_{11}^+(z)[1 - P_{11}z^{-1}] - P_{21}z^{-1}\Pi_{12}^+(z) &= 1 \\ \Pi_{12}^+(z)[1 - P_{22}z^{-1}] - P_{12}z^{-1}\Pi_{11}^+(z) &= 0 \end{aligned}$$

This system may now be conveniently expressed in matrix form:

$$\begin{bmatrix} 1 - P_{11}z^{-1} & -P_{21}z^{-1} \\ -P_{12}z^{-1} & 1 - P_{22}z^{-1} \end{bmatrix} \begin{bmatrix} \Pi_{11}^+(z) \\ \Pi_{12}^+(z) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

It is easily solved, the result being

$$\begin{aligned} \Pi_{11}^+(z) &= \frac{1 - P_{22}z^{-1}}{(1 - P_{11}z^{-1})(1 - P_{22}z^{-1}) - P_{12}P_{21}z^{-2}} \\ \Pi_{12}^+(z) &= \frac{P_{12}z^{-1}}{(1 - P_{11}z^{-1})(1 - P_{22}z^{-1}) - P_{12}P_{21}z^{-2}} \end{aligned}$$

In order to solve for $P_{ij}^{(\delta)}$, it is required to invert $\Pi_{ij}^+(z)$; the required expression for the inversion is [29]

$$x(n) = \frac{1}{2\pi j}\oint_C X(z)^+ z^{n-1} dz,$$

where $j = \sqrt{-1}$ and C is a contour in the region of convergence of $X^+(z)$ taken in the counterclockwise direction. The inversion of $\Pi_{ij}^+(z)$ is easily accomplished by recalling the residue theorem [29]. A similar procedure may be used to compute the transition probabilities in system S_2. Therefore, after inversion, the δ-step transition probabilities are

$$P_{11}^{(\delta)} = \frac{(1 - P_{22}) + (1 - P_{11})(P_{11} + P_{22} - 1)^\delta}{2 - P_{11} - P_{22}}$$

$$
\begin{aligned}
P_{12}^{(\delta)} &= \frac{P_{12} - P_{12}(P_{11} + P_{22} - 1)^{\delta}}{2 - P_{11} - P_{22}} \\
P_{21}^{(\delta)} &= \frac{P_{21} - P_{21}(P_{11} + P_{22} - 1)^{\delta}}{2 - P_{11} - P_{22}} \\
P_{22}^{(\delta)} &= \frac{(1 - P_{11}) + (1 - P_{22})(P_{11} + P_{22} - 1)^{\delta}}{2 - P_{11} - P_{22}}
\end{aligned}
$$

Returning to the parameters of Gilbert-Elliott's model, the transition probabilities P_{ij} correspond as follows:

$$
\begin{aligned}
P_{11} &= 1 - P_{gb} \\
P_{12} &= P_{gb} \\
P_{21} &= P_{bg} \\
P_{22} &= 1 - P_{bg}
\end{aligned}
$$

Finally, the δ-step transition probabilities of the Gilbert-Elliott model are

$$
\begin{aligned}
P_{gg}^{(\delta)} &= \frac{P_{bg} + P_{gb}(1 - P_{gb} - P_{bg})^{\delta}}{P_{gb} + P_{bg}} \\
P_{gb}^{(\delta)} &= \frac{P_{gb} - P_{gb}(1 - P_{gb} - P_{bg})^{\delta}}{P_{gb} + P_{bg}} \\
P_{bg}^{(\delta)} &= \frac{P_{bg} - P_{bg}(1 - P_{gb} - P_{bg})^{\delta}}{P_{gb} + P_{bg}} \\
P_{bb}^{(\delta)} &= \frac{P_{gb} + P_{bg}(1 - P_{gb} - P_{bg})^{\delta}}{P_{gb} + P_{bg}}
\end{aligned}
$$

B.2 $\Pr(\Omega_i = \mathbf{G})$ and $\Pr(\Omega_i = \mathbf{B})$

These probabilites may be easily computed by using the expressions for $P_{ii}^{(\delta)}$ and letting $\delta \to \infty$. Assuming that $|P_{11} + P_{22} - 1| < 1$, then

$$
\begin{aligned}
\Pr(\Omega_i = \mathrm{G}) &= \lim_{\delta \to \infty} P_{gg}^{(\delta)} = \frac{P_{bg}}{P_{gb} + P_{bg}} \\
\Pr(\Omega_i = \mathrm{B}) &= \lim_{\delta \to \infty} P_{bb}^{(\delta)} = \frac{P_{gb}}{P_{gb} + P_{bg}}
\end{aligned}
$$

as claimed in Section 2.1.

B.3 Joint Transition Probabilities

In this section, the δ-step joint transition probabilties are computed for the depth-2, depth-3 and depth-4 systems.

B.3.1 $\Pr(\Omega_{i-\delta}, \Omega_i)$

Let Ω_i be the current state of the model and $\Omega_{i-\delta}$ be its state δ transitions in the past. Then $\Pr(\Omega_{i-\delta}, \Omega_i)$ may be computed as follows:

$$\Pr(\Omega_{i-\delta}, \Omega_i) = \Pr(\Omega_i \mid \Omega_{i-\delta}) \Pr(\Omega_{i-\delta}),$$

where $\Pr(\Omega_i \mid \Omega_{i-\delta})$ is recognized to be the δ-step transition probability from state $\Omega_{i-\delta}$ to state Ω_i which was computed in Appendix B.1. Therefore,

$$\begin{aligned}
\Pr(\Omega_{i-\delta} = \mathrm{G}, \Omega_i = \mathrm{G}) &= \frac{P_{gb}^2 + P_{gb}P_{bg}(1 - P_{gb} - P_{bg})^\delta}{(P_{gb} + P_{bg})^2} \\
\Pr(\Omega_{i-\delta} = \mathrm{G}, \Omega_i = \mathrm{B}) &= \frac{P_{gb}P_{bg} - P_{gb}P_{bg}(1 - P_{gb} - P_{bg})^\delta}{(P_{gb} + P_{bg})^2} \\
\Pr(\Omega_{i-\delta} = \mathrm{B}, \Omega_i = \mathrm{G}) &= \frac{P_{gb}P_{bg} - P_{gb}P_{bg}(1 - P_{gb} - P_{bg})^\delta}{(P_{gb} + P_{bg})^2} \\
\Pr(\Omega_{i-\delta} = \mathrm{B}, \Omega_i = \mathrm{B}) &= \frac{P_{bg}^2 + P_{gb}P_{bg}(1 - P_{gb} - P_{bg})^\delta}{(P_{gb} + P_{bg})^2}
\end{aligned}$$

B.3.2 $\Pr(\Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i)$

Let Ω_i and $\Omega_{i-\delta}$ be as defined in Appendix B.3.1 and let $\Omega_{i-2\delta}$ be the state of the channel model 2δ transitions in the past, then $\Pr(\Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i)$ may be computed as follows:

$$\Pr(\Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i) = \Pr(\Omega_i \mid \Omega_{i-\delta}) \Pr(\Omega_{i-\delta} \mid \Omega_{i-2\delta}) \Pr(\Omega_{i-2\delta}),$$

where $\Pr(\Omega_i \mid \Omega_{i-\delta})$ is as defined in Appendix B.3.1 and $\Pr(\Omega_{i-\delta} \mid \Omega_{i-2\delta})$ is the δ-step transition probability from $\Omega_{i-2\delta}$ to $\Omega_{i-\delta}$. Therefore,

$$\begin{aligned}
&\Pr(\Omega_{i-2\delta} = \mathrm{G}, \Omega_{i-\delta} = \mathrm{G}, \Omega_i = \mathrm{G}) \\
&= \frac{(P_{bg} + P_{gb}(1 - P_{gb} - P_{bg})^\delta)^2 P_{bg}}{(P_{gb} + P_{bg})^3}
\end{aligned}$$

$$\Pr(\Omega_{i-2\delta} = \mathrm{G}, \Omega_{i-\delta} = \mathrm{G}, \Omega_i = \mathrm{B}) = \frac{(P_{gb} - P_{gb}(1 - P_{gb} - P_{bg})^{\delta})(P_{bg} + P_{gb}(1 - P_{gb} - P_{bg})^{\delta})P_{bg}}{(P_{gb} + P_{bg})^3}$$

$$\Pr(\Omega_{i-2\delta} = \mathrm{G}, \Omega_{i-\delta} = \mathrm{B}, \Omega_i = \mathrm{G}) = \frac{(P_{bg} - P_{bg}(1 - P_{gb} - P_{bg})^{\delta})(P_{gb} - P_{gb}(1 - P_{gb} - P_{bg})^{\delta})P_{bg}}{(P_{gb} + P_{bg})^3}$$

$$\Pr(\Omega_{i-2\delta} = \mathrm{G}, \Omega_{i-\delta} = \mathrm{B}, \Omega_i = \mathrm{B}) = \frac{(P_{gb} + P_{bg}(1 - P_{gb} - P_{bg})^{\delta})(P_{gb} - P_{gb}(1 - P_{gb} - P_{bg})^{\delta})P_{bg}}{(P_{gb} + P_{bg})^3}$$

$$\Pr(\Omega_{i-2\delta} = \mathrm{B}, \Omega_{i-\delta} = \mathrm{G}, \Omega_i = \mathrm{G}) = \frac{(P_{bg} + P_{gb}(1 - P_{gb} - P_{bg})^{\delta})(P_{bg} - P_{bg}(1 - P_{gb} - P_{bg})^{\delta})P_{gb}}{(P_{gb} + P_{bg})^3}$$

$$\Pr(\Omega_{i-2\delta} = \mathrm{B}, \Omega_{i-\delta} = \mathrm{G}, \Omega_i = \mathrm{B}) = \frac{(P_{gb} - P_{gb}(1 - P_{gb} - P_{bg})^{\delta})(P_{bg} - P_{bg}(1 - P_{gb} - P_{bg})^{\delta})P_{gb}}{(P_{gb} + P_{bg})^3}$$

$$\Pr(\Omega_{i-2\delta} = \mathrm{B}, \Omega_{i-\delta} = \mathrm{B}, \Omega_i = \mathrm{G}) = \frac{(P_{bg} - P_{bg}(1 - P_{gb} - P_{bg})^{\delta})(P_{gb} + P_{bg}(1 - P_{gb} - P_{bg})^{\delta})P_{gb}}{(P_{gb} + P_{bg})^3}$$

$$\Pr(\Omega_{i-2\delta} = \mathrm{B}, \Omega_{i-\delta} = \mathrm{B}, \Omega_i = \mathrm{B}) = \frac{(P_{gb} + P_{bg}(1 - P_{gb} - P_{bg})^{\delta})^2 P_{gb}}{(P_{gb} + P_{bg})^3}$$

B.3.3 $\Pr(\Omega_{i-3\delta}, \Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i)$

Let Ω_i, $\Omega_{i-\delta}$ and $\Omega_{i-2\delta}$ be as defined in Appendix B.3.2 and let $\Omega_{i-3\delta}$ be the state of the channel model 3δ transitions in the past, then $\Pr(\Omega_{i-3\delta}, \Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i)$ may be computed as follows:

$$\Pr(\Omega_{i-3\delta}, \Omega_{i-2\delta}, \Omega_{i-\delta}, \Omega_i) = \Pr(\Omega_i \mid \Omega_{i-\delta}) \Pr(\Omega_{i-\delta} \mid \Omega_{i-2\delta}) \Pr(\Omega_{i-2\delta} \mid \Omega_{i-3\delta}) \Pr(\Omega_{i-3\delta}),$$

where $\Pr(\Omega_i \mid \Omega_{i-\delta})$ and $\Pr(\Omega_{i-\delta} \mid \Omega_{i-2\delta})$ are as defined in Appendix B.3.2 and $\Pr(\Omega_{i-2\delta} \mid \Omega_{i-3\delta})$ is the δ-step transition probability from $\Omega_{i-3\delta}$ to $\Omega_{i-2\delta}$. Therefore,

$$\Pr(\Omega_{i-3\delta} = \mathrm{G}, \Omega_{i-2\delta} = \mathrm{G}, \Omega_{i-\delta} = \mathrm{G}, \Omega_i = \mathrm{G}) = \frac{(P_{bg} + P_{gb}(1 - P_{gb} - P_{bg})^{\delta})^3 P_{bg}}{(P_{gb} + P_{bg})^4}$$

$$\Pr(\Omega_{i-3\delta} = \mathrm{G}, \Omega_{i-2\delta} = \mathrm{G}, \Omega_{i-\delta} = \mathrm{G}, \Omega_i = \mathrm{B}) = \frac{(P_{gb} - P_{gb}(1 - P_{gb} - P_{bg})^{\delta})(P_{bg} + P_{gb}(1 - P_{gb} - P_{bg})^{\delta})^2 P_{bg}}{(P_{gb} + P_{bg})^4}$$

$$\Pr(\Omega_{i-3\delta} = \mathrm{G}, \Omega_{i-2\delta} = \mathrm{G}, \Omega_{i-\delta} = \mathrm{B}, \Omega_i = \mathrm{G}) = \frac{(P_{bg} - P_{bg}(1 - P_{gb} - P_{bg})^{\delta})(P_{gb} - P_{gb}(1 - P_{gb} - P_{bg})^{\delta})(P_{bg} + P_{gb}(1 - P_{gb} - P_{bg})^{\delta})P_{bg}}{(P_{gb} + P_{bg})^4}$$

$$\Pr(\Omega_{i-3\delta} = \mathrm{G}, \Omega_{i-2\delta} = \mathrm{G}, \Omega_{i-\delta} = \mathrm{B}, \Omega_i = \mathrm{B}) = \frac{(P_{gb} + P_{bg}(1 - P_{gb} - P_{bg})^{\delta})(P_{gb} - P_{gb}(1 - P_{gb} - P_{bg})^{\delta})(P_{bg} + P_{gb}(1 - P_{gb} - P_{bg})^{\delta})P_{bg}}{(P_{gb} + P_{bg})^4}$$

$$\Pr(\Omega_{i-3\delta} = \mathrm{G}, \Omega_{i-2\delta} = \mathrm{B}, \Omega_{i-\delta} = \mathrm{G}, \Omega_i = \mathrm{G}) = \frac{(P_{bg} + P_{gb}(1 - P_{gb} - P_{bg})^{\delta})(P_{bg} - P_{bg}(1 - P_{gb} - P_{bg})^{\delta})(P_{gb} - P_{gb}(1 - P_{gb} - P_{bg})^{\delta})P_{bg}}{(P_{gb} + P_{bg})^4}$$

$$\Pr(\Omega_{i-3\delta} = \mathrm{G}, \Omega_{i-2\delta} = \mathrm{B}, \Omega_{i-\delta} = \mathrm{G}, \Omega_i = \mathrm{B}) = \frac{(P_{gb} - P_{gb}(1 - P_{gb} - P_{bg})^{\delta})^2(P_{bg} - P_{bg}(1 - P_{gb} - P_{bg})^{\delta})P_{bg}}{(P_{gb} + P_{bg})^4}$$

$$\Pr(\Omega_{i-3\delta} = \mathrm{G}, \Omega_{i-2\delta} = \mathrm{B}, \Omega_{i-\delta} = \mathrm{B}, \Omega_i = \mathrm{G}) = \frac{(P_{bg} - P_{bg}(1 - P_{gb} - P_{bg})^{\delta})(P_{gb} + P_{bg}(1 - P_{gb} - P_{bg})^{\delta})(P_{gb} - P_{gb}(1 - P_{gb} - P_{bg})^{\delta})P_{bg}}{(P_{gb} + P_{bg})^4}$$

$$\Pr(\Omega_{i-3\delta} = \mathrm{G}, \Omega_{i-2\delta} = \mathrm{B}, \Omega_{i-\delta} = \mathrm{B}, \Omega_i = \mathrm{B}) = \frac{(P_{gb} + P_{bg}(1 - P_{gb} - P_{bg})^{\delta})^2(P_{gb} - P_{gb}(1 - P_{gb} - P_{bg})^{\delta})P_{bg}}{(P_{gb} + P_{bg})^4}$$

$$\Pr(\Omega_{i-3\delta} = \mathrm{B}, \Omega_{i-2\delta} = \mathrm{G}, \Omega_{i-\delta} = \mathrm{G}, \Omega_i = \mathrm{G}) = \frac{(P_{bg} + P_{gb}(1 - P_{gb} - P_{bg})^{\delta})^2(P_{bg} - P_{bg}(1 - P_{gb} - P_{bg})^{\delta})P_{gb}}{(P_{gb} + P_{bg})^4}$$

$$\Pr(\Omega_{i-3\delta} = \mathrm{B}, \Omega_{i-2\delta} = \mathrm{G}, \Omega_{i-\delta} = \mathrm{G}, \Omega_i = \mathrm{B})$$
$$= \frac{(P_{gb} - P_{gb}(1 - P_{gb} - P_{bg})^\delta)(P_{bg} + P_{gb}(1 - P_{gb} - P_{bg})^\delta)(P_{bg} - P_{bg}(1 - P_{gb} - P_{bg})^\delta)P_{gb}}{(P_{gb} + P_{bg})^4}$$

$$\Pr(\Omega_{i-3\delta} = \mathrm{B}, \Omega_{i-2\delta} = \mathrm{G}, \Omega_{i-\delta} = \mathrm{B}, \Omega_i = \mathrm{G})$$
$$= \frac{(P_{bg} - P_{bg}(1 - P_{gb} - P_{bg})^\delta)^2(P_{gb} - P_{gb}(1 - P_{gb} - P_{bg})^\delta)P_{gb}}{(P_{gb} + P_{bg})^4}$$

$$\Pr(\Omega_{i-3\delta} = \mathrm{B}, \Omega_{i-2\delta} = \mathrm{G}, \Omega_{i-\delta} = \mathrm{B}, \Omega_i = \mathrm{B})$$
$$= \frac{(P_{gb} + P_{bg}(1 - P_{gb} - P_{bg})^\delta)(P_{gb} - P_{gb}(1 - P_{gb} - P_{bg})^\delta)(P_{bg} - P_{bg}(1 - P_{gb} - P_{bg})^\delta)P_{gb}}{(P_{gb} + P_{bg})^4}$$

$$\Pr(\Omega_{i-3\delta} = \mathrm{B}, \Omega_{i-2\delta} = \mathrm{B}, \Omega_{i-\delta} = \mathrm{G}, \Omega_i = \mathrm{G})$$
$$= \frac{(P_{bg} + P_{gb}(1 - P_{gb} - P_{bg})^\delta)(P_{bg} - P_{bg}(1 - P_{gb} - P_{bg})^\delta)(P_{gb} + P_{bg}(1 - P_{gb} - P_{bg})^\delta)P_{gb}}{(P_{gb} + P_{bg})^4}$$

$$\Pr(\Omega_{i-3\delta} = \mathrm{B}, \Omega_{i-2\delta} = \mathrm{B}, \Omega_{i-\delta} = \mathrm{G}, \Omega_i = \mathrm{B})$$
$$= \frac{(P_{gb} - P_{gb}(1 - P_{gb} - P_{bg})^\delta)(P_{bg} - P_{bg}(1 - P_{gb} - P_{bg})^\delta)(P_{gb} + P_{bg}(1 - P_{gb} - P_{bg})^\delta)P_{gb}}{(P_{gb} + P_{bg})^4}$$

$$\Pr(\Omega_{i-3\delta} = \mathrm{B}, \Omega_{i-2\delta} = \mathrm{B}, \Omega_{i-\delta} = \mathrm{B}, \Omega_i = \mathrm{G})$$
$$= \frac{(P_{bg} - P_{bg}(1 - P_{gb} - P_{bg})^\delta)(P_{gb} + P_{bg}(1 - P_{gb} - P_{bg})^\delta)^2 P_{gb}}{(P_{gb} + P_{bg})^4}$$

$$\Pr(\Omega_{i-3\delta} = \mathrm{B}, \Omega_{i-2\delta} = \mathrm{B}, \Omega_{i-\delta} = \mathrm{B}, \Omega_i = \mathrm{B})$$
$$= \frac{(P_{gb} + P_{bg}(1 - P_{gb} - P_{bg})^\delta)^3 P_{gb}}{(P_{gb} + P_{bg})^4}$$

Appendix C

$P_C(w)$ of Selected KM Codes

This appendix summarizes the error correction capabilities $P_c(w)$ of selected KM codes. The generator matrices for these codes are presented in [1]. The KM code decoder is implemented by selecting the codeword which is closest (in the sense of Hamming distance) to the combined block in question. Note that whenever the combined block $\mathcal{B}$ currently under consideration is equally close to the zero-codeword as well as a non-zero codeword, the latter codeword is chosen as the decoder output so that $\mathcal{B}$ is considered uncorrectable. This ensures a lower bound on the probability of successful error correction decoding.

	$P_c(w)$	
w	(8,4,3) KM Code	(12,4,5) KM Code
0	1.0	1.0
1	1.0	1.0
2	0.0	1.0
3	0.0	0.12727
$4 \leq w \leq 8$	0.0	0.0
$9 \leq w \leq 12$	–	0.0

Table C.1: $P_c(w)$ of the (8,4,3) and (12,4,5) KM Codes

	$P_c(w)$	
w	(10,5,3) KM Code	(15,5,5) KM Code
0	1.0	1.0
1	1.0	1.0
2	6.6667×10^{-2}	1.0
3	0.0	0.76483
4	0.0	0.12234
$5 \leq w \leq 10$	0.0	0.0
$11 \leq w \leq 15$	–	0.0

Table C.2: $P_c(w)$ of the (10,5,3) and (15,5,5) KM Codes

	$P_c(w)$		
w	(12,6,3) KM Code	(18,6,6) KM Code	(24,6,9) KM Code
0	1.0	1.0	1.0
1	1.0	1.0	1.0
2	0.40909	1.0	1.0
3	4.5455×10^{-3}	0.90564	1.0
4	0.0	0.45523	1.0
5	0.0	1.6223×10^{-2}	0.91481
6	0.0	0.0	0.56123
7	0.0	0.0	9.9724×10^{-2}
8	0.0	0.0	4.1606×10^{-4}
$9 \leq w \leq 12$	0.0	0.0	0.0
$13 \leq w \leq 18$	–	0.0	0.0
$19 \leq w \leq 24$	–	–	0.0

Table C.3: $P_c(w)$ of the (12,6,3), (18,6,6) and (24,6,9) KM Codes

	$P_c(w)$	
w	(14,7,3) KM Code	(21,7,6) KM Code
0	1.0	1.0
1	1.0	1.0
2	0.52747	1.0
3	5.4945×10^{-3}	0.98496
4	0.0	0.74052
5	0.0	0.19441
6	0.0	1.3637×10^{-3}
$7 \leq w \leq 14$	0.0	0.0
$15 \leq w \leq 21$	–	0.0

Table C.4: $P_c(w)$ of the (14,7,3) and (21,7,6) KM Codes

	$P_c(w)$	
w	(16,8,3) KM Code	(24,8,6) KM Code
0	1.0	1.0
1	1.0	1.0
2	0.57500	1.0
3	0.0	0.98024
4	0.0	0.83851
5	0.0	0.42024
6	0.0	4.7587×10^{-2}
7	0.0	1.4735×10^{-4}
$8 \leq w \leq 16$	0.0	0.0
$17 \leq w \leq 24$	–	0.0

Table C.5: $P_c(w)$ of the (16,8,3) and (24,8,6) KM Codes