Automatic repeat request on fading channels

Patrick Beirouti

Department of Electrical Engineering
McGill University
Montréal, Canada

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

Automatic repeat request (ARQ) is a widespread technique for error control in data communication systems. This research examines the performance of conventional ARQ schemes over fading channels. The basic Gilbert-Elliott two-state Markov model is used to represent these channels. This basic model is recursively extended to a 2^n-state Markov chain suitable for n-bit block codes. Using this 'Extended Gilbert-Elliott' model, an approximation of the throughput efficiency of the conventional ARQ schemes is derived. This approximation is particularly valid for slow fading channels. Furthermore, performance plots are obtained, showing the effects of channel fading on throughput efficiency: ARQ performance deteriorates with slower fading, or alternatively, higher channel memory. Consequently, frequency-hopped codeword transmission is explored, a technique by which channel memory can be reduced. Throughput performance of frequency-hopped ARQ systems is derived, which shows significant potential improvements over systems with no frequency hopping.

Précis

La retransmission à requête automatique (ARQ) est l'une des techniques les plus répandues pour le controle d'erreurs dans les systèmes de communications. Cette recherche traite de la performance des techniques ARQ conventionnelles sur des canaux à mémoire. Ces canaux sont représentés par une chaine de Markov à deux états, mieux connue sous le nom de 'modèle Gilbert-Elliott'. Ce modèle de base est ensuite augmenté, à l'aide d'une méthode récursive, à une chaine de Markov à 2^n états. En utilisant cette extension au modèle Gilbert-Elliott, on peut alors représenter les changements d'état entre des mots de code de n bits de long, et ainsi dériver une approximation de la performance, en terme de débit, des systèmes ARQ conventionnels. Cette approximation est valide surtout lorsque l'on opère sur des canaux à haute mémoire. Cette recherche montre également, à l'aide de plusieurs graphes, les effets de la mémoire du canal sur le débit des systèmes ARQ. Selon ces graphes, la performance des systèmes ARQ est affectée négativement par une croissance de la mémoire du canal. Conséquemment, nous examinons des systèmes ARQ avec changement automatique de fréquence, car cette dernière technique constitue un moyen de reduire la mémoire du canal. Des expressions montrant le débit de tels systèmes sont derivées. Ces expressions montrent qu'une ammélioration substantielle peut être obtenue par l'utilisation de systèmes ARQ à changement automatique de fréquence.

Acknowledgements

Many thanks are due to Dr. H. Leib for his close supervision The numerous discussions we had throughout my research were very helpful in bringing this work to fruition. I am also very grateful to Dr. S. D. Morgera for his guidance, insight and support.

I am also indebted to Kiran Mehta, Patrick Lie Chin Cheong and Ronny Quesnel for many constructive discussions. Thanks to Salvatore Torrente and Albert Pang for their help with the computer systems at the Information Networks and Systems Lab. To Vince Iacoviello and Jacek Stachurski for breaking the daily routine at school, and for many enjoyable evenings at Thomson House. Many thanks to my parents for their continuous support. The value they place in education was a decisive factor in my university studies.

Finally, I would like to acknowledge the Natural Sciences and Engineering Research Council of Canada for its financial assistance, as well as the Canadian Institute for Telecommunications Research for sponsoring this project.

Contents

List of Figures

Chapter 1

Introduction

In this chapter, some background material pertaining to the subject matter of this thesis is presented. A general introduction to digital communication systems and communications theory is the subject of the first section. Particular attention is given to channel coding techniques, such as forward error correction and automatic repeat request. The latter scheme is described at length in Section 1.3, as automatic repeat request schemes constitute the main object of our study. In Section 1.4, fading channels, which are the transmission media of interest in this research, are briefly described. Finally, a summary of the issues discussed in this thesis is given at the end of this introductory chapter.

1.1 Communication Systems

The study of communication systems is primarily concerned with the reliable transmission of information from one point to another across a given medium. This transmission medium is commonly referred to as a communications channel, or simply *channel*. Typical channels include telephone lines, satellite communication links, wireless radio links, data storage media, etc. The information to be transmitted over such channels could be, for example, speech, music, video images, weather data, etc. According to the information theory work developed by Shannon in the 1940's, information can always be reliably represented as a sequence of binary digits [1]. The problem of information representation is referred to as *source coding*. In this research, we assume information is already available in binary format and focus instead on ways

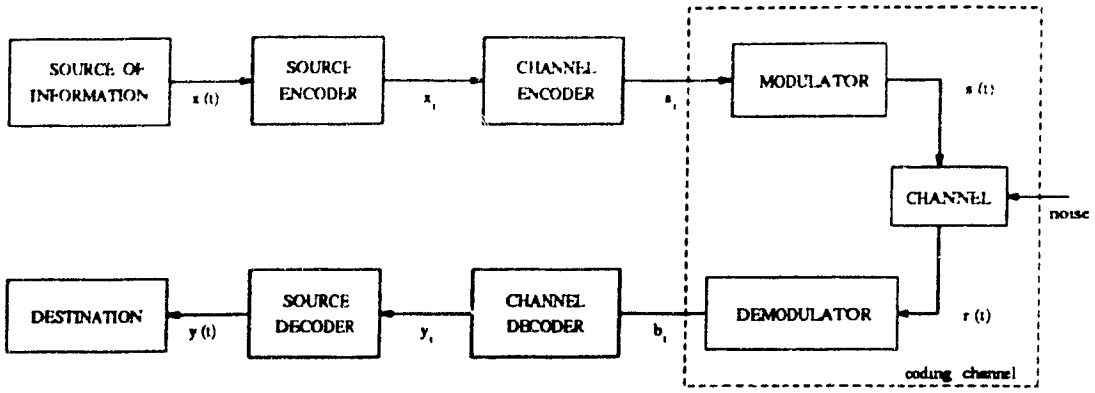

Figure 1.1: **Digital communications system**

of reliably sending the binary data over the transmission medium or channel.

Information transmission is almost always subject to some kind of disturbance, commonly referred to as *noise*. This disturbance could be in the form of thermal noise, interference from other users of the same channel, physical obstacles to the propagation of electromagnetic waves, and so on. The presence of noise over a channel causes errors to occur in the data stream In this research, we are interested in the problem of *channel coding*, that is the study of various techniques aimed at reducing errors in the information delivered at the receiving end of the communication system. A block diagram of a typical digital communication system is shown in Figure 1 1.

The diagram of Figure 1.1 shows the information flow and processing in the communication system. Information is generated by the source, where it is represented by the time waveform $x(t)$. This waveform is then processed by a source encoder, which is responsible for transforming the information into a time-discrete sequence of binary digits x_t. This binary sequence is then encoded into another binary sequence a_t. Encoding source data is done so that any errors corrupting the data as a result of channel noise can be detected and corrected at the receiving end of the communication system. The correction for errors is done in conjunction with the channel decoder at the receiving site. The data stream a_t resulting from the channel encoder is subsequently modulated into a time signal $s(t)$ suitable for transmission over a given real channel. At the receiving end of the channel, the resulting signal $r(t)$ is demodulated back into a binary sequence b_t, which is then fed to the channel decoder for error detection and correction. The resulting error-corrected sequence y_t is then delivered to the destination user, via the source decoder which returns the data sequence to its original form. Since the aim of a communication system is to

transmit information from one point to another as reliably as possible, the role of channel coding is to return to the source decoder a sequence y_t that is as close to the original sequence x_t as possible. Note that for the problem of channel coding, i.e. the design of the channel encoder and decoder pair, the modulator, demodulator and physical channel are considered as one and referred to collectively as the *coding channel.*

The performance of a communication system is measured via a number of parameters. For analog communication systems, performance is measured in terms of output signal to noise ratio (SNR), frequency spectrum occupancy (or bandwidth) and power consumption. In digital communication systems, the parameters measuring performance are, instead, output bit error rate, information delivery delay, bandwidth, and power. The frequency spectrum is usually a scarce resource in most ground-based communication systems. Thus, in the design of such systems, power consumption is assumed constant, and one looks at minimizing output bit error rate while keeping bandwidth low. In satellite communications, on the other hand, frequency bandwidth is usually plentiful, and one looks at keeping power consumption low while having an acceptably low error rate. Thus, the designer of digital communication systems is usually faced with a tradeoff between bandwidth, bit error rate, information delivery delay, and power consumption. After this general overview of communication systems, we now examine the specific problem of channel coding. For a more exhaustive treatment of communication systems, the reader is referred to various books on classical communications theory [2] - [4].

1.2 Channel Coding

In a few words, channel coding is the design of encoding and decoding schemes so that source data can be purged of errors resulting from channel noise. Channel encoders / decoders can be either of the ***block code*** type or of the ***convolutional code*** type. In this research we focus only on block coding techniques. Now, block encoders / decoders can be readily classified in one of two distinct categories. These are the forward error correcting (FEC) schemes and the automatic repeat request (abbreviated as ARQ) schemes. In both schemes, data is segmented in blocks of k bits. These data packets are then transformed or encoded into blocks of n bits, also referred to as *codewords*, with $n > k$. In so doing, additional information is added to the k bits in a predictable

fashion. These $n-k$ bits of additional information, also referred to as *parity bits*, are used at the receiver to detect and possibly correct for possible errors in the data. One can also picture the parity bits as 'backup' information, which is used at the receiver to double-check the validity of the data and to correct for any errors that may have occurred over the transmission channel. Since data is partitioned in packets of k bits, each codeword at the output of the channel encoder can take on one of 2^k possible sequences. The mapping of the 2^k k-bit data packets into n-bit codewords is referred to as a (n,k) *block code*. For convenience, we denote the k-bit data at the input of the channel encoder as vector $\underline{x}$, and the n-bit codewords at the output of the channel encoder as vector $\underline{a}$. Again, the set of all vectors $\underline{a}$ which are mappings of all possible input vectors $\underline{x}$ is referred to as a code, which we denote by $\mathcal{C}$. Since there are 2^k possible input vectors $\underline{x}$ to the channel encoder, the set $\mathcal{C}$ has a total of 2^k elements. Now, since one can form 2^n possible sequences with a n-bit vector $\underline{a}$, there are $2^n - 2^k$ n-bit sequences which do not belong to the code $\mathcal{C}$. This property is used at the channel decoder to detect errors induced by channel noise. Indeed, when a codeword $\underline{a} \in \mathcal{C}$ is transmitted over the channel, one of three situations may occur:

1. A codeword $\underline{b} = \underline{a}$ is received at the output of the channel. No error has occurred
2. Due to channel noise, $\underline{b} \neq \underline{a}$ and $\underline{b} \notin \mathcal{C}$. Vector $\underline{b}$ received at the output of the channel is incorrect. However, since $\underline{b} \notin \mathcal{C}$, the channel decoder can declare with certainty that $\underline{b}$ contains one or more errors
3. Due to channel noise, $\underline{b} \neq \underline{a}$. However, this time $\underline{b} \in \mathcal{C}$. Channel noise is such that it has transformed codeword $\underline{a}$ into another codeword $\underline{b}$. Since $\underline{b} \in \mathcal{C}$, the channel decoder cannot detect the error(s).

Case 2 above shows how channel decoders can detect that the received vector $\underline{b}$ contains one or more errors. However, as shown by case 3, error detection is not foolproof. It is likely that channel noise is such that a codeword $\underline{a}$ is transformed into another codeword $\underline{b}$ which also belongs to the code $\mathcal{C}$. In such a case, the channel decoder cannot determine that an error has occurred. The probability of case 3 occurring, or *probability of undetected error* can be reduced by making n large compared to k (so that $2^n \gg 2^k$) and by careful design of the encoding scheme. The probability of undetected error P_u is a figure of merit of any (n,k) block code $\mathcal{C}$; the smaller P_u, the better the code $\mathcal{C}$.

In forward error correcting codes, the size of the n-bit codeword $\underline{a}$ is larger than the information vector $\underline{x}$. The number $n-k$ of parity bits is typically larger for error correcting schemes than it is for error detection. In other words, for FEC codes, the total number of sequences that can be represented with n bits is much larger than the code size (i.e. $2^n \gg 2^k$). In so doing, if a received vector $\underline{b}$ has only a small number of bit errors, the decoder may still be able to recognize the original codeword $\underline{a}$, provided the code is appropriately designed. Usually, the larger the number of parity bits per information bit, the larger the error detecting or correcting capability of the channel coding scheme. This is only a general rule, however. The exact error-detecting and error-correcting capability of a channel coding scheme depends on the particular code used. For a complete treatment of error control codes, the reader is referred to [5] and [6]. For the purpose of the present discussion, it suffices to note that for a fixed information rate, the channel bit rate must be increased by a factor n/k to obtain an error-correcting capability. This increase in channel bit rate requires a parallel increase in channel bandwidth, however, and bandwidth is often a scarce resource in communication systems. For a given modulation scheme, bandwidth is inversely proportional to the k/n ratio, also known as the *code rate*.

From the above discussion, one can see a clear tradeoff between bandwidth and probability of error in the receiver, in FEC channel coding schemes. This tradeoff assumes the information is delivered to the user at a fixed rate and the transmission power level is fixed. Automatic repeat request schemes (ARQ) are different in this respect; in ARQ schemes, the probability of bit error at the receiver is fixed, and channel bandwidth is traded off with information delivery rate. In ARQ systems, codewords are transmitted over the channel and checked for errors at the receiver or channel decoder. If a codeword is detected in error, a retransmission of that same codeword is requested by the receiver. Automatic repeat request schemes are thus considered as *feedback error control* techniques. The probability of undetected error is dependent on the particular block code used. From the above arguments, it can be seen that ARQ schemes provide a fixed bit error rate to the end user, no matter how bad the transmission channel is. On the other hand, for bad channels, many retransmissions may be necessary, and thus a large delay may be needed before the data is delivered to its destination. Hence, the performance of a particular ARQ scheme over a given channel is not measured in terms of its probability of codeword error, as it is the case for FEC schemes, but rather in terms of its so called *throughput*. ARQ error-control schemes are treated in more detail in the next section.

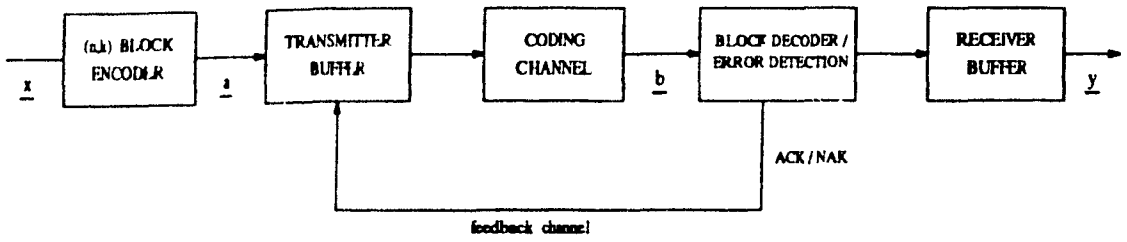

Figure 1.2: **ARQ error control system**

1.3 Automatic Repeat Request

In this section, the various feedback error-control schemes are examined. Various protocols exist for feedback communications, the best known or *conventional* ARQ protocols being Stop-and-Wait (SW), Go-Back-N (GBN) and Selective Repeat (SR). In all ARQ systems, data is first encoded using a (n, k) error detecting code and sent as n-bit codewords $\underline{a}$ over the channel. At the receiver, the channel decoder checks each received data vector for errors. If an error is detected, a negative acknowledgement (NAK) message is sent back to the transmitter, thus requesting a retransmission. The block diagram of an ARQ error control system is shown in Figure 1.2.

The simplest ARQ scheme is the so called *Stop-and-Wait* (SW) protocol. In the SW protocol, the transmitter must wait for either a positive (ACK) or a negative (NAK) acknowledgement signal before sending any subsequent codeword. Thus, after each data packet is sent, the transmitter must wait idle until it receives an ACK / NAK signal from the receiver. The exact idle time depends on such factors as channel propagation delay and data processing delay. In communication networks, this time delay can also depend on the time needed to service a queue of data packets arriving at a network node [7], [8]. Because of this delay, the Stop-and-Wait scheme can become rather inefficient.

The performance of feedback error-control systems was briefly discussed in the previous section. Since many retransmissions of the same codeword may be required, an adequate measure of performance in ARQ systems is data throughput efficiency. Throughput, denoted by η, is defined as the ratio of information bits delivered to the destination user to the total number of bits transmitted. The latter quantity includes the number of bits that could have been sent during the time the transmitter was

idle. For the SW ARQ protocol, throughput can thus be written as

$$\eta_{SW} = \frac{k}{n + R\tau} \frac{1}{E[T]}, \tag{1.1}$$

where R is the data bit rate (in bits/s), τ is the channel round trip delay or idle time (in seconds), and $E[T]$ is the average number of transmissions of the same codeword. The parameters k and n are the number of bits at the input and output of the (n, k) channel encoder, respectively. Note that the product $R\tau$ gives the idle time in bits. Also, as expected, throughput is inversely proportional to the average number of codeword transmissions $E[T]$. Indeed, the larger the number of transmissions required to get a data codeword correctly across the channel, the poorer the performance of the ARQ scheme, and thus, the lower its throughput efficiency. Also worth noting is that throughput is normalized to unity, with $\eta = 1$ being the maximum attainable throughput.

Due to the $R\tau$ term in the denominator of Equation 1.1, the throughput efficiency of the Stop-and-Wait ARQ protocol can become very small for large channel idle time. In order to circumvent such shortcoming, other so called *continuous* ARQ protocols have been devised. In these protocols, the transmitter no longer waits for an acknowledgement before sending the next codeword. The simplest such continuous scheme is the Go-Back-N (GBN) protocol, whose name somewhat indicates its operating principle. In GBN ARQ, the transmitter sends data codewords continuously, until a negative acknowledgement is received, at which time the transmitter backs up by a sufficient number of codewords N to retransmit the codeword in error and all subsequent codewords. The number of codewords N by which the transmitter backs up is equivalent to the channel idle time. The transmitter also needs to keep in memory the last N codewords transmitted, in case a retransmission is requested. Thus, a buffer of size $N \times n$ bits is required at the transmitter. A diagram illustrating the operation of a typical Go-Back-N ARQ system is given in Figure 1.3. As for the throughput performance of the GBN protocol, it can be written as

$$\begin{aligned} \eta_{GBN} &= \frac{k}{n} \frac{1}{1 + N(E[T] - 1)} \\ &= \frac{k}{n} \frac{1}{NE[T] - (N - 1)}. \end{aligned} \tag{1.2}$$

The above expression can be easily justified. The denominator on the first line of Equation 1.2 indicates that if only one transmission is required, only one codeword is sent over the channel; this is the case of $E[T] = 1$. If more than one codeword

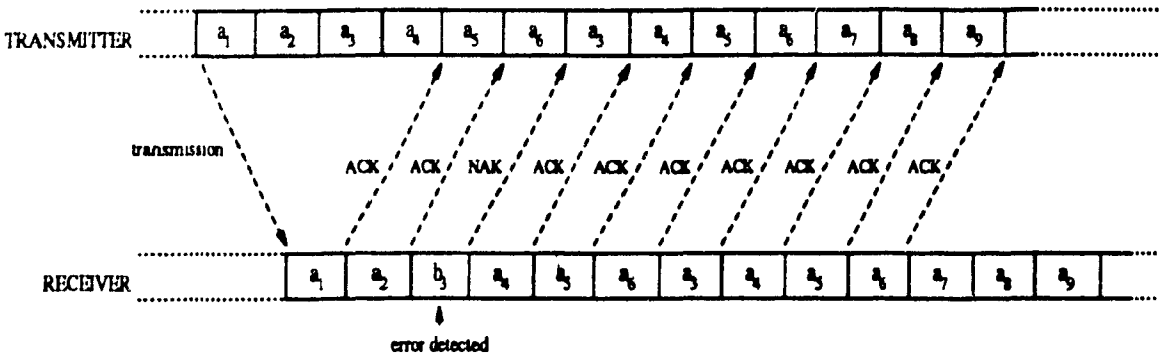

Figure 1.3: **Illustration of the Go-Back-N protocol (with $N = 4$)**

transmission is required, the excess number of codewords (i.e. $E[T] - 1$) is transmitted along with $N - 1$ subsequent codewords since the transmitter backs up by N codewords; this gives the $N(E[T] - 1)$ term in the above equation.

From the discussion in the previous paragraph, it can be seen that, in the GBN protocol, every time a codeword is detected in error, that codeword is retransmitted along with the $N - 1$ codewords that follow. The $N - 1$ codewords following the codeword in error are retransmitted regardless of whether these data packets are correct or not. If these codewords were transmitted correctly in the first place, then resending them along with the codeword in error is obviously wasteful. In order to circumvent this shortfall, a so called *Selective Repeat* scheme can be used. In Selective Repeat (SR) ARQ, only the codeword in error is retransmitted. This scheme is more efficient in terms of throughput performance than the GBN protocol. Indeed, throughput efficiency for the SR protocol is

$$\eta_{SR} = \frac{k}{n}\frac{1}{E[T]}. \tag{1.3}$$

However, buffers (as in Figure 1.2) are needed both at the transmitter and receiver to implement the SR scheme. If a codeword is received in error at the output of the channel, all subsequent codewords must be kept in memory until that erroneous codeword is replaced by a correct copy. This is necessary so as to keep the correct order in the data delivered to the destination user. Now, the size of the buffer required at the receiver depends on the channel idle time and on how many retransmissions are necessary to get a codeword correctly across the channel. However, the number T of transmissions of the same codeword across the channel is a random variable with a nonzero probability distribution over the entire range $[1, \infty)$. Thus, there is a nonzero probability that the number of transmissions T is very large. Hence,

ideally, the buffer required at the receiver must be infinitely large. Of course, this cannot be achieved in practice, and one must contend with a finite size buffer. The use of receiver buffers of finite size causes a degradation in the throughput efficiency of SR ARQ schemes, as compared to the infinite buffer case [5]. The expression in Equation 1.3 gives throughput efficiency for the selective repeat protocol when an infinite receiver buffer is assumed.

The three ARQ protocols discussed here differ significantly in complexity. The SW protocol is the simplest of the three. It requires only a (n, k) block encoder, a small buffer to store the last codeword transmitted and a decoder for its implementation. In the GBN protocol, a buffer of size $N \times n$ bits is required at the transmitter so as to keep in memory the N codewords that may have to be transmitted. Finally, the implementation of the SR protocol requires in addition to a transmitter buffer, an ideally infinite buffer at the receiver. Although complex to implement, the latter scheme offers a throughput performance superior to that of its two simpler counterparts, as shown in Equations 1.1 - 1.3. Clearly, the three ARQ protocols trade throughput performance for hardware complexity. Depending on the application, one scheme may be more advantageous than the others. For instance, the Go-Back-N protocol is used in the Synchronous Data Link Control (SDLC) computer communications protocol [8], as it offers higher throughput performance than the Stop-and-Wait ARQ, yet it is not too demanding in its hardware requirements.

The expressions in Equations 1.1 - 1.3 and all subsequent work in this thesis assumes error free transmission of ACK / NAK signals. Some investigation of the case in which errors can occur in the acknowledgement signal has been undertaken [9], [10]. However, most research in ARQ systems assumes acknowledgement signals are error free. This assumption is quite safe since only one bit is needed to convey the acknowledgement information, and this bit can be encoded, for instance, with a $(m, 1)$ majority logic coding scheme where m is fairly large. The resulting large number of parity bits ($m = 6$ or 7, for example) make the probability of error in the ACK / NAK signal negligibly small. Such large number of parity bits can be easily afforded in practice since acknowledgement information is typically a very small fraction of the data to be transmitted.

In the previous two sections, the three most common ARQ schemes were discussed and compared to FEC error control techniques. Each of the two types of error control techniques has its own advantages and drawbacks. In FEC schemes, errors resulting

from channel noise are corrected at the receiver using the 'backup' parity bits introduced in each codeword. However, in particularly noisy or fluctuating channels, the probability of error at the output of the FEC decoder can become unacceptably high. In ARQ systems, on the other hand, the probability of error in the data delivered to the user is fixed, regardless of the state of the channel; however, throughput can be significantly reduced in very noisy channels. Now, the advantages of both techniques can be combined by using *hybrid* ARQ schemes. For instance, in the so called *Type - II hybrid* ARQ scheme [11], an error correcting code is used in conjunction with a regular ARQ protocol. When a data word is received at the channel decoder, error detection is first attempted. If the codeword is found in error, a NAK signal is returned to the transmitter, requesting the transmission of parity bits for error correction. Upon receipt of the parity bits, the decoder attempts to correct the errors in the preceding data packet. If error correction fails, the cycle is repeated. Although hybrid ARQ schemes can give higher throughput than regular ARQ techniques, especially for very noisy channels, they are more complex to implement. Thus, in this research, only the three conventional ARQ protocols are considered. For a more thorough treatment of hybrid ARQ techniques, the reader is referred to [12], [13], [14].

1.4 Fading Channels

Much of communications theory is based on the assumption that the transmission medium or channel is stationary and memoryless; that is, the quality of the channel is constant in time. Although it is widespread, this assumption is quite unrealistic in a large number of communication systems. For instance, fluctuations in weather conditions can temporarily affect the quality of wireless and satellite communication links. Such temporary deterioration of the transmission medium is commonly referred to as *channel fading.*

The fluctuating quality of transmission channels is most obvious in mobile wireless communications. Examples of mobile communication systems include cellular telephones, satellite-to-mobile links, personal communication systems, etc. In such systems the state of the channel changes constantly as the mobile transmitter and/or receiver moves about. For instance, in cellular telephony, the quality of the channel can substantially deteriorate when the mobile unit enters a zone with many buildings, or when the transmitter and receiver grow further apart or become separated by, say,

a hill or a mountain range. Such hindrance to the transmission by a large obstacle is referred to as *shadowing*. Other factors causing channel fading are the well known problems of *multi-path*. Multi-path fading is caused by the destructive interference of two or more electromagnetic waves originating from the same transmitter, but delayed with respect to each other. These delays are caused by the reflection of electromagnetic waves on physical obstacles such as buildings, walls, etc. Thus, multi-path interference is highly dependent on the transmission wavelength and the position of the mobile unit. Multi-path interference can cause severe fading in applications such as indoor communications and cellular telephony in urban areas.

Due to fading, the state of the channel changes with time. The extent with which channel conditions change with time determines the memory of a channel. Indeed, when a vehicle equipped with a cellular telephone is moving, say, straight across the countryside, the state of the channel at any particular time t is very likely to be the same a short time Δt thereafter. In such a case, the channel is said to have high memory. On the other hand, for a vehicle moving rapidly through a large urban area, fading is much faster, due to the quick appearance and disappearance of buildings in the transmission path. In the latter case, the channel is then said to exhibit low memory. More on fading channels and on ways to model them mathematically can be found in Chapter 2.

1.5 In This Thesis...

This section gives a quick overview of the topics to be discussed in this thesis.

As discussed in the abstract, this research is mainly concerned with the analysis of ARQ schemes over non-stationary or fading channels. Thus, in Chapter 2, various ways of modelling fading channels are presented. The Gilbert-Elliott model and its ability to model fading processes is particularly discussed. This model is then extended in a recursive fashion to represent the effects of channel memory on words of data rather than on single data bits. This approach is needed since ARQ systems transmit data in packets or codewords of n bits. Finally, a short review of previous work in the field is also presented.

As discussed in the introduction, an adequate measure of performance in feedback communication systems is throughput efficiency. Chapter 3 presents new techniques

for computing throughput performance for conventional ARQ systems used over the Gilbert-Elliott fading channel, extended to account for word oriented data transmission. For slow fading channels, an approximate transition matrix is developed which gives probabilities of transitions between channel state vectors. Using the properties of this transition matrix, an approximation on throughput efficiency is derived. This chapter also discusses throughput for fast fading channels. Upper and lower bounds on throughput efficiency are derived, which are useful to bound throughput performance for the case of fast fading. These bounds are useful in confirming the approximations on ARQ throughput obtained for the slow channel fading case. Some representative plots showing throughput performance for the three common ARQ protocols are also presented in this chapter. These performance plots are obtained using the throughput evaluation techniques developed in this research. Finally, the effect of channel fading on the performance of ARQ systems is examined, and ways of improving this performance are discussed.

In Chapter 4, frequency-hopped codeword modulation is introduced as a means of improving ARQ performance on fading channels. In frequency hopping ARQ, codewords are transmitted alternatively on m independent frequencies. Expressions for the throughput efficiency of frequency-hopped ARQ systems over slow fading channels are derived. It is shown that, depending on the fading channel characteristics, throughput efficiency can be significantly increased by the use of such frequency hopping scheme.

Finally, Chapter 5 draws some concluding remarks on this research and discusses some areas for further research.

Chapter 2

Fading Channel Modelling

As outlined in the previous sections, this research is concerned mainly in the analysis of ARQ error control systems over fading channels. In this section, we present the various models that can be used to mathematically represent a fading or memory channel. We focus our discussion on the Gilbert-Elliott model which we will use throughout this thesis to describe fading channels.

Since we are interested here in the problem of error control coding, we consider only the so called *coding channel* (see Figure 1.1) which includes the modulator and demodulator (with hard decisions) in addition to the transmission channel. Hence, the input and output to our coding channel, subsequently referred to as the *channel*, are sequences of binary digits or bits. The input digit a_t enters the channel where it is corrupted by noise n_t, giving an output b_t

$$b_t = a_t + n_t$$

where $a_t, b_t, n_t \in \{0, 1\}$ and the above '+' sign is modulo-2 addition. Errors hence occur when the noise digit is $n_t = \varepsilon_t = 1$. ε_t is the error digit occurring at time t in the error sequence $\underline{\varepsilon}$. Note that the noise and error sequences are equivalent stochastic processes.

Since errors are binary digits corrupting the input data stream, the coding channel can be seen as a binary symmetric channel (BSC) with a given bit error rate (BER) p. The BSC bit error rate, also referred to as *crossover probability*, is the probability

$$p = P(b_t = 1 | a_t = 0) = P(b_t = 0 | a_t = 1)$$

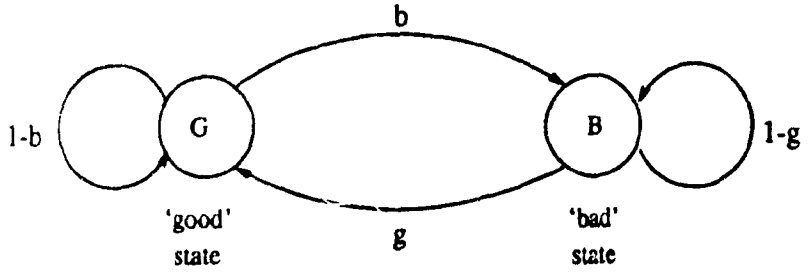

Figure 2.1: **Gilbert's model for memory channels**

2.1 The Gilbert-Elliott Model

As discussed in Section 1.4, errors in real communications channels tend to occur in bursts. This burstiness of the error process, or equivalently of the noise process, implies that the channel has memory. This property was observed by Gilbert on telephone channels. In a paper on the subject [15], Gilbert introduces a two-state Markov chain to model the behaviour of such memory channels. The model proposed by Gilbert assumes the channel is binary symmetric and is in either of two states, a 'bad' state in which the probability of a bit error is large, and a 'good' state in which no errors can occur The transitions between the 'good' and 'bad' state are governed by the Markov chain shown in Figure 2.1.

When in the 'good' state, the channel is error free, whereas in the 'bad' state, errors can occur with probability p. By making the transition probabilities b and g in Figure 2.2 small, the errors generated by the model tend to be clustered in bursts, with error free periods in between the bursts.

While Gilbert's model adequately represents the burst patterns the noise process, it suffers from the fact that when in the 'burst' state, the probability of making an error does not decrease after the channel has been in this state for a significant period of time. Hence, Gilbert's model gives rise to a renewal error process which does not accurately represent real channels. In order to circumvent this problem, Elliott proposes a modification [16] to Gilbert's channel model. In Elliott's proposal, the 'good' state is no longer error free; bit errors can occur in the 'good' state with probability p_0. The 'bad' state still gives rise to errors, with a much higher probability p_1. In other words, Elliott proposes a binary symmetric channel (BSC) with crossover probability p_0 in the 'good' state and p_1 in the 'bad' state. Transitions between the

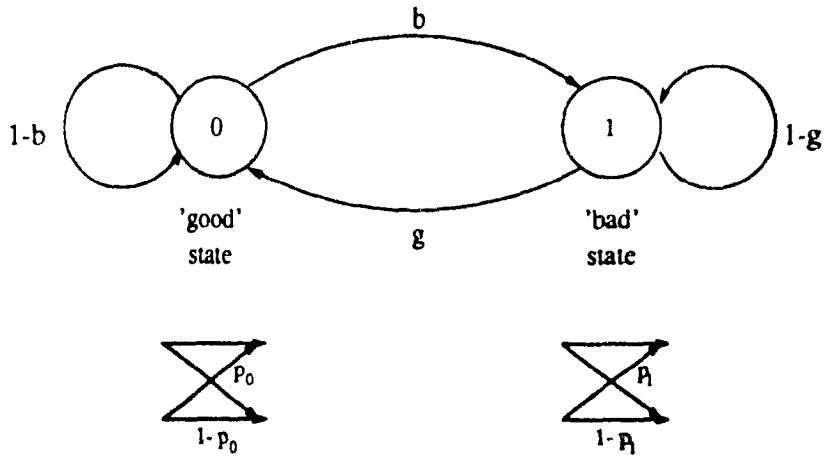

Figure 2.2: **Two-state Gilbert-Elliott Markov model for fading channels**

two states are still Markov as described in the Gilbert model. The resulting two-state Markov model, which we will refer to as the *Gilbert-Elliott model*, is shown in Figure 2.2. This model has the advantage of providing for some background noise when the channel is in the 'good' state, while still giving errors in bursts when the channel is in the 'bad' state In order to illustrate this point, the 'good' and 'bad' states are also referred to in the literature as the 'random-error' and 'burst-error' states, respectively

The provision for errors in the 'good' state makes the Gilbert-Elliott model the generater of a non-renewal error process which was shown to approximate with fair accuracy the behaviour of many communication channels. While other channel models have been proposed over the years (Section 2 4), the Gilbert-Elliott model remains a very popular one for modelling channels with memory, as shown by the numerous analyses carried on this model [18] - [22]. The popularity of the Gilbert-Elliott model is due in large part to its relative simplicity, as compared to other models.

Finally, it must be noted that, in the Gilbert-Elliott model, the Markov chain does not represent the error process *per se*, but rather the channel state process, which itself gives rise to errors with probability p_0 in the 'good' state and p_1 in the 'bad' state. Given an error sequence $\underline{e}$, one cannot reconstitute the history of the channel state Markov process, since there is no way of distinguishing if a '1' digit in the error sequence originated in the 'good' or 'bad' state. Such a model is referred to as a 'unifilar' source of errors [17]. This property of the Gilbert-Elliott model has no consequence on our ensuing analysis, however.

2.2 Parametrization of the Gilbert-Elliott Model

Section 2.1 discussed the basic two-state Gilbert-Elliott model and its applicability for modelling fading channels. Figure 2.2 shows that the Gilbert-Elliott model is completely defined by the parameters b, g, p_0, and p_1. In this section, we investigate the relationship of these parameters to physical fading channel characteristics. Such investigation is valuable in determining the range of model parameters values over which we should focus in any subsequent analysis. Indeed, in Sections 3.2 and 3.3, ARQ system performance for the special cases of both slow and fast fading are analyzed. The following questions arise: For what ranges of the parameters b and g is the channel considered to be a slow fading channel? For what ranges is the channel considered to be a fast fading channel? What model parameters values should be used to describe a typical *land-to-mobile* channel? What are typical model parameters values for *satellite-to-mobile* channels? Such issues are addressed in the present section.

Heuristically speaking, a slow fading channel is one which changes state (from 'good' to 'bad' or from 'bad' to 'good') at a very slow rate. In such channels, the channel state sequence is typically a long string of consecutive zeroes or ones [17]. Such channels are also said to exhibit high memory. Indeed, when long strings of zeroes or ones prevail, the state of the channel at a given time instant $t = t_0$ is highly correlated with the state of the channel at the previous time instant $t = t_0 - 1$. A slow fading process translates into small probabilities of 'good-to-bad' and 'bad-to-good' transitions in the Gilbert-Elliott model. In other words, for slow fading channels or high memory channels, the parameters b and g are typically small. Indeed, one definition of channel memory is given in [20] in terms of the parameters b and g:

$$\mu = 1 - b - g \tag{2.1}$$

The memory μ of the channel increases with decreasing b and g, which is in accordance with the heuristic argument above. The smaller b and g, i.e. the slower the fading process, or equivalently, the burstier the noise process, the larger the memory μ of the channel. Conversely, for fast fading, or equivalently, low memory channels, the parameters b and g are typically large. For such channels, the state of the channel at any given time instant is less dependent on the channel state at another time instant, as compared to slow fading channels. Typical channel state sequences are shown in Figure 2.3 to illustrate slow and fast fading.

(a)

000000000111000000011111110000000000000000111110000

(b)

001001110000110101000110100010100100011001000110010

Figure 2.3: Typical channel state sequences for (a) a slow fading channel, and (b) a fast fading channel

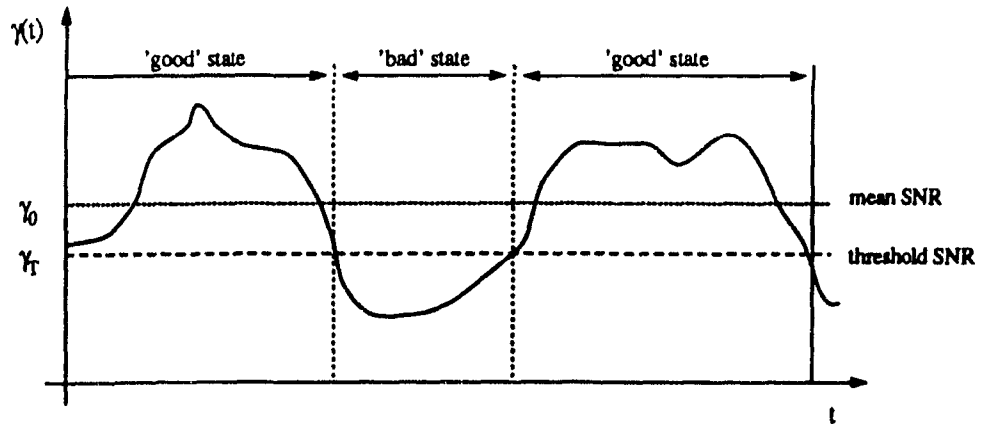

Figure 2.4: SNR as a function of time, threshold SNR and mean SNR

One way of determining the set b, g, p_0, p_1 of Gilbert-Elliott model parameters is proposed by Krishnamurthi and Gupta [21]. The latter show that the BSC crossover probabilities p_0 and p_1 are a function of the mean signal to noise ratio (SNR) γ_0 prevailing over a given fading channel, and of a threshold SNR γ_T. The threshold SNR is the SNR below which the channel is considered to be in the 'bad' state, and vice versa, as shown in Figure 2.4 which illustrates SNR variations with time in a fading channel. Again from the figure, it can be seen that γ_0 is the time average SNR prevailing over the channel. Now, summarizing the results in [21], we have

$$\begin{aligned} p_0 &= \frac{\exp(-\frac{\gamma_T}{2})}{(\gamma_0+2)} \\ p_1 &= \frac{1-\exp[-\gamma_T(\frac{1}{2}+\frac{1}{\gamma_0})]}{(\gamma_0+2)[1-\exp(-\gamma_T/\gamma_0)]}. \end{aligned} \tag{2.2}$$

The probabilities p_0 and p_1 can be seen as the average bit error probabilities when the channel is in the 'good' and 'bad' state respectively. The derivation of p_0 and p_1 was achieved assuming a noncoherent Frequency Shift Keying (FSK) modulation and demodulation scheme is used for channel signalling. Results for coherent FSK, regular Binary Phase Shift Keying (BPSK) and differentially encoded BPSK modulation / demodulation can also be found in [21]. The case of non coherent FSK is retained here as it constitutes the worse case communications scheme.

Krishnamurthi and Gupta have also derived the 'good-to-bad' and 'bad-to-good' transition probabilities (i.e. b and g respectively) from the physical mobile channel characteristics. Their results are summarized as

$$\begin{aligned} b &= \frac{Tvf_c}{c}\sqrt{2\pi\gamma_T/\gamma_0} \\ g &= \frac{Tvf_c}{c}\frac{\sqrt{2\pi\gamma_T/\gamma_0}}{\exp(\gamma_T/\gamma_0)-1}, \end{aligned} \tag{2.3}$$

where T is the data bit period (in seconds), v is the velocity of the mobile radio (in m/s), f_c is the carrier frequency (in Hz), and c is the speed of light. Note here that the faster the vehicle, the larger b and g, and thus, the faster the fading. Similarly, the larger the data rate, i.e. the smaller T, the smaller b and g, and thus, the slower the fading. This is intuitively correct, since the faster data transmission is, the slower will channel fading appear for a given codeword.

Plots of p_0 and p_1 versus mean SNR and threshold SNR are shown in Figures 2.5 and 2.6, respectively. The curves shown are obtained from Equation 2.2 and are

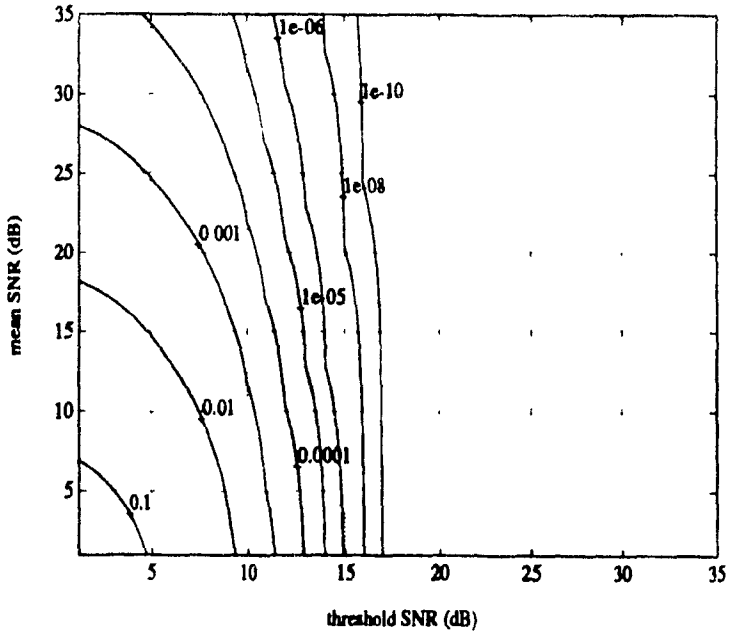

Figure 2.5: **Contour plots of crossover probability p_0 as a function of mean SNR and threshold SNR**

contour lines of p_0 and p_1. For instance, given a mean SNR $\gamma_0 = 20$ dB and threshold SNR $\gamma_T = 12$ dB, the average bit error rate in the 'good' state is $p_0 = 1 \times 10^{-5}$.

Also, contour plots of the state transition probabilities b and g as a function of mean SNR and threshold SNR are shown in Figures 2.7 and 2.8. These plots are obtained directly from Equation 2.3 for a mobile radio transmitting data at a rate of 40.6 kb/s, at carrier frequency $f_c = 1.0$ GHz. The mobile radio is moving at a speed $v = 20$ m/s.

Using the above results, it is possible to obtain a set of Gilbert-Elliott model parameters for a typical mobile radio channel. We again take the example of the above vehicle moving at 20 m/s and transmitting data at a rate of 40.6 kb/s at a carrier frequency $f_c = 1.0$ GHz. We assume a mean SNR $\gamma_0 = 25$ dB, which is reasonable in a cellular telephony environment. We take an arbitrary threshold SNR $\gamma_T = 13$ dB. The resulting Gilbert-Elliott channel model parameters are

$$\begin{aligned} p_0 &\simeq 5 \times 10^{-7} \\ p_1 &\simeq 0.06 \\ b &= 0.001 \end{aligned}$$

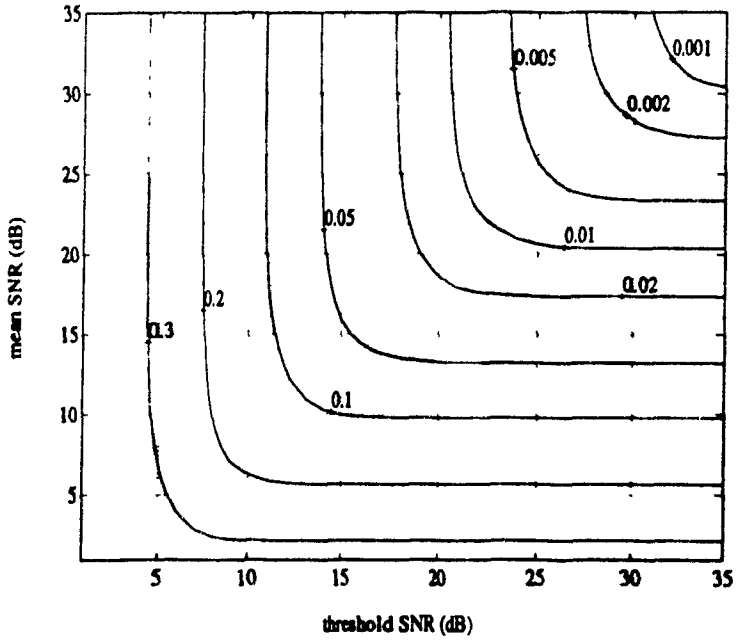

Figure 2.6: Contour plots of crossover probability p_1 as a function of mean SNR and threshold SNR

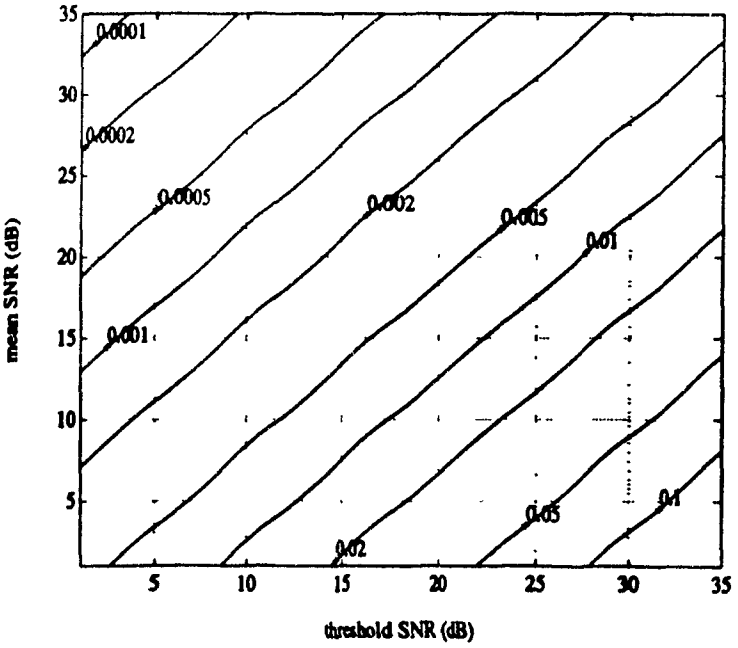

Figure 2.7: Contour plot of 'good-to-bad' transition probability b as a function of mean SNR and threshold SNR

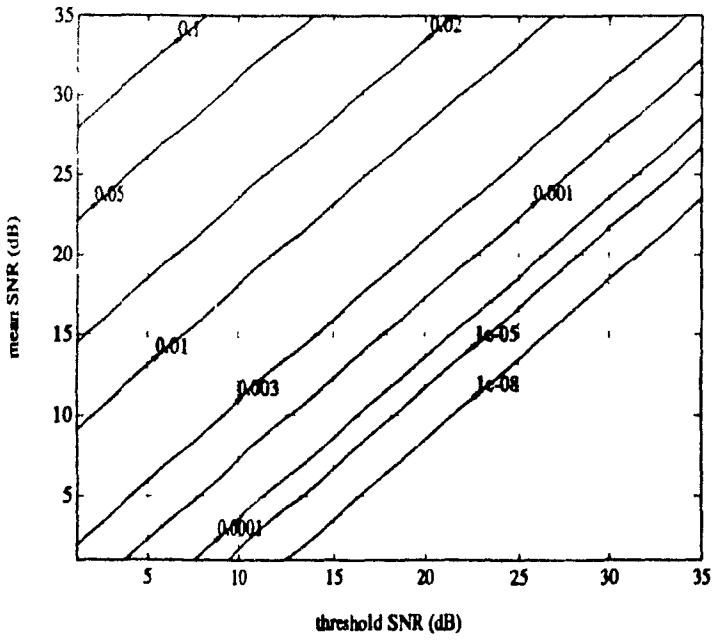

Figure 2.8: **Contour plot of 'bad-to-good' transition probability g as a function of mean SNR and threshold SNR**

	p_0	p_1	b	g
Downtown Munich, $v = 40$ km/h	2.1×10^{-4}	0.317	3.95×10^{-4}	1.05×10^{-4}
Suburban Hamburg, $v = 40$ km/h	3.4×10^{-4}	0.298	2.1×10^{-4}	1.54×10^{-4}
Highway driving, $v = 90$ km/h	1.1×10^{-4}	0.194	2.96×10^{-5}	1.29×10^{-4}

Table 2.1: **Gilbert-Elliott model parameters for satellite-to-mobile links**

$$g = 0.015$$

Other investigations in physical channel modelling and parametrization can be found in the literature. Cygan *et al.* [22] analyze *satellite-to-mobile* links in cities and on highways in Germany. Their findings in terms of the Gilbert-Elliott model parameters are gathered in Table 2.2. The mobile radio used in the experiment transmits at a rate of 1.2 kb/s, at a frequency $f = 1.546$ GHz.

From the observations in this section, one can get a rough idea of the magnitude of the parameters in the Gilbert-Elliott model. For instance, given the power levels and channels used in mobile radio applications, the crossover probabilities p_0 and p_1

are in general of the order

$$p_0 \sim 10^{-\epsilon}$$
$$p_1 \sim 10^{-1}$$

The 'good-to-bad' and 'bad-to-good' transition probabilities b and g are more difficult to quantify; they depend highly on such factors as the data bit rate, the speed of the vehicle, the carrier frequency, the type of channel, etc.

2.3 Extended Gilbert-Elliott Fading Channel Model

The Gilbert-Elliott channel model presented in Section 2.1 describes channel state transitions affecting *consecutive* bits in a data stream. In ARQ and many other systems, however, data is transmitted, not one bit at a time, but, rather, in blocks of n bits. Thus, in order to perform an analysis of ARQ performance, it is necessary to look at transition probabilities between channel states over n-bit codewords. In order to do this, we extend the one-bit binary symmetric channel (BSC) to an n-bit vector channel made up of n BSC's [23]. This n-bit channel corresponds in a natural way to the n-bit transmitted codewords. An n-bit vector channel is needed since, in the analysis of ARQ systems, we are interested not in the probability of one bit error, but in the probability of detecting an error in one entire codeword. Now, since each BSC in the vector channel can be in either of two states, this vector channel can take on 2^n possible states; hence, in order to describe transitions between these 2^n states, a Markov chain with 2^n possible states is necessary. In this section, an ***Extended Gilbert-Elliott*** channel model is derived using the basic two-state Gilbert-Elliott model as a starting point. This extended channel model describes the transition probabilities between n-bit codewords $\underline{S}_i$. The extended channel model is a 2^n-state Markov chain described by a $2^n \times 2^n$ transition matrix denoted by $P^{(n)}$ in what follows. The transition matrix $P^{(n)}$ provides the probabilities of transition between two given n-bit vector channel states $\underline{S}_1$ and $\underline{S}_2$, as illustrated in Figure 2.9.

The $2^n \times 2^n$ transition matrix $P^{(n)}$ is recursively developed from the following 2×2 transition matrix P of the Gilbert-Elliott model:

$$P = \begin{pmatrix} 1-b & b \\ g & 1-g \end{pmatrix} \tag{2.4}$$

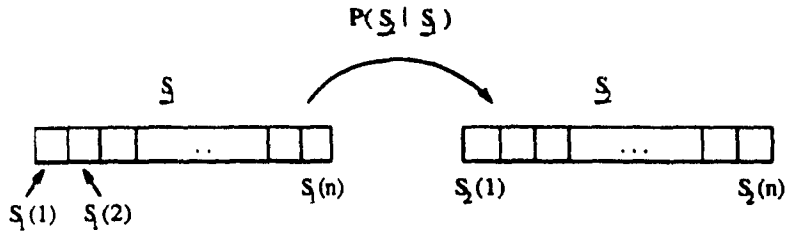

Figure 2.9: **Transitions between two channel state vectors**

In order to show this, we first derive the transition matrix $P^{(2)}$ for a two-bit vector channel. Denoting two consecutive two-bit channel state vectors (CSV's) by $\underline{S}_1 = (S_1(1), S_1(2))$ and $\underline{S}_2 = (S_2(1), S_2(2))$, and using the Markov property, we can write the transition probabilities between $\underline{S}_1$ and $\underline{S}_2$ as

$$\begin{aligned} P(\underline{S}_2|\underline{S}_1) &= P(\underline{S}_2|S_1(2)) \\ &= P(S_2(2)|S_2(1))P(S_2(1)|S_1(2)). \end{aligned}$$

The transition probabilities for all possible vector channel states $\underline{S}_1$ and $\underline{S}_2$ are given below:

$$\begin{aligned} P(\times 0 \rightarrow 00) &= (1-b)^2 & P(\times 1 \rightarrow 00) &= g(1-b) \\ P(\times 0 \rightarrow 01) &= (1-b)b & P(\times 1 \rightarrow 01) &= gb \\ P(\times 0 \rightarrow 10) &= bg & P(\times 1 \rightarrow 10) &= g(1-g) \\ P(\times 0 \rightarrow 11) &= b(1-g) & P(\times 1 \rightarrow 11) &= (1-g)^2 \end{aligned}$$

where X can take on either 0 or 1 without. The corresponding transition matrix is then

$$P^{(2)} = \begin{pmatrix} (1-b)^2 & (1-b)b & bg & b(1-g) \\ g(1-b) & gb & (1-g)g & (1-g)^2 \\ (1-b)^2 & (1-b)b & bg & b(1-g) \\ g(1-b) & gb & (1-g)g & (1-g)^2 \end{pmatrix}.$$

From the above, it can be seen that the two-bit channel state vector (CSV) transition matrix $P^{(2)}$ has only two distinct rows, which we denote by

$$\begin{aligned} R_0^{(2)} &= ((1-b)^2 , (1-b)b , bg , b(1-g)) \\ R_1^{(2)} &= (g(1-b) , gb , (1-g)g , (1-g)^2) \end{aligned}$$

This is because the transition probability $P(\underline{S}_2|\underline{S}_1)$ does not depend on the entire previous channel state vector $\underline{S}_1$), but only on the last bit $S_1(n)$, which can only be 0 or 1.

Now, following the same reasoning, one can show in general that the $(k+1)$-bit CSV transition matrix $P^{(k+1)}$ can be obtained from the k-bit transition matrix $P^{(k)}$ as follows:

$$P^{(k+1)} = \begin{pmatrix} (1-b)R_0^{(k)} & bR_1^{(k)} \\ gR_0^{(k)} & (1-g)R_1^{(k)} \\ \vdots & \vdots \\ gR_0^{(k)} & (1-g)R_1^{(k)} \end{pmatrix}, \tag{2.5}$$

where $R_0^{(k)}$ and $R_1^{(k)}$ are the first and second rows of $P^{(k)}$, respectively. Continuing in this manner, we see that it is possible to recursively obtain the $2^n \times 2^n$ transition matrix $P^{(n)}$, starting from the 2×2 basic Gilbert-Elliott transition matrix P:

$$P^{(n)} = \begin{pmatrix} (1-b)R_0^{(n-1)} & bR_1^{(n-1)} \\ gR_0^{(n-1)} & (1-g)R_1^{(n-1)} \\ \vdots & \vdots \\ gR_0^{(n-1)} & (1-g)R_1^{(n-1)} \end{pmatrix} \tag{2.6}$$

Note that the size of the transition matrix doubles at each step of the recursive procedure. Starting with the 2×2 matrix P, we obtain the $2^n \times 2^n$ matrix $P^{(n)}$ in $n-1$ steps.

Now, faulty codeword retransmissions are separated by δ bits of idle time, due to the channel round-trip delay required for the negative acknowledgement (NAK) signal to reach the transmitter (see Section 1.3). It is, therefore, necessary to modify the above transition matrix $P^{(n)}$ to account for this idle time of $\delta = R\tau = M - 1$ bits. In order to obtain this modified transition matrix, which we denote by $\tilde{P}^{(n)}$, we write the transition probabilities between the two n-bit channel CSV's $\underline{S}_1$ and $\underline{S}_2$ as follows:

$$\begin{aligned} P(\underline{S}_2|\underline{S}_1) &= P(S_2(n)|S_2(n-1))P(S_2(n-1)|S_2(n-2))\cdots \\ &\quad \cdots P(S_2(2)|S_2(1))P^{(M)}(S_2(1)|S_1(n)). \end{aligned} \tag{2.7}$$

Here, $P^{(M)}(S_2(1)|S_1(n))$ is an M^{th} order transition probability described by the M^{th} order transition matrix P^M of the two-state Gilbert-Elliott channel model:

$$P^M = \begin{pmatrix} P_{00}^{(M)} & P_{01}^{(M)} \\ P_{10}^{(M)} & P_{11}^{(M)} \end{pmatrix}. \tag{2.8}$$

The expression for the probability of transition $P(\underline{S}_2|\underline{S}_1)$ between states $\underline{S}_1$ and $\underline{S}_2$, taking into account channel idle time, is identical to that for $P(\underline{S}_2|\underline{S}_1)$, except that

the first order probability $P(S_2(1)|S_1(n))$ is replaced by the M^{th} order probability $P^{(M)}(S_2(1)|S_1(n))$. In matrix form, this translates into changing the form of 2.6 to

$$\tilde{P}^{(n)} = \begin{pmatrix} P_{00}^{(M)} R_0^{(n-1)} & P_{01}^{(M)} R_1^{(n-1)} \\ P_{10}^{(M)} R_0^{(n-1)} & P_{11}^{(M)} R_1^{(n-1)} \\ \vdots & \vdots \\ P_{10}^{(M)} R_0^{(n-1)} & P_{11}^{(M)} R_1^{(n-1)} \end{pmatrix}. \tag{2.9}$$

where $R_0^{(n-1)}$ and $R_1^{(n-1)}$ are still the first and second rows of $P^{(n-1)}$, respectively. The matrix $P^{(n-1)}$ is the unmodified transition matrix for the $n-1$-bit CSV. This matrix is recursively obtained starting from P as shown in (2.5).

A summary of the procedure described above for obtaining matrix $\tilde{P}^{(n)}$ is given below:

Step 1: Initialization

We start with the basic Gilbert-Elliott model transition matrix:

$$P = \begin{pmatrix} 1-b & b \\ g & 1-g \end{pmatrix} \equiv P^{(0)},$$

Step 2: For $k = 1, 2, \cdots, n-1$,

Apply the recursion:

$$P^{(k+1)} = \begin{pmatrix} R_0^{(k+1)} \\ R_1^{(k+1)} \\ \vdots \\ R_1^{(k+1)} \end{pmatrix} = \begin{pmatrix} (1-b)R_0^{(k)} & bR_1^{(k)} \\ gR_0^{(k)} & (1-g)R_1^{(k)} \\ \vdots & \vdots \\ gR_0^{(k)} & (1-g)R_1^{(k)} \end{pmatrix}.$$

Step 3: M^{th} order one-bit transition probabilities

Compute:

$$P^M = \begin{pmatrix} P_{00}^{(M)} & P_{01}^{(M)} \\ P_{10}^{(M)} & P_{11}^{(M)} \end{pmatrix}$$

This is necessary since gaps of $\delta = M-1$ bits separate consecutive codewords.

Step 4: Transition matrix for the n-bit codeword

$$\tilde{P}^{(n)} = \begin{pmatrix} P_{00}^{(M)} R_0^{(n-1)} & P_{01}^{(M)} R_1^{(n-1)} \\ P_{10}^{(M)} R_0^{(n-1)} & P_{11}^{(M)} R_1^{(n-1)} \\ \vdots & \vdots \\ P_{10}^{(M)} R_0^{(n-1)} & P_{11}^{(M)} R_1^{(n-1)} \end{pmatrix}$$

Note that $\tilde{P}^{(n)}$ is the transition matrix for the n-bit CSV, modified to account for idle time between retransmissions.

Again, extending the 2-state Gilbert-Elliott channel model to a 2^n-state Markov chain with transition matrix $\tilde{P}^{(n)}$ is necessary in order to represent the effects of channel memory on each codeword, and the memory induced between consecutive codewords.

2.4 Memory Channels: Other Models

Gilbert's and Elliott's models were among the first attempts at representing physical fading channel behaviour. A number of other models were developed since. Channel models can be readily classified in one of two distinct categories. These are the *generative models*, which include the Gilbert and Gilbert-Elliott model, and the *descriptive models* [17]. Generative models are often Markov chains, with a finite or infinite number of states. These states usually map into error bits, and thus state transition progressions generate error sequences. A physical channel can hence be modelled by appropriately selecting the model's Markov chain parameters. Having obtained a generative model, one can then analyse various error control schemes and obtain statistics on their performance. On the other hand, descriptive models base themselves on various statistics obtained from the examination of a real channel. From a descriptive model, statistics can be derived that indicate the performance of error control schemes.

The Gilbert and Gilbert-Elliott models were shown in Section 2.1 to be unifilar sources of errors; channel state sequences cannot be inferred from an observed error sequence. This property makes these unifilar models difficult to parametrize, i.e. it is difficult to obtain parameters for the Markov models given sequences of errors observed in a physical channel. In order to bypass this difficulty, Berkovits and Cohen [24] propose a modification to the Gilbert model. A third error producing state is

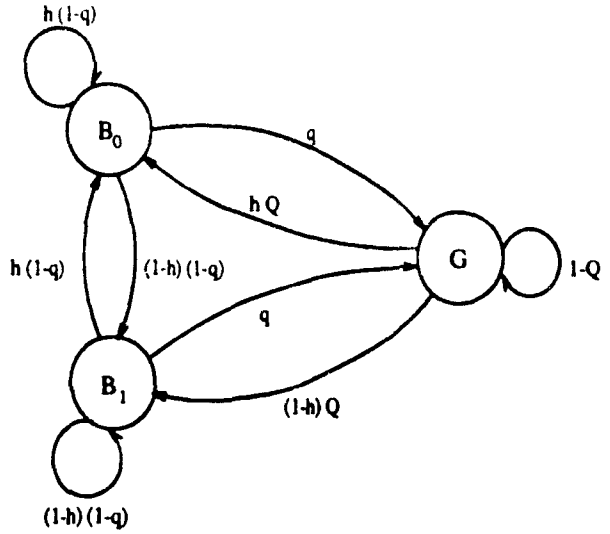

Figure 2.10: **The Berkovits & Cohen 3-state model**

included in the Markov chain, as shown in Figure 2.10 As in the Gilbert model, no errors can occur in the 'good' state G. However, the 'bad' state is split in two, B_0 and B_1, with errors allowed to occur only in state B_1. The advantage of the Berkovits & Cohen generative model over Gilbert's model lies in the fact that the parameters q, Q and h in Figure 2.10 can be easily derived from a sample error sequence.

Another Markov model whose parameters are easily obtained from observed error sequences is proposed by McCullough [25]. The McCullough channel model, shown in Figure 2.11, admits errors in both the 'good' and 'bad' states. In Figure 2.11, the variable Σ_i represents the state of the channel at time i: $\Sigma_i = 0$, with probability $1 - P_i$, for the 'good' state and $\Sigma = 1$, with probability P_i for the 'bad' state. Z is the noise bit generated by the model; for $Z = 0$, no error occurs, and for $Z = 1$, an error bit is produced. The model parameters p_{ij}, q_{ij} and P_i are easily deduced from sample error sequences, where

$$\begin{aligned} p_{ij} &= P(\Sigma_n = j | \Sigma_{n-1} = i, Z_{n-1} = 0) \\ q_{ij} &= P(\Sigma_n = j | \Sigma_{n-1} = i, Z_{n-1} = 1) \\ P_i &= P(Z_n = 1 | \Sigma_n = i) \end{aligned}$$

From the above definitions, P_0 is the bit error rate in the 'good' state and P_1 is the BER in the 'bad' state. Also note that the McCullough model, also known as the *bit regenerative model*, reduces to the Gilbert-Elliott model for $p_{ij} = q_{ij}$.

A fundamentally different generative channel model is proposed by Fritchman [26].

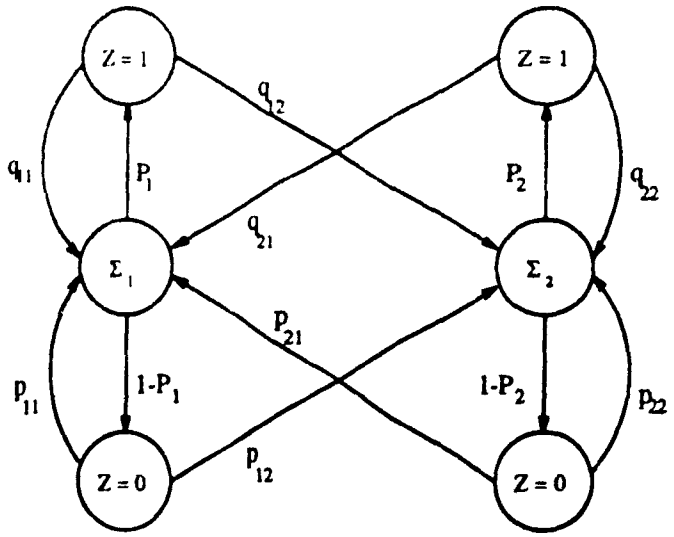

Figure 2.11: **McCullough's bit regenerative channel model**

This model, shown in Figure 2.12, is a Markov chain with N states. The state space of the model is partitioned into two sets, the set A composed of k error-free states, and the set B with $N-k$ error states. A function ϕ mapping the state space into an error sequence is defined by

$$\phi(i) = \begin{cases} 0 & \text{for } i \in A \\ 1 & \text{for } i \in B \end{cases}$$

Using the above definition and the N-state Markov chain, Fritchman derives the *error gap distribution* (EGD) from which performance measures of error-control schemes can be derived. The EGD is a statistic frequently used to describe burst-noise channels. The EGD is the probability distribution of gaps between consecutive bursts of error bits. From the EGD, the $P(m,n)$ distribution, which is the probability of having m errors in an n-bit codeword, can be derived. The $P(m,n)$ distribution is an important statistic for the performance evaluation of FEC codes.

One of the early descriptive channel models was introduced by Berger and Mandelbrot [27] to fit experimental data in telephone circuits. Berger and Mandelbrot suggest that the so called *Pareto function* is adequate in modelling the error gap distribution in the telephone circuits error data:

$$P(0^j|1) = 1/j^{\theta}$$

$P(0^j|1)$ is the EGD, ie. the probability of having j error-free bits after an error has occurred. θ is a parameter obtained from the error sequence data to be modelled.

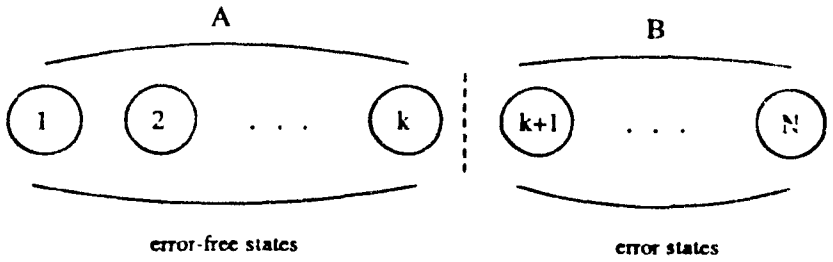

Figure 2.12: **Fritchman's partitioned state space**

Note that the Pareto function above models a renewal error process; the probability of having j zeroes is independent of how long the preceding error burst was. Other models also exist which try to represent error gap distributions in experimental data [28] [29] [30]. From the EGD, the probability distribution $P(m,n)$ is again easily derived.

As shown in this section, a number of models have been developed to represent fading or burst-error channels. Some of the models, such as the one put forth by Fritchman, are fairly complex but reproduce fading channel behaviour accurately; whereas other models, such as Berger and Mandelbrot's are quite simple, but do not faithfully represent real channels. There is certainly a tradeoff at play here between model complexity and accuracy. A good comparison of the various channel models is difficult and has not been attempted to this day. In this research, we adopt the Gilbert-Elliott model as it represents a good compromise between complexity and accuracy in channel representation. Also, as outlined in Section 2.1, the Gilbert-Elliott model is widely used in the literature to represent burst-error channels.

2.5 Previous Work in Feedback Communications on Fading Channels

In this section, a review of some of the previous work in ARQ error-control over fading channels is presented and contrasted to the study conducted in this thesis.

One of the firsts attempts at designing ARQ schemes for high error rate and bursty channels was undertaken by Sastry. In his paper [33], Sastry suggests the transmission of M identical codewords each time a retransmission is requested. This technique is well suited for Stop-and-Wait schemes, since the channel idle time is then

exploited to transmit copies of the original codeword. This technique is shown to yield a substantial improvement in throughput efficiency over the regular SW protocol for channels with very high bit error rates ($P_e > 10^{-2}$).

In 1976, Arazi proposes a Stop-and-Wait ARQ scheme with partial codeword retransmission [34]. Arazi suggests the interleaving of parity bits evenly within each codeword. In doing so, one can locate single bursts of errors that may occur within a codeword. Consequently, only that part of the codeword which is found in error need to be retransmitted.

In the same vein as Arazi's work, Turney proposes a new ARQ scheme which is roughly a hybrid between the Stop-and-Wait and the Selective Repeat protocols [35]. K-bit long packets of data are sent in a Stop-and-Wait fashion. These K bits are partitioned into n 'parts'. Each part is checked for errors, and only those part(s) found in error are retransmitted. Turney shows that such a scheme can provide up to 40% throughput improvement over regular Stop-and-Wait systems for mobile radio channels with BER $P_e = 10^{-3}$. However, in his analysis, Turney assumes that, for sufficiently long codewords, errors in consecutive codewords are uncorrelated. This assumption is somewhat questionable, especially for slow fading channels.

In an effort to improve ARQ throughput in channels with varying BER, Martins and De Carvalho Alves propose an ARQ scheme with adaptive codeword length [36]. For instance, for the Stop-and-Wait protocol, there exists an optimum codeword length for maximum throughput efficiency. This optimum length depends mainly on the BER of the transmission channel. Hence, Martins and De Carvalho Alves show that significant throughput improvement can be achieved by making the codeword length variable in the SW and GBN ARQ protocols. Although this technique seems to be well suited for slow fading channels, it can be very complex to implement. Indeed, the codeword length to be used at any given time depends on the detected bit error rate; a potential for conflict exists, for instance, if the transmitter and receiver detect different error rates.

Another analysis of the performance of ARQ systems over memory channels was undertaken by Fujiwara *et al.* [19]. The analysis describes the performance of conventional and hybrid Go-Back-N ARQ schemes over both memoryless and memory channels, with channel memory modelled using the two-state Gilbert Markov chain. In their paper [19], Fujiwara *et al.* exhibit performance curves of P_{be}, the 'bit error

rate after decoding', as a function of codeword length n. These curves show that Go-Back-N ARQ error control schemes are superior to their Forward Error Correction (FEC) counterparts, especially for memory channels.

In a paper published in 1984 [37], Comroe and Costello examine the performance of various ARQ protocols over land mobile radio channels. The Stop-and-Wait, Selective Repeat and Type - II hybrid Selective Repeat ARQ techniques are examined in this work, and the expected number of transmissions is used to measure the performance of these ARQ schemes. As expected, the Type - II hybrid schemes are found to be the most efficient, especially in high error rate channels. Comroe and Costello use a non standard channel model which assumes that a particular codeword is in error when the received signal strength falls below a given threshold during the transmission of the codeword. The analysis developed in this work gives valuable first insight on the performance of ARQ schemes over mobile radio channels corrupted by multipath fading.

Other non-standard fading channels have been used in the literature to analyse feedback communication techniques. For instance, Leung, Kikumoto and Sorensen [38] as well as Towsley [39] assume the error process (not the channel state process) is Markov in their analysis of the Go-Back-N ARQ protocol over channels with memory.

More recently, the performance of hybrid ARQ techniques over Gilbert-Elliott channels was analysed [18], [40]. Deng and Costello analyse a scheme based on Type - II hybrid ARQ [18]. The analysis is achieved assuming a non-stationary channel, modelled using the Gilbert-Elliott Markov chain. In this work, the authors assume the channel changes state only at the boundary between codewords. In other words, codewords are assumed to be either entirely in the 'good' state or entirely in the 'bad' state. In Lugand's work [40], convolutional codes and Viterbi decoding are used (instead of block codes) in combination with ARQ error control, in a Type - II hybrid fashion. This work also assumes the channel does not change state throughout any given data packet.

Chapter 3

ARQ Throughput for Fading Channels

This chapter discusses ways of obtaining ARQ throughput performance for data transmission over a fading channel. The first section describes a procedure for computing ARQ throughput from the Extended Gilbert-Elliott model transition probabilities and the crossover error probabilities p_0 and p_1. Although this procedure yields exact results, it is computationally intensive and impractical for large codewords. This problem can be circumvented, however, by considering the cases of slow fading in Section 3.2 and fast fading in Section 3.3.

3.1 Exact Throughput Computation

As discussed in Chapter 1, three principal ARQ protocols are employed for data transmission. These are the 'Stop-and-Wait' (SW) protocol, the 'Go-Back-N' (GBN) protocol and the 'Selective Repeat' (SR) protocol [5]. It was also shown in Chapter 1 that throughput is used as a measure of performance in feedback communication systems. The throughput expressions for the three ARQ protocols are repeated here for convenience.

$$\eta_{SW} = \frac{k}{n + R\tau} \frac{1}{E[T]} \tag{3.1}$$

$$\eta_{GBN} = \frac{k}{n} \frac{1}{NE[T] - (N-1)} \tag{3.2}$$

$$\eta_{SR} = \frac{k}{n}\frac{1}{E[T]} \tag{3.3}$$

Again, R is the transmission rate, τ the round trip channel delay time, k/n the code rate, and T is a random variable representing the number of transmissions of the same codeword required to get the codeword correctly across the channel. N is the idle time due to round-trip channel delay, expressed in number of codewords. It is important to note here that all three throughput expressions are functions of the average number of transmissions $E[T]$. Thus, the problem of computing throughput performance for any ARQ scheme reduces to that of computing the average number of transmissions $E[T]$. Now, in order to compute $E[T]$, we define an indicator random variable A_i as follows:

$$A_i = \begin{cases} 1 & \text{, if an error is detected in the } i^{th} \text{ transmission} \\ 0 & \text{, otherwise} \end{cases} \tag{3.4}$$

Using this definition, the number of transmissions T can be written as:

$$\begin{aligned} T &= 1 + A_1 + A_1A_2 + A_1A_2A_3 + \cdots \\ &= 1 + \sum_{l=1}^{\infty}\prod_{i=1}^{l} A_i, \end{aligned} \tag{3.5}$$

and the average number of transmissions is then:

$$\begin{aligned} E[T] &= 1 + \sum_{l=1}^{\infty} E[\prod_{i=1}^{l} A_i] \\ &= 1 + \sum_{l=1}^{\infty} P(A_1 = 1, A_2 = 1, \cdots, A_l = 1) \\ &= 1 + \sum_{l=1}^{\infty} P(A^l = I^l), \end{aligned} \tag{3.6}$$

where $A^l = (A_1, A_2, \ldots, A_l)$ and $I^l = (1, 1, \ldots, 1)$. Hence $P(A^l = I^l)$ is the joint probability of detecting an error in l consecutive codeword transmissions. The expression in 3.6 is clearly an infinite series expansion. The series converges, however, since $P(A^{l+1} = I^{l+1}) < P(A^l = I^l)$, ie. the probability of detecting an error in $l+1$ consecutive codewords is strictly smaller than that of detecting errors in only l consecutive codewords.

From the above series expansion, it can be seen that computing ARQ throughput reduces to computing the joint probabilities $P(A^l = I^l)$, $l = 1, 2, \cdots$. Since the series in Equation 3.6 converges, the joint probabilities $P(A^l = I^l)$ become negligibly small

for sufficiently large l. The infinite series can thus be truncated at some value of l for computational purposes

The next step in obtaining an expression for ARQ throughput is to write the joint probabilities $P(A^l = I^l)$ as conditioned on the channel state $\underline{S}^l$. Using Bayes' law of probability, $P(A^l = I^l)$ can be rewritten as

$$P(A^l = I^l) = \sum_{\underline{S}^l} P(A^l = I^l|\underline{S}^l)P(\underline{S}^l), \tag{3.7}$$

where $\underline{S}^l = (\underline{S}_1, \underline{S}_2, \ldots, \underline{S}_l)$ is a 'compound' channel state vector (CSV) made up of the concatenations of l consecutive n-bit CSV's $\underline{S}_i$. The events of detecting an error in the i^{th} transmission of a codeword ($i = 1, 2, \cdots, l$) are independent of each other when conditioned on the CSV. This stems from the Gilbert-Elliott model which stipulates that the channel, in any given state, is discrete, binary symmetric and memoryless (the channel memory is represented by the transition probabilities between states. Hence, the conditional probabilities in Equation 3.7 can be rewritten as

$$P(A^l = I^l|\underline{S}^l) = \prod_{i=1}^{l} P(A_i = 1|\underline{S}_i).$$

Also, using the Markov property, $P(\underline{S}^l)$ can be written as:

$$P(\underline{S}^l) = P(\underline{S}_l|\underline{S}_{l-1})P(\underline{S}_{l-1}|\underline{S}_{l-2})\cdots P(\underline{S}_2|\underline{S}_1)P(\underline{S}_1).$$

Grouping the above two expressions, one gets

$$\begin{aligned} P(A^l = I^l) &= \sum_{\underline{S}_1}\sum_{\underline{S}_2}\cdots\sum_{\underline{S}_l} P(A_1 = 1|\underline{S}_1)P(\underline{S}_1)P(A_2 = 1|\underline{S}_2)P(\underline{S}_2|\underline{S}_1)\cdots \\ &\quad \cdots P(A_l = 1|\underline{S}_l)P(\underline{S}_l|\underline{S}_{l-1}) \\ &= \sum_{\underline{S}_1} P(A_1 = 1|\underline{S}_1)P(\underline{S}_1)\sum_{\underline{S}_2} P(A_2 = 1|\underline{S}_2)P(\underline{S}_2|\underline{S}_1)\cdots \\ &\quad \cdots\sum_{\underline{S}_l} P(A_l = 1|\underline{S}_l)P(\underline{S}_l|\underline{S}_{l-1}) \end{aligned}$$

Introducing the variables $\beta_k, k = 1, 2, \ldots, l$, one can calculate $P(A^l = I^l)$ by using a backward recursion as follows:

$$\begin{aligned} \beta_l(\underline{S}_{l-1}) &= \sum_{\underline{S}_l} P(A_l = 1|S_l)P(\underline{S}_l|\underline{S}_{l-1}) \\ \beta_{l-1}(\underline{S}_{l-2}) &= \sum_{\underline{S}_{l-1}} \beta_l(\underline{S}_{l-1})P(A_{l-1} = 1|\underline{S}_{l-1})P(\underline{S}_{l-1}|\underline{S}_{l-2}) \end{aligned}$$

$$\vdots$$

$$\beta_k(\underline{S}_{k-1}) = \sum_{\underline{S}_k} \beta_{k+1}(\underline{S}_k) P(A_k = 1|\underline{S}_k) P(\underline{S}_k|\underline{S}_{k-1})$$

$$\vdots$$

$$\beta_1(\underline{S}_0) = \sum_{\underline{S}_1} \beta_2(\underline{S}_1) P(A_1 = 1|\underline{S}_1) P(\underline{S}_1|S_0(n))$$

$$\begin{aligned} P(A^l = I^l) &= \sum_{S_0(n)} \beta_1(S_0(n)) P(S_0(n)) \\ &= \beta_1(S_0(n) = 0) P(S_0(n) = 0) + \beta_1(S_0(n) = 1) P(S_0(n) = 1) \end{aligned} \quad (3.8)$$

This type of recursive procedure for computing the probability of a joint event is also used in Hidden Markov models theory [42]. A new state $S_0(n)$ is introduced here to represent the initial state of the Markov chain. Assuming the Markov chain is stationary - which is a safe assumption given the characteristics of the physical channel - the probabilities $P(S_0(n) = 0)$ and $P(S_0(n) = 1)$ are then the stationary probabilities of the channel being in the 'good' and 'bad' state respectively. From the basic Gilbert-Elliott model, these stationary probabilities are [41]:

$$P(S_0(n) = 0) = \frac{g}{b+g}, \qquad P(S_0(n) = 1) = \frac{b}{b+g}. \quad (3.9)$$

Using the above stationary probabilities, the joint probability $P(A^l = I^l)$ can be rewritten as

$$P(A^l = I^l) = \beta_1^0 \frac{g}{b+g} + \beta_1^1 \frac{b}{b+g}. \quad (3.10)$$

We have introduced here the notation

$$\beta_k(S_{k-1}(n)) = \begin{cases} \beta_k^0 & \text{, for } S_{k-1}(n) = 0 \\ \beta_k^1 & \text{, for } S_{k-1}(n) = 1 \end{cases} \quad (3.11)$$

Note that, in the above recursion, $\beta_k(\underline{S}_{k-1})$ is a function of $\underline{S}_{k-1}$ only since

$$\beta_k(\underline{S}_{k-1}) = \sum_{\underline{S}_k} \beta_{k+1}(\underline{S}_k) P(A_k = 1|\underline{S}_k) P(\underline{S}_k|\underline{S}_{k-1})$$

is an averaging operation over all possible states $\underline{S}_k$ conditioned on $\underline{S}_{k-1}$. Furthermore, since $P(\underline{S}_k|\underline{S}_{k-1})$ depends on $S_{k-1}(n)$ only, due to the Markov property, one can rewrite $\beta_k(\underline{S}_{k-1})$ as follows.

$$\beta_k(S_{k-1}(n)) = \sum_{\underline{S}_k} \beta_{k+1}(\underline{S}_k) P(A_k = 1|\underline{S}_k) P(\underline{S}_k|S_{k-1}(n)). \quad (3.12)$$

Writing β_k in this fashion, one can see that $\beta_k(\underline{S}_{k-1})$ can only take on two possible values β_k^0 and β_k^1, as defined in 3.11.

Now, each of the β_k's is a function of β_{k+1}, $P(A_k = 1|\underline{S}_k)$ and $P(\underline{S}_k|S_{k-1}(n))$, as can be seen from Equation 3.12 above. The probabilities $P(\underline{S}_k|S_{k-1}(n))$ are obtained from the transition matrix $\tilde{P}^{(n)}$ for the Extended Gilbert-Elliott channel model, as shown in Section 2.3. $P(A_k = 1|\underline{S}_k)$ is the probability of detecting an error while the channel is in state $\underline{S}_k$. For a memoryless binary symmetric channel (BSC) with error rate p, the probability of having one or more bit errors in an n-bit codeword is

$$\begin{aligned} P(error) &= P_e = 1 - P(no\ errors) \\ &= 1 - (1-p)^n. \end{aligned} \tag{3.13}$$

For the Gilbert-Elliott model, the channel is also binary symmetric with error rate p_0 when in the 'good' state, and p_1 when in the 'bad' state. When conditioned on a specific CSV, the probability of codeword error is then simply

$$P(error\,|\underline{S}_k) = 1 - (1-p_1)^{N_1^{(n)}(k)}(1-p_0)^{N_0^{(n)}(k)}, \tag{3.14}$$

where $N_1^{(n)}(k)$ and $N_0^{(n)}(k)$ are the number of bits in the 'bad' and 'good' states respectively, in an n-bit codeword. Now, assuming we use a good error detecting code, the probability of undetected error is very small compared to that of detecting an error [5], ie. $P_u \ll P_d$. Also, since $P_e = P_u + P_d$, we have that $P_d \simeq P_e$ for a good code, and hence the probability of detecting an error given that the channel is in state $\underline{S}_k$ is

$$P(A_k = 1|\underline{S}_k) \simeq 1 - (1-p_1)^{N_1^{(n)}(k)}(1-p_0)^{N_0^{(n)}(k)}. \tag{3.15}$$

In summary, in order to compute $P(A^l = I^l)$, one must first obtain the quantities $\beta_l^0, \beta_l^1, \beta_{l-1}^0, \beta_{l-1}^1, \ldots, \beta_1^0, \beta_1^1$ in that order, using the recursion of Equation 3.8. Once β_1^0 and β_1^1 are obtained, the joint probability $P(A^l = I^l)$ is then simply as given in Equation 3.10.

Now, the number of computations required to obtain $P(A^l = I^l)$ can be very large. Each of the β_k^q, $q = 0, 1$, is a sum with 2^n terms, each term being a product of three factors. Thus, computing each β_k^q requires approximately 2^n additions and 2×2^n multiplications. Hence, to compute $P(A^l = I^l)$, $2l2^n$ additions and $4l2^n$ multiplications are needed. Clearly, the number of computations grows exponentially with n, and thus, for large n ($n \geq 40$), the computing time required becomes impractically large. Alternate ways of computing $P(A^l = I^l)$ must then be sought.

3.2 Throughput for Slow Fading Channels

In the previous section, it was shown that since the Extended Gilbert-Elliott model Markov chain has 2^n distinct states, the number of computations required to obtain thr ughput is also proportional to 2^n. the probability of having several 'good-to-bad' or 'bad-to-good' state transitions over one codeword is very small, and can be safely assumed to be zero for computational purposes. Thus, the state transition probability matrix for such slow fading channel can be approximated by an equivalent sparse matrix. In doing so, the number of non-zero elements in the transition matrix is drastically reduced, and becomes linear with codeword length n, instead of exponential.

In this section, we show how for such slow fading channels, ie. for small values of the channel parameters b and g and small codeword length n, an approximate Markov model can be found which has only $n+1$ non-zero elements in each row of the transition matrix $\tilde{P}^{(n)}$. Consequently, the number of computations required in order to obtain ARQ throughput is reduced. Using this sparse matrix approximation allows the derivation of closed form expressions for ARQ throughput performance.

3.2.1 Channel State Vector Transition Probabilities

The approximate channel Markov model and its corresponding transition matrix can be obtained by closely examining the channel state stochastic process affecting data bits in a codeword. We assume data is transmitted in codewords of n bits. We take two consecutive codewords transmitted over a vector channel having CSV's $\underline{S}_k$ and $\underline{S}_{k+1}$, as shown in Figure 3.1. The probability P_{ij} shown in the figure $(i,j=0,1)$ is an element of the two-state Gilbert-Elliott transition matrix

$$P = \begin{pmatrix} 1-b & b \\ g & 1-g \end{pmatrix} \tag{3.16}$$

Looking at the diagram of Figure 3.1, the transition probability between the two consecutive CSV's can be written as

$$P(\underline{S}_{k+1}|\underline{S}_k) = P_{00}^{q_{00}} P_{11}^{q_{11}} P_{01}^{q_{01}} P_{10}^{q_{10}}, \tag{3.17}$$

where q_{ij} is the number of transitions between states i and j within the $\underline{S}_{k+1}$ CSV and channel state bit $S_k(n)$. It is assumed for the time being that there is no channel idle

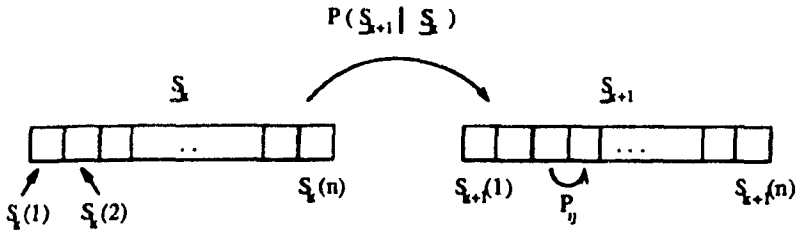

Figure 3.1: **Transitions between two consecutive CSV's**

time ($\delta = 0$) between transmissions of two consecutive codewords. Channel round trip delay will be considered at a later point in this analysis.

For small channel parameters b and g, that is for small P_{01} and P_{10}, the probability $P(\underline{S}_{k+1}|\underline{S}_k)$ in Equation 3.17 becomes negligibly small with increasing q_{10} and q_{01}. Thus, in the transition matrix $P^{(n)}$ of the Extended Gilbert-Elliott channel model, the elements for which $q_{01} + q_{10} \geq 2$ are much smaller in value than those for which $q_{01} + q_{10} \leq 1$. This is true for small b and g, of the order $b, g \leq 0.01$. Thus, for such slow fading channels, one can make a first order approximation of the elements of $P^{(n)}$, or in other words, set to zero all probabilities of state transitions $\underline{S}_k \rightarrow \underline{S}_{k+1}$ having $q_{01}+q_{10} > 1$. In doing so, many of the elements of transition matrix $P^{(n)}$ become zero, making $P^{(n)}$ a sparse matrix, which is easier to deal with than the original matrix with 2^n elements per row. The only non-zero elements in this sparse matrix are those corresponding to transition probabilities describing CSV transitions in which one or less 'good-to-bad' and 'bad-to-good' state transition occurs. For instance, the following are examples of CSV transitions having non-zero probabilities:

$$\begin{aligned} \times \times \cdots \times 0 &\longrightarrow 000111\cdots111 \\ \times \times \cdots \times 1 &\longrightarrow 110000\cdots000 \\ \times \times \cdots \times 0 &\longrightarrow 000000\cdots000 \end{aligned}$$

Also, looking at the above bit pattern, one can see that CSV transition probabilities are non-zero if

$$\begin{aligned} \underline{S}_k &= 2^m - 1, \qquad m = 1, 2, \cdots, n \\ \text{or} \quad \underline{S}_k &= 2^n - 2^m, \qquad m = 1, 2, \cdots, n \end{aligned} \tag{3.18}$$

From Equations 3.17 and 3.18, the non-zero transition probabilities are thus of the form

$$P(\underline{S}_{k+1}|\underline{S}_k) = P_{11}^{W_H(\underline{S}_{k+1})+W_H(S_k(n))-1} P_{00}^{n-W_H(\underline{S}_{k+1})-W_H(S_k(n))} P_{ij} \tag{3.19}$$

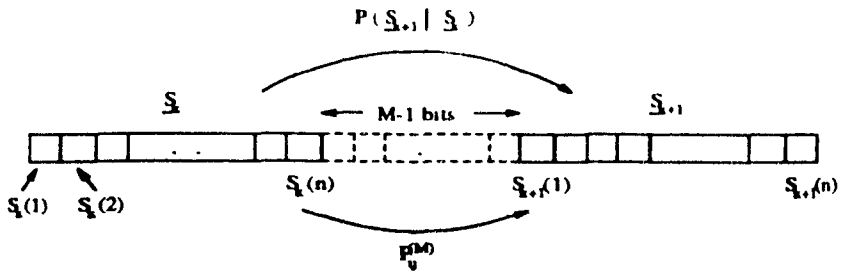

Figure 3.2: **Transitions between two consecutive CSV's with channel round-trip delay δ**

where W_H is the Hamming weight of a sequence of bits. The Hamming weight is by definition [5] the number of '1' bits in a given binary sequence. P_{ij} is the probability of the $0 \rightarrow 1$ or $1 \rightarrow 0$ transition within the concatenation of CSV $\underline{S}_{k+1}$ with $S_k(n)$. As an example, if we have, say, the following state sequence:

$$\underbrace{\cdots 010}_{\underline{S}_k}\underbrace{11001110}_{\underline{S}_{k+1}}$$

The Hamming weights in Equation 3.19 are then $W_H(\underline{S}_{k+1}) = 5$ and $W_H(S_k(n)) = 0$. Also $n = 8$ and $P_{ij} = P_{01}$.

Having obtained an expression for the non-zero CSV transition probabilities (Equation 3.19), we now integrate these findings in the transition probability matrix $P^{(n)}$. The resulting 'approximate' transition matrix for the case of slow channel fading is

$$\tilde{P}^{(n)} = \begin{pmatrix} P_{00}^{(M)}[1-(n-1)b] & P_{00}^{(M)}b & 0 & P_{00}^{(M)}b & 0 & \cdots & 0 & 0 & 0 & P_{01}^{(M)} \\ P_{10}^{(M)} & 0 & 0 & 0 & \cdots & 0 & P_{11}^{(M)}g & 0 & P_{11}^{(M)}g & P_{11}^{(M)}[1-(n-1)g] \\ & \vdots & & & & & & & \vdots & \\ P_{10}^{(M)} & 0 & 0 & 0 & \cdots & 0 & P_{11}^{(M)}g & 0 & P_{11}^{(M)}g & P_{11}^{(M)}[1-(n-1)g] \end{pmatrix} \tag{3.20}$$

The tilde ($\tilde{}$) symbol indicates that channel round-trip delay $\delta = M - 1$ is taken into account, i.e. $\tilde{P}^{(n)}$ gives transition probabilities between codewords separated by a gap of δ bits, as shown in Figure 3.2. The derivation of $\tilde{P}^{(n)}$ from Equation 3.19 is shown in Appendix A.

As an example, for codeword length $n = 3$, and for nb and ng small, the approximate transition matrix is

$$\tilde{P}^{(3)} = \begin{pmatrix} P_{00}^{(M)}(1-2b) & P_{00}^{(M)}b & 0 & P_{00}^{(M)}b & 0 & 0 & 0 & P_{01}^{(M)} \\ P_{10}^{(M)} & 0 & 0 & 0 & P_{11}^{(M)}g & 0 & P_{11}^{(M)}g & P_{11}^{(M)}(1-2g) \\ & \vdots & & & & & \vdots & \\ P_{10}^{(M)} & 0 & 0 & 0 & P_{11}^{(M)}g & 0 & P_{11}^{(M)}g & P_{11}^{(M)}(1-2g) \end{pmatrix} \tag{3.21}$$

By inspection, the above matrices are sparse, having many zero elements. Each row of the $\tilde{P}^{(n)}$ matrix has only $n+1$ non-zero elements, out of a total of 2^n elements. Furthermore, out of the $n+1$ non-zero elements, $n-1$ elements have the same value. These two properties - the sparsity of the transition matrix and the equality of some of its elements - can be exploited, as will be shown later, to obtain a closed form expression for $P(A^l = I^l)$. From $P(A^l = I^l)$, computing throughput for the three popular ARQ schemes is a trivial matter, as shown in Equations 3.1 through 3.6.

It is also important to note at this point that $\tilde{P}^{(n)}$ is still a stochastic matrix. Indeed, the sum of all elements in any given row of $\tilde{P}^{(n)}$ in Equation 3.20 is equal to one. For instance, summing the elements of the first row of $\tilde{P}^{(3)}$ in Equation 3.21 gives:

$$P_{00}^{(M)}(1-2b) + P_{00}^{(M)}b + P_{00}^{(M)}b + P_{01}^{(M)} = P_{00}^{(M)} + P_{01}^{(M)}$$

Consequently, the approximations applied in this section for slow fading channels conserve the Markov properties of the CSV transition process. The approximations used here transform the 2^n-state Markov chain into one in which many transitions are not allowed. These are the less probable transitions in the original process. The matrix $\tilde{P}^{(n)}$ of Equation 3.20 can be seen as describing a Markov process that approximates the channel state stochastic process for channel parameters nb and ng small.

3.2.2 Computation of $P(A^l = I^l)$ for small b and g

In the previous section, it was shown that, for slow fading channels, many of the CSV transition probabilities can be approximated by zero. In doing so, only $n+1$ of the 2^n elements in each row of $\tilde{P}^{(n)}$ are non-zero. In this section, we show that this property simplifies the computation of $P(A^l = I^l)$ and thus of ARQ throughput.

It was shown in Section 3.1 that $P(A^l = I^l)$ can be written as

$$P(A^l = I^l) = \beta_1^0 \frac{g}{b+g} + \beta_1^1 \frac{b}{b+g}$$

where β_1^0 and β_1^1 are obtained recursively starting with β_l^0 and β_l^1, using the relation of Equation 3.12 repeated here for convenience:

$$\beta_k(S_{k-1}(n)) = \sum_{\underline{S}_k} \beta_{k+1}(\underline{S}_k) P(A_k = 1|\underline{S}_k) P(\underline{S}_k|S_{k-1}(n))$$

Now, substituting the transition probabilities from matrix $\tilde{P}^{(n)}$ and the probabilities $P(A_k = 1|\underline{S}_k)$ from 3.15 in the above equation gives the following expression for the joint probability $P(A^l = I^l)$

$$P(A^l = I^l) = \left(\frac{g}{b+g} \quad , \quad \frac{b}{b+g}\right) Q^l \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \tag{3.22}$$

where

$$Q = \begin{pmatrix} P_{00}^{(M)}[1-(n-1)b][1-(1-p_0)^n] & P_{00}^{(M)} b\alpha(n) + P_{01}^{(M)}[1-(1-p_1)^n] \\ P_{11}^{(M)} g\alpha(n) + P_{10}^{(M)}[1-(1-p_0)^n] & P_{11}^{(M)}[1-(n-1)g][1-(1-p_1)^n] \end{pmatrix},$$

and

$$\alpha(n) = n - 1 - (1-p_0)^n r \frac{1-r^{n-1}}{1-r}, \qquad r = \frac{1-p_1}{1-p_0}.$$

A detailed derivation of Equation 3.22 is given in Appendix B.

The above result is of great importance, since it gives us $P(A^l = I^l)$, and thus ARQ throughput, as a closed form expression. Furthermore, we no longer need a recursive relation to compute $P(A^l = I^l)$; ARQ throughput for slow fading channels is now a direct function of the channel parameters b, g, δ, p_0, p_1, and the codeword length n. In the next section, some illustrative throughput performance plots of ARQ systems over slow fading channels are obtained using the expression of Equation 3.22.

3.2.3 ARQ Performance Plots for Slow Fading Channels

Some typical throughput performance plots for the three popular ARQ protocols over slow fading channels are given here. In Figure 3.3, throughput is plotted as a function of codeword length n, all other channel parameters kept constant, for the three popular ARQ protocols. These plots are obtained after computing the

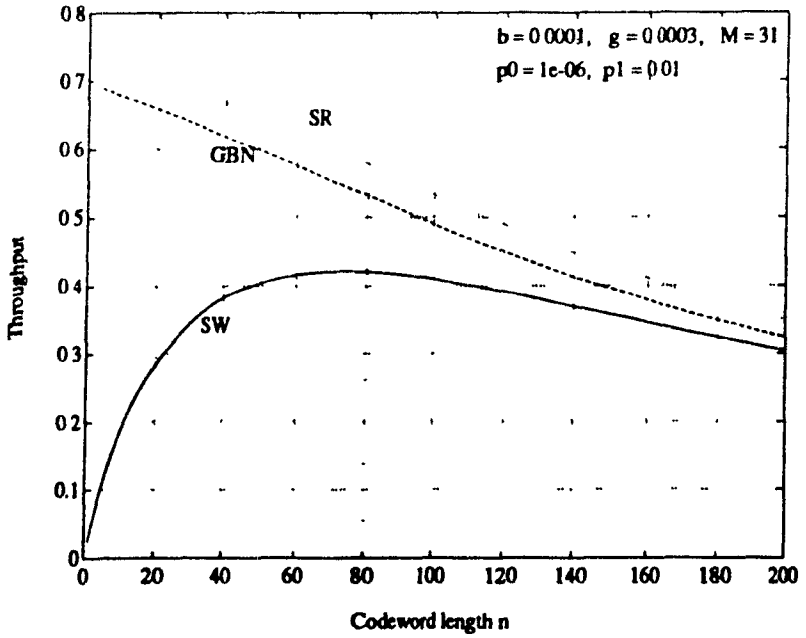

Figure 3.3: **Throughput as a function of codeword length for the SW, GBN, and SR ARQ protocols**

joint probabilities $P(A^l = I^l)$ using the closed form relation of Equation 3.22. The $P(A^l = I^l)$'s are computed for $l = 1, 2, \cdots$ up to the point where they become negligibly small ($P(A^l = I^l) < 10^{-4}$). These joint probabilities are then summed as shown in Equation 3.6 to give the average number of transmissions $E[T]$. From $E[T]$, the throughput η for the three ARQ protocols discussed in this thesis is found using Equations 3.1, 3.2 and 3.3.

The throughput performance plots shown in Figure 3.3 are for a typical slow fading channel, having small 'good-to-bad' and 'bad-to-good' state transition probabilities ($b = 0.0001$, $g = 0.0003$). The average bit error probabilities in the 'good' and 'bad' states are taken to be $p_0 = 1 \times 10^{-6}$ and $p_1 = 0.01$, respectively. The channel round trip delay is assumed to be $\delta = 30$ bits, and the code rate $R = 0.75$.

The plots of throughput efficiency versus codeword length are useful in that they show how codeword length should be selected to maximize ARQ throughput. From Figure 3.3, it can be seen that throughput decreases with increasing n for both the Selective Repeat and the Go-Back-N protocols. This is due to the fact that the larger a codeword is, the larger the probability of having an error in that codeword. However, one must bear in mind that larger codewords are generally more powerful than their shorter counterparts. For a fixed code rate R, longer codewords can provide superior

error detecting / correcting capability.

For the Stop-and-Wait ARQ protocol, Figure 3.3 shows that codeword length has an optimum at $n = n^* = 75$, for the given channel. Throughput efficiency for the SW scheme increases with n up to a maximum, and then starts to slowly decrease with increasing n, in a manner similar to that for the GBN and SR schemes. That η_{SW} increases with increasing n for $n \leq n^*$ is due to the wasted time incurred in the SW scheme between consecutive codeword transmissions. Since a fixed overhead of δ bits is incurred for each codeword transmission, it makes sense to increase the amount of information (i.e. the codeword size), sent over the channel at each transmission. However, for very large n, the probability of codeword error becomes significantly large and offsets the advantage gained by increasing n. These two factors cause the throughput efficiency of the SW ARQ protocol to have a maximum as shown in Figure 3.3. This behaviour has also been shown in previous work on codeword length optimization [36], [31], [32].

Throughput efficiency is often expressed in the literature [5] as a function of the bit error rate prevailing over the channel. In the case of the Gilbert-Elliott fading channel, we cannot talk of a channel bit error rate, but rather of an *average bit error rate* P_{av}, which we define as the weighted average of the crossover probabilities in the 'good' and 'bad' states

$$P_{av} = \frac{g}{b+g} p_0 + \frac{b}{b+g} p_1.$$

A plot of throughput versus average bit error rate for the three conventional ARQ protocols is shown in Figure 3.4 for the same channel parameters as in Figure 3.3, that is $b = 0.0001$, $g = 0.0003$. The codeword length is fixed at $n = 30$, and N (the number of codewords between successive retransmissions of the same codeword) is taken to be equal to 2, giving channel idle time $\delta = 30$. A code rate $R = 0.75$ is also assumed here.

From Figure 3.4, one can see that throughput efficiency is fairly constant for low average bit error rates. For larger P_{av} ($P_{av} > 10^{-3}$), throughput drops rapidly with increasing average bit error rate, for the three protocols considered here. This behaviour is in line with typical throughput vs. bit error rate plots given in the literature [5]. This exact relationship between throughput efficiency and average bit error rate depends of course on such parameters as codeword length, channel round-trip delay, fading channel parameters. More on throughput performance and its dependence on the above parameters can be found in Section 3.4.

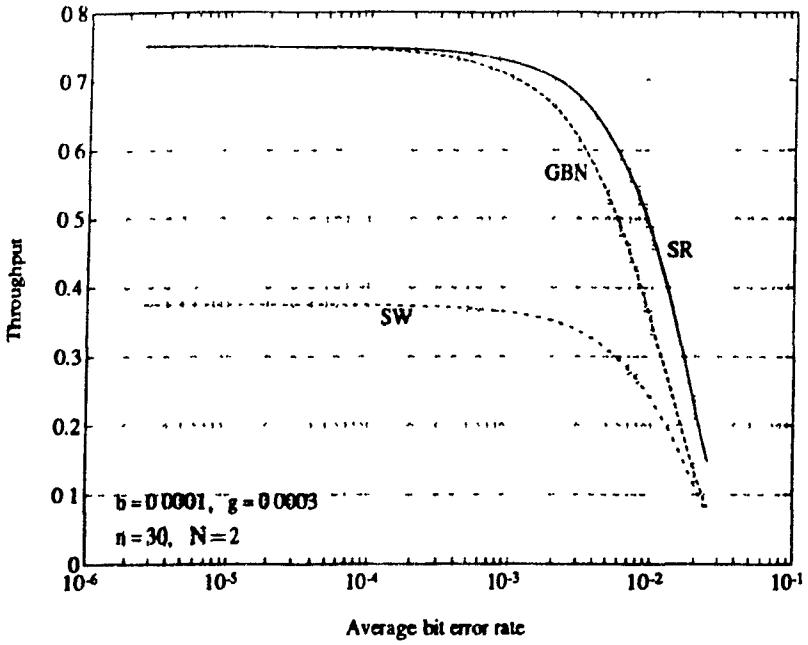

Figure 3.4: **Throughput versus average bit error rate for the SW, GBN, and SR ARQ protocols**

3.3 Throughput for Fast Fading Channels

In this section, it is shown that upper and lower bounds on $P(A^l = I^l)$ can be obtained, thus circumventing the problem of performing computations of order 2^n, as exposed in Section 3.1. The bounds developed in this section become increasingly tight for increasing ng and nb. Thus, for fast fading channels, these bounds give a good estimate of ARQ throughput performance, and also confirm the approximation developed in Section 3.2 for slow fading channels.

3.3.1 Upper and Lower Bounds on Throughput

Bounds on ARQ throughput efficiency for fading channels were first derived by Beirouti *et al* [23]. This derivation is reiterated in this section for convenience. In order to develop upper and lower bounds on throughput, we first define $N_0^{(n)}(i)$ and $N_1^{(n)}(i)$ to be the number of visits to the 'good' and 'bad' state, respectively, in n steps at the i^{th} consecutive codeword transmission. Note that

$$N_0^{(n)}(i) + N_1^{(n)}(i) = n.$$

The variables $N_0^{(n)}(i)$ and $N_1^{(n)}(i)$ are also referred to as the state 0 and state 1 occupation times random variables. Using these definitions, the $P(A_k = 1|\underline{S}_k)$ terms in (3.8) can be approximated by

$$P(A_k = 1|\underline{S}_k) \simeq 1 - (1-p_1)^{N_1^{(n)}(k)}(1-p_0)^{N_0^{(n)}(k)}, \tag{3.23}$$

as was shown in Equations 3.13, 3.14 and 3.15. Again, we assume a good error detecting code is used. This result comes from the probability of receiving a correct codeword conditioned on the channel state vector $\underline{S}_k$. Now, substituting the above expression in variable $\beta_k(\underline{S}_{k-1})$ of Equation 3.8 and making some approximations, it can be shown that the joint probability $P(A^l = I^l)$ is upper and lower bounded as follows:

$$[np_0\frac{g}{b+g} + np_1\frac{b}{b+g} - b\rho(p_1-p_0)]^l \le P(A^l = I^l) \le [np_0\frac{g}{b+g} + np_1\frac{b}{b+g} + g\rho(p_1-p_0)]^l,$$

where

$$\rho = \frac{(1-b-g)^{(\delta+1)}}{(b+g)^2}[1-(1-b-g)^n].$$

These bounds are derived in Appendix C.

It must be noted here that the above bounds on $P(A^l = I^l)$ are of the form $P(A^l = I^l) \le r^l$, or $P(A^l = I^l) \ge r^l$, ie. $P(A^l = I^l)$ is upper bounded by a constant raised to the power l. $0 \le r \le 1$, since $P(A^l = I^l)$ is a probability measure. Now, from Equation 3.6, it can be seen that the average number of transmissions $E[T]$ becomes a geometric series expansion since:

$$\begin{aligned} E[T] &= 1 + \sum_{l=1}^{\infty} P(A^l = I^l) \\ &= 1 + P(A^1 = I^1) + P(A^2 = I^2) + \cdots \\ &= 1 + r + r^2 + r^3 + \cdots + r^l + \cdots \\ &= \frac{1}{1-r} \\ &= \frac{1}{1-P(A^1 = I^1)}. \end{aligned} \tag{3.24}$$

Now, from equations C.5 and C.6, the bounds on $P(A^l = I^l)$ translate into bounds on $E[T]$ as follows:

$$\frac{1}{1-[np_0\frac{g}{b+g} + np_1\frac{b}{b+g} + g\rho(p_1-p_0)]} \le E[T] \le \frac{1}{1-[np_0\frac{g}{b+g} + np_1\frac{b}{b+g} - b\rho(p_1-p_0)]}. \tag{3.25}$$

The above bounds on $E[T]$ can then be substituted in (3.1), (3.2) and (3.3), and thus bounds on throughput can be obtained for the three ARQ schemes discussed in Section 1.3.

3.3.2 Utility of the Bounds

Upper and lower bounds on average number of transmissions $E[T]$, and thus on ARQ throughput were developed in the previous section. The following questions arise, however: How tight are these bounds? And, do these bounds agree with the throughput approximation obtained in Section 3.2 for slow fading channels?

Looking at Equation 3.25, the upper and lower bounds on $E[T]$ approach each other as the terms $b\rho(p_1 - p_0)$ and $g\rho(p_1 - p_0)$ decrease in value. The variable ρ is a function of the channel parameters b, g, δ, and the codeword length n, and is repeated here for convenience:

$$\rho = \frac{(1-b-g)^{(\delta+1)}}{(b+g)^2}[1-(1-b-g)^n]. \tag{3.26}$$

From Equation 3.26 above, ρ decreases with increasing b, g and δ. Thus, the bounds on throughput become increasingly tight with increasing $b+g$ and increasing channel round trip delay δ. These conditions translate into lower channel memory, since the larger δ, the lower the correlation between consecutive codewords; and the larger b and g, the faster the fading, and thus the lower the memory of the channel. Also, in order to have tight bounds, $p_1 - p_0$ must be small, and since, typically, $p_1 \gg p_0$, this requirement translates into p_1 being small.

Now, for $b\rho(p_1 - p_0)$ and $g\rho(p_1 - p_0)$ small, the average number of transmissions tends to

$$E[T] \longrightarrow \frac{1}{1-[np_0\frac{g}{b+g} + np_1\frac{b}{b+g}]}. \tag{3.27}$$

The term in square brackets can be seen as a weighted average of the crossover probabilities p_0 and p_1, since $g/(b+g)$ and $b/(b+g)$ are the two-state Gilbert-Elliott model stationary probabilities. Increasing channel parameters b, g and δ can be seen as decreasing the memory of the channel. Indeed, the expression in (3.27) is equivalent to that for a memoryless channel, since from [5], the average number of codeword transmissions is

$$E[T] = \frac{1}{1-P_d},$$

where P_d is the probability of detecting an error at the ARQ receiver. Also, from Equation 3.13,

$$P_d \simeq 1-(1-p)^n \simeq np,$$

for $np \ll 1$, where p is the crossover probability of the memoryless BSC. Thus, as b

and g increase, the bounds converge to the throughput efficiency for a memoryless channel

Plots of throughput as a function of codeword length n are shown in Figure 3.5 for the three ARQ protocols discussed in this thesis. Both these bounds and the curves resulting from the application of the approximations of Section 3.2 are plotted in Figure 3.5. The upper and lower bounds on throughput are shown as dashed lines whereas the throughput as computed using the approximations of Section 3.2 is shown as a solid line. These plots assume crossover probabilities $p_0 = 1 \times 10^{-6}$, $p_1 = 0.01$, channel round trip delay $\delta = 49$, and code rate $R = 0.75$. As can be seen from the figure, the bounds on throughput are quite tight for channel parameters $b = 0.01$ and $g = 0.03$. These bounds become looser for smaller b and g as shown in Figure 3.6 which shows throughput performance plots for the case $b = 0.008$, $g = 0.024$. From the two figures, the convergence of the bounds for increasing b and g can hence easily be seen.

The bounds obtained in this section have no practical significance, however, since they converge towards the trivial case of a memoryless channel; one might take the plot of throughput versus n for the memoryless channel to estimate ARQ performance over fast fading channels. However, the bounds remain useful in order to validate the approximations on throughput efficiency obtained in Section 3.2. Indeed, from Figure 3.5, one can see that the throughput curves obtained using the approximation techniques are well within the upper and lower bounds, for a significant range of values of n. However, for $n > 175$, the plots using the approximation technique are no longer between the upper and lower bounds. This is due to the fact that nb and ng are no longer small for $n > 175$.

From the plots in Figure 3.5, it can be seen that the approximation technique for slow fading channels can safely be used, provided nb and ng are small. In the plots displayed here, the product ng at which the 'approximate' throughput starts diverging from within the bounds is $ng = 5.25$. Thus, keeping $nb, ng \ll 1$ should be sufficient to ensure the accuracy of the approximation technique. One can hence examine channels with $b, g \leq 10^{-4}$ using the approximation technique (for a reasonably large range of codeword lengths n) and compare this to the trivial memoryless channel case. The behaviour of channels with 'intermediate' and low memory ($1 < b, g \leq 10^{-4}$) can be extrapolated from the slow fading and memoryless channel case. In so doing, one can infer the behaviour of ARQ schemes for a wide range of channel memory. Hence, in the

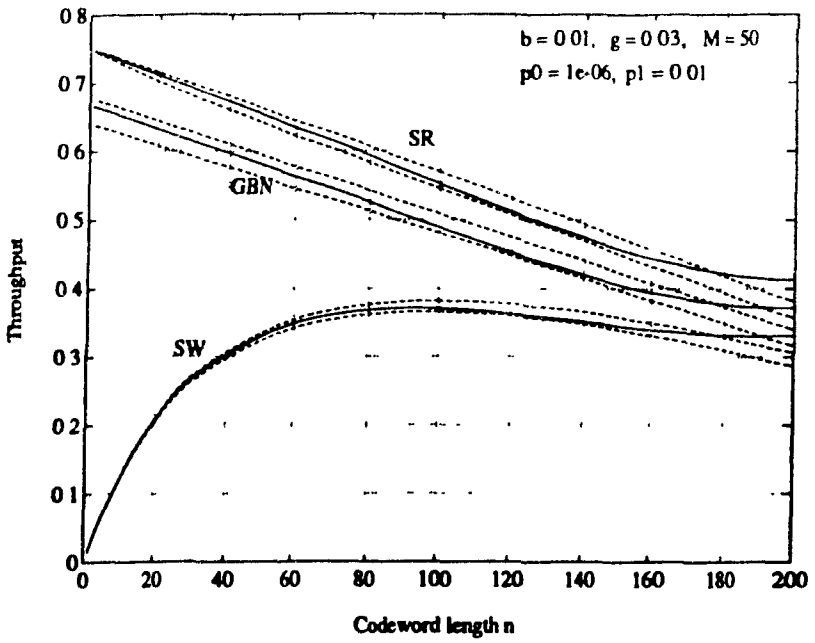

Figure 3.5: **Throughput as a function of codeword length using both the approximation technique (solid line) and the bounding technique (dashed line), for $b = 0.01$, $g = 0.03$**

forthcoming sections, we analyse ARQ schemes over channels with $b, g \leq 10^{-4}$ using the approximation technique, and compare this with ARQ schemes over memoryless channels. Throughput for channels with no memory is trivial to compute, since again,

$$E[T] = \frac{1}{1 - P_d},$$

where P_d is the probability of detecting an error in a given codeword:

$$P_d \simeq 1 - (1 - p)^n.$$

The probability p is the bit error rate of the channel.

3.4 Fading Effects on ARQ Throughput

We have seen in Chapters 1 and 2 that channel memory or fading results in errors occurring in bursts; the larger the channel memory, the burstier the resulting channel error sequences. In this section, we examine the effects of channel memory on the throughput efficiency of ARQ systems.

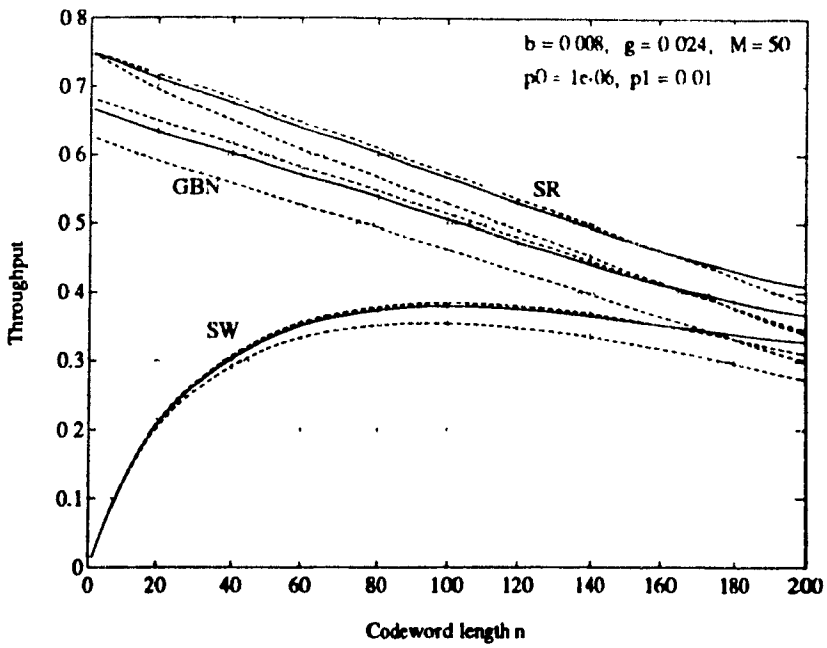

Figure 3.6: **Throughput vs. average bit error rate for(solid line) and the bounding technique (dashed line)**

How channel memory impacts ARQ throughput performance is not directly obvious from the results obtained thus far in this thesis. The relationship between throughput efficiency and the Gilbert-Elliott model parameters is clouded by the shear complexity of the expressions for ARQ throughput obtained in Sections 3.2 and 3.3. Hence, one must resort to graphically representing throughput efficiency for various degrees of channel memory. The graphs in Figure 3.7 show throughput performance for both the Selective Repeat and Go-Back-N ARQ schemes. The dotted lines give the throughput for the memoryless channel case. The dashed lines give throughput for a channel with large memory, namely $b = 1 \times 10^{-6}$, $g = 3 \times 10^{-6}$, whereas the solid lines show throughput efficiency for an 'intermediate' memory case, namely $b = 1 \times 10^{-4}$ and $g = 3 \times 10^{-4}$. A codeword length $n = 30$ and a gap $N = 7$ between codewords is assumed. From the plots in Figure 3.7, it can be seen that the larger the memory of the channel, the lower the throughput efficiency of both ARQ schemes. This relationship is especially apparent for high average bit error rates ($P_{av} > 10^{-2}$). For instance, for $P_{av} = 0.02$, the throughput efficiency (of the SR scheme) for the highly fading channel ($b = 1 \times 10^{-6}$, $g = 3 \times 10^{-6}$) is exactly half that for the memoryless channel; $\eta_{SR} = 0.55$ for the memoryless channel, and only $\eta_{SR} = 0.275$, for the high memory channel.

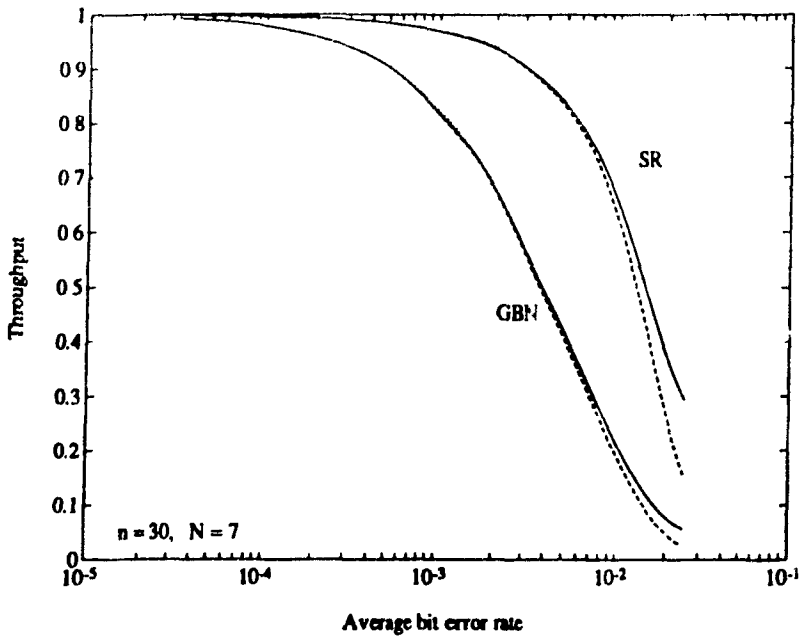

Figure 3.7: Throughput vs. average bit error rate for a memoryless channel (dotted line), for the case $b = 1 \times 10^{-4}$, $g = 3 \times 10^{-4}$ (solid line), and the case $b = 1 \times 10^{-6}$, $g = 3 \times 10^{-6}$ (dashed line)

From the above observations, it is now clear that improving ARQ throughput efficiency on a fading channel can be achieved by decreasing the memory of that channel. One way of decreasing channel memory is to delay retransmission of codewords detected in error, i.e. increasing the gap N between retransmissions of the same codeword. Note that performance of the Go-Back-N scheme deteriorates with increasing N, since η_{GBN} is inversely proportional to N, as shown in Equation 3.2. Thus, increasing N would be advantageous only in the SR protocol. To show the advantage gained from delaying retransmission of consecutive codewords, throughput performance of the SR ARQ scheme for several values of N is shown in Figure 3.8.

From the figure, one can see that a significant improvement in throughput efficiency can be obtained by increasing N from 1 to 20, for the given fading channel ($b = 1 \times 10^{-4}$, $g = 3 \times 10^{-4}$). Increasing N has the effect of decorrelating the states of consecutive codewords, thus decreasing channel memory. Furthermore, for $N = 60$, the throughput efficiency of the SR ARQ system over the given fading channel exceeds that of an equivalent system over a memoryless channel. Although surprising at first, this fact can be explained as follows: The memory intrinsic to the channel causes successive channel state bits to be highly correlated. Hence, channel state

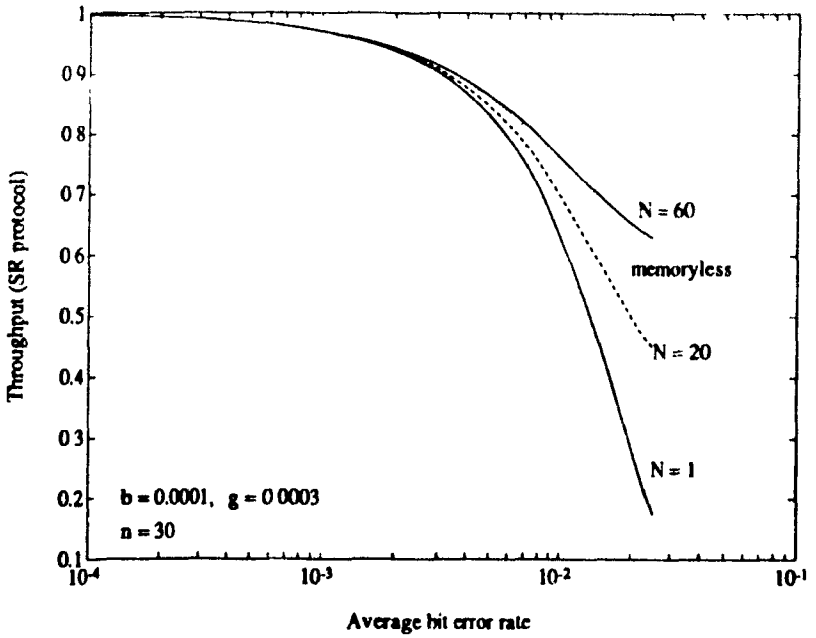

Figure 3.8: **Throughput efficiency of the SR ARQ protocol for several values of N**

vectors (CSV's) tend to be either entirely in the 'good' state, or entirely in the 'bad' state. Now, since in this particular channel example (Figure 3.8), the steady-state probability of the channel being in the 'good' state is fairly high ($g/(b+g) = 0.75$), the probability that an entire codeword is in the 'good' state is also very high. In this way, the throughput performance of the SR ARQ scheme over the given fading channel can be made better than that over a memoryless channel. Stated in a different way, the combined effect of high memory within a CSV and no memory between consecutive CSV's can bring about an improvement in throughput efficiency.

The relationship between throughput efficiency and the gap δ between consecutive codewords is shown more explicitly in Figure 3.9. Again, increasing the gap δ decorrelates consecutive CSV's, and thus improves throughput efficiency of the SR ARQ channel coding scheme. Also, the improvement in ARQ throughput is greater for larger average bit error rates. Indeed, as can be seen in Figure 3.9, throughput efficiency can be increased three folds, for $p_1 = 0.1$, by introducing a gap of 1000 bits between consecutive codewords; for $p_1 = 0.05$, the gain in throughput is smaller (only 30 % better).

Although increasing N does result in improved throughput performance, such a

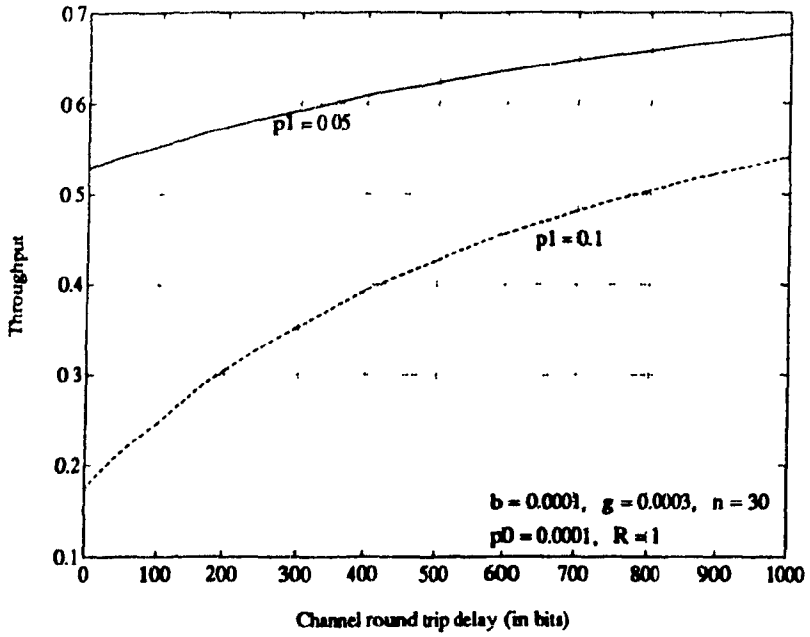

Figure 3.9: **Throughput vs. channel round trip delay δ, for two values of p_1**

scheme is not desirable in practice. Indeed, increasing the delay between retransmissions of codewords translates into large delays in delivering data to the end user, as well as increased buffering requirement and thus hardware memory; it is shown in [5] that for adequate system performance, the buffer size at the ARQ receiver must increase with N. For this reason, practical ARQ systems and other error-control schemes have traditionally used code interleaving to suppress channel memory, as discussed in Section 2.5. However, code interleaving requires extensive buffering, both at the transmitter and receiver, and hence causes an additional delay in delivering data to the end user. In this research, we use frequency hopping, i.e. switching transmission channel at a constant pace, to suppress channel memory, without increasing the data delivery delay. Frequency-hopped ARQ systems are the object of the next chapter.

Chapter 4

Frequency-Hopped ARQ

In the previous section, it was shown that increasing the time delay N between codeword retransmissions improves the throughput efficiency of Selective Repeat ARQ systems. This improvement is realized mainly by decreasing the memory of the communication channel. In this chapter, the use of *frequency-hopped* codeword retransmission is introduced as an alternate means of decreasing channel memory, and thus, improving ARQ throughput. These improvements in throughput performance are illustrated and discussed in Section 4.3

4.1 Frequency Hopping

In a few words, frequency-hopped data transmission consists of sending consecutive data packets alternatively onto one of m different carrier frequencies. A frequency-hopped data communications system is shown in Figure 4.1. Going through the system block diagram, it can be seen that data packets or codewords leaving the channel encoder enter a frequency-hopped modulator which modulates the incoming codewords onto a number of different carrier frequencies $f_1, f_2, f_3, \cdots, f_m$, according to a pre-determined 'key' sequence. The resulting signal is then sent over a physical channel and demodulated at the receiving end using the same 'key' frequency sequence.

In a frequency-hopped system, the frequencies on which data is transmitted are 'independent', i.e. they represent statistically independent channels. The frequencies $f_1, f_2, f_3, \cdots, f_m$ are chosen in such a way that the channel conditions prevailing at

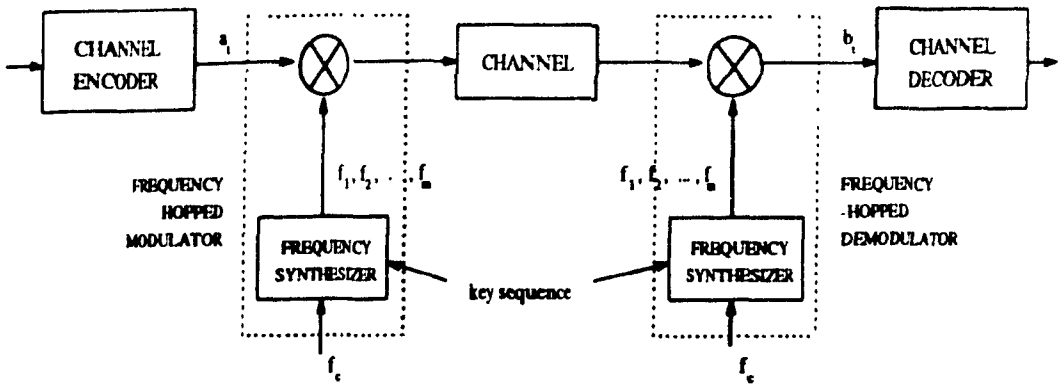

Figure 4.1: **Frequency-hopped modulation / demodulation**

frequency f_1 are independent of those prevailing at frequencies f_2, f_3, $\cdots$, f_m, and similarly for the other frequencies. Thus, in sending consecutive codewords onto these independent frequencies, channel memory between codewords is reduced. How these independent frequencies are obtained physically is not important here. We assume m independent frequencies, i.e. m independent channels can be obtained on a given communications link, and base our subsequent analysis on this assumption.

Frequency hopping systems were initially used in the late 1960's, for military communications applications [43]. By sending data onto various carrier frequencies, one is able to escape or reduce interference from hostile jamming sources. Later, frequency hopping was considered as a potential modulation scheme for a new standard in digital cellular telephony to be implemented in the early 1990's [44]. One of the advantages of using frequency hopping in such wireless network applications is again to escape interference on a given channel from adjacent channel users. Avoiding such interference is achieved by switching (or hopping) carrier frequency at regular intervals. Due to its use of a number of different carrier frequencies for a single user, frequency-hopped modulation is considered a spread-spectrum communications scheme. More on spread-spectrum techniques can be found in [4] and [45].

In this research, we use frequency hopping as a means of escaping deep fades that may occur in a given channel. If a codeword is found in error at the ARQ receiver, it is statistically very likely that the particular channel in use at that time is in a 'bad' state. Thus, retransmitting the data onto another independent frequency (and thus onto another independent channel) may be more successful than using the same channel which is probably undergoing a deep fade. One can hence see heuristically that frequency-hopped codeword modulation can bring about a possible improvement

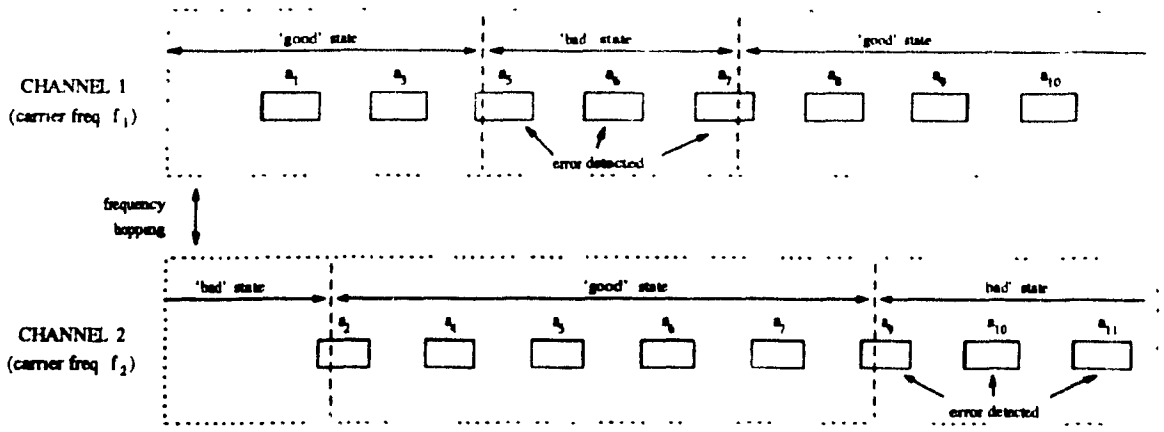

Figure 4.2. Illustration of frequency hopping for ARQ data transmission

in ARQ throughput efficiency.

An illustration of the use of frequency hopping to escape bad channel conditions is shown in Figure 4.2 for the case of two independent channels. From the figure, it can be seen that data codewords are sent over the two independent channels, in an alternating fashion. When a codeword is detected in error, which is likely due to 'bad' channel conditions, it is retransmitted onto the other channel, which may then be in a better state. Thus, one can intuitively see a possible improvement in ARQ throughput. For example, looking at Figure 4.2, the codewords a_1 through a_4 are sent alternatively on Channels 1 and 2, which are both in the 'good' state, at that time. Later on, the conditions on Channel 1 deteriorate, causing an error to occur in codeword a_5. As a consequence, a_5 is retransmitted, this time on Channel 2, which is still in a 'good' state. Hence, codeword a_5 is successfully transmitted after a total of two trials, instead of possibly more trials, if no frequency hopping is used. Figure 4.2 shows the case of hopping over two frequencies only. This scheme can of course be generalized to the case of m independent frequencies. The frequency hopping scheme described here is 'automatic', i.e. the carrier frequency is changed at each codeword transmission.

The frequency hopping protocol discussed in the previous paragraph and illustrated in Figure 4.2 assumes no channel round trip delay. Now, since consecutive codeword transmissions are done on different frequencies, it may very well be that, due to channel round trip delay, retransmission of an erroneous codeword occurs on the same frequency as the original codeword in error. Such scenario, which may occur for both GBN and SR protocols used over a channel with non-negligible round trip

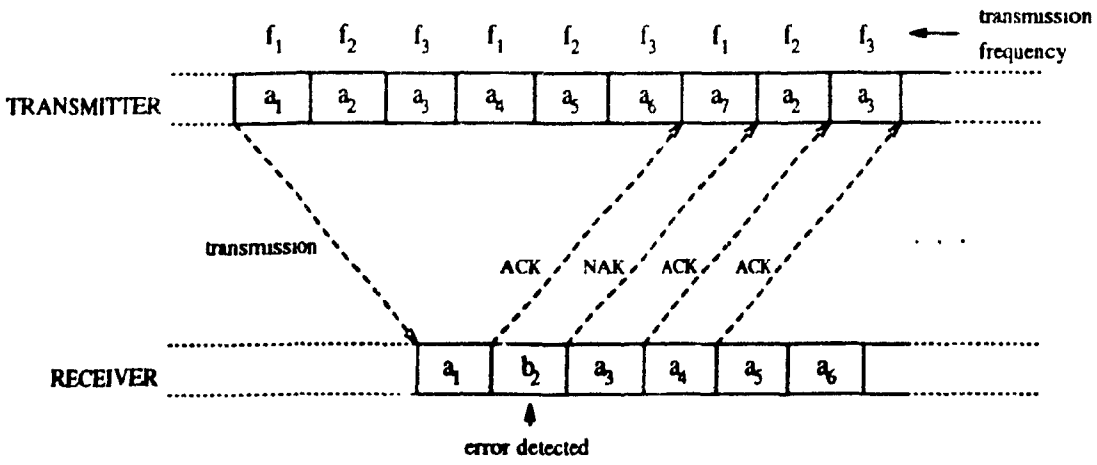

Figure 4.3: **Codeword retransmission on the same frequency**

delay, is illustrated in Figure 4.3. The ARQ system in Figure 4.3 assumes the Go-Back-N protocol is used with hopping on three independent frequencies. The channel round trip delay is assumed to result in $N = 6$, here.

As shown in the figure, codeword a_2 is received in error, and, consequently, a retransmission is requested. However, due to channel round trip delay, codeword a_2 is retransmitted at frequency f_2, which is the same frequency as the one used in the initial transmission of a_2. Such occurrence is a worse case scenario, however, since it defeats the purpose of frequency hopping. It corresponds to the case where the parameter N is a multiple of the number of independent frequencies used for hopping. Also note that this worse case scenario applies only for the GBN and SR protocols. In the Stop-and-Wait protocol, one can make sure that erroneous codewords are retransmitted onto a different frequency, since the transmitter must wait for an ACK / NAK signal before proceeding.

Now, in the worse case scenario where erroneous codewords are always retransmitted on the same frequency, the performance of the frequency-hopped ARQ system is then comparable to that of a regular ARQ system without frequency hopping. In order to avoid such event, other possibly more complex frequency hopping protocols must be used for GBN and SR ARQ. For instance, hopping may be achieved according to a pseudo random frequency sequence with very long period. The pseudo random sequence would be known to both transmitter and receiver. Or alternatively, one could hold in memory the last frequency on which a codeword was transmitted. In this fashion, one could ensure that an erroneous codeword is not retransmitted on the

same frequency. An even better automatic frequency hopping protocol would keep in memory the m previous frequencies on which an erroneous codeword has been sent so as to ensure that the codeword is retransmitted alternatively on all m available frequencies. The resulting hopping algorithm may be complex and impractical to implement, however. So far, we have discussed only 'automatic' frequency hopping schemes. One can also consider hopping 'on demand', in which the transmission frequency changes only when a NAK signal is received. With little thought, the reader can see that such a protocol is likely to yield better throughput performance than automatic frequency hopping. Frequency hopping may also be subject to various standards on bandwidth allocation. The design of an 'optimal' or 'quasi optimal' automatic hopping scheme is in itself an interesting area for further research. In this thesis, we assume an optimum automatic frequency hopping protocol can be found. In the next section, we derive expressions for the throughput of such optimum automatic frequency-hopped ARQ system. Of course, non-optimal hopping protocols will result in lower throughput efficiency.

4.2 Throughput of Frequency-Hopped ARQ Systems

This section presents a derivation of expressions for the throughput performance of optimal frequency-hopped ARQ systems. The general case of m independent frequencies (i.e. m independent channels) is considered here.

It was shown in Section 3.1 that the average number of codeword transmissions is given by:

$$\begin{aligned} E[T] &= 1 + P(A^1 = I^1) + P(A^2 = I^2) + P(A^3 = I^3) + \cdots + P(A^l = I^l) + \cdots \\ &= 1 + P(A_1 = 1) + P(A_1 = 1, A_2 = 1) + \cdots + P(A_1 = 1, A_2 = 1, \cdots, A_l = 1) + \cdots \end{aligned}$$

Recall that A_i is the event of detecting an error in the i^{th} transmission of the same codeword. If an m-frequency hopping scheme is used with frequencies f_1, f_2, f_3, $\cdots$, f_m, the events A_1, A_2, A_3, $\cdots$, A_m are independent of each other, whereas the events A_1, A_{m+1}, A_{2m+1}, $\cdots$ are dependent on each other. In general, the events A_j, A_{m+j}, A_{2m+j}, $\cdots$ $(j = 1,2,3,\cdots,m)$ are dependent on each other. Thus, the probability $P(A^l = I^l)$ can be rewritten as

$$P(A^l = I^l) = P(A_1 = 1, A_{m+1} = 1, A_{2m+1} = 1, \cdots, A_{(r_1-1)m+1} = 1)$$

$$\times P(A_2 = 1, A_{m+2} = 1, A_{2m+2} = 1, \cdots, A_{(r_2-1)m+2} = 1) \times \cdots$$
$$\cdots \times P(A_j = 1, A_{m+j} = 1, A_{2m+j} = 1, \cdots, A_{(r_j-1)m+j} = 1) \times \cdots$$
$$\cdots \times P(A_m = 1, A_{2m} = 1, A_{3m} = 1, \cdots, A_{r_m m} = 1).$$

Or, alternatively, to shorten our notation, we define

$$u_{lj} \equiv P(A_j = 1, A_{m+j} = 1, A_{2m+j} = 1, \quad \cdot, A_{(r_j-1)m+j} = 1)$$

and hence, the joint probability $P(A^l = I^l)$ becomes

$$P(A^l = I^l) = u_{l1} \times u_{l2} \times u_{l3} \times \cdots \times u_{lj} \times \cdots \times u_{lm}.$$

The constant r_j is the number of events A_i to be considered in the joint probability u_{lj} and can be easily found to be:

$$r_j = \begin{cases} l \text{ div } m & \text{if } (l \text{ rem } m) < j \\ (l \text{ div } m) + 1 & \text{if } (l \text{ rem } m) \geq j \end{cases}$$

The operation 'l div m' gives the integer part of the ratio l/m, whereas the 'l rem m' operation gives the remainder from the division l/m.

We now need to find each of the u_l's in terms of th Gilbert-Elliott model parameters. Once that is done, the joint probabilities $P(A^l = I^l)$ can be obtained, and hence the throughput performance of the three ARQ protocols can be derived.

Now, it was shown in Section 3.1 that $P(A^l = I^l)$ can be written as:

$$\begin{aligned} P(A^l = I^l) &= \sum_{\underline{S}^l} P(A^l = I^l | \underline{S}^l) P(\underline{S}^l) \\ &= \sum_{\underline{S}_1} P(A_1 = 1 | \underline{S}_1) P(\underline{S}_1) \sum_{\underline{S}_2} P(A_2 = 1 | \underline{S}_2) P(\underline{S}_2 | \underline{S}_1) \cdots \\ &\quad \cdots \sum_{\underline{S}_l} P(A_l = 1 | \underline{S}_l) P(\underline{S}_l | \underline{S}_{l-1}) \end{aligned}$$

This development can also be applied to each of the probabilities $P(A_j = 1, A_{m+j} = 1, A_{2m+j} = 1, \cdots, A_{(r_j-1)m+j} = 1)$ as follows:

$$\begin{aligned} u_{lj} &= \sum_{\underline{S}_j} P(A_j = 1 | \underline{S}_j) P(\underline{S}_j) \sum_{\underline{S}_{m+j}} P(A_{m+j} = 1 | \underline{S}_{m+j}) P(\underline{S}_{m+j} | \underline{S}_j) \\ &\quad \sum_{\underline{S}_{2m+j}} P(A_{2m+j} = 1 | \underline{S}_{2m+j}) P(\underline{S}_{2m+j} | \underline{S}_{m+j}) \cdots \\ &\quad \cdots \sum_{\underline{S}_{(r_j-1)m+j}} P(A_{(r_j-1)m+j} = 1 | \underline{S}_{(r_j-1)m+j}) P(\underline{S}_{(r_j-1)m+j} | \underline{S}_{(r_j-2)m+j}) \end{aligned}$$

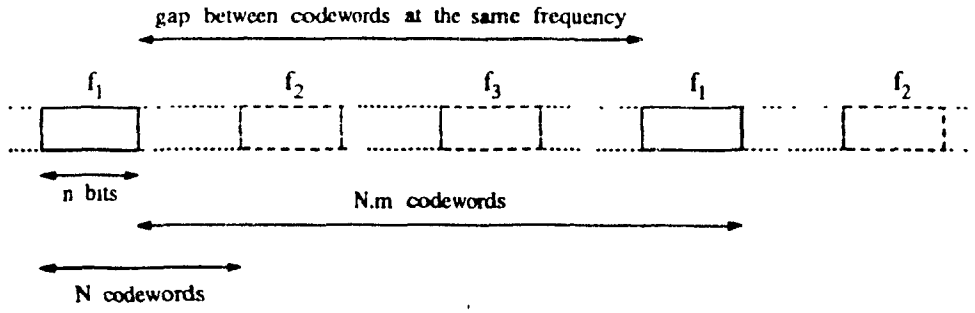

Figure 4.4: **Gap between consecutive codewords at the same frequency ($m = 3$, $N = 2$)**

Thus, the results of Sections 3.1 and 3.2 also apply here, except that we should use transition probabilities $P(\underline{S}_{(k+1)m+j}|\underline{S}_{km+j})$ instead of $P(\underline{S}_{k+1}|\underline{S}_k)$. Thus, for a slow fading channel, we have:

$$\begin{aligned} P(A_j = 1, A_{m+j} = 1, A_{2m+j} = 1, \cdots, A_{r_j m+j} = 1) &= \left(\frac{g}{b+g} \quad , \quad \frac{b}{b+g}\right) Q^{r_j} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad (4.1) \\ &= \underline{\pi}^T Q^{r_j} \underline{1} \end{aligned}$$

where the matrix Q is given by

$$Q = \begin{pmatrix} P_{00}^{(M)}[1-(n-1)b][1-(1-p_0)^n] & P_{00}^{(M)} b\alpha(n) + P_{01}^{(M)}[1-(1-p_1)^n] \\ P_{11}^{(M)} g\alpha(n) + P_{10}^{(M)}[1-(1-p_0)^n] & P_{11}^{(M)}[1-(n-1)g][1-(1-p_1)^n] \end{pmatrix},$$

$\underline{\pi}$ is the steady-state probability vector of the two-state Gilbert-Elliott Markov model,

$$\underline{\pi}^T = \left(\frac{g}{b+g} \quad , \quad \frac{b}{b+g}\right)$$

and $\underline{1}$ is a unity vector with two components. Here, the gap M between consecutive codewords is no longer equal to the round trip delay $R\tau$ plus one, but rather

$$\begin{aligned} M &= [(N-1)m + (m-1)]n + 1 \\ &= (Nm-1)n + 1, \end{aligned}$$

as can be seen from Figure 4.4.

It is interesting at this point to analyse the case $m \to \infty$, that is the case where infinitely many independent frequencies are available for hopping. In the limit as $m \to \infty$, one will always be able to retransmit data at a frequency that has not been used before (we design the frequency hopping protocol that way). Hence, all transmission channels can be made independent, and the joint probability $P(A^l = I^l)$

can always be written as

$$\begin{aligned} P(A^l = I^l) &= [P(A_1 = 1)]^l \\ &\equiv q^l \end{aligned}$$

Consequently, the average number of transmissions of the same codeword is a geometric series expansion

$$\begin{aligned} E[T] &= 1 + q + q^2 + q^3 + \cdots + q^l + \cdots \\ &= \frac{1}{1-q} \end{aligned}$$

since $q = P(A_1 = 1) < 1$. Here $q = \underline{\pi}^T Q \underline{1}$, and the gap M between retransmissions of a codeword at the same frequency tends to infinity as $m \rightarrow \infty$. Thus, the probabilities $P_{00}^{(M)}$, $P_{01}^{(M)}$, $P_{10}^{(M)}$, and $P_{11}^{(M)}$ are the steady-state probabilities of the two-state Gilbert-Elliott Markov chain:

$$\begin{aligned} P_{00}^{(M)} &= P_{10}^{(M)} = g/(b+g) \\ P_{01}^{(M)} &= P_{11}^{(M)} = b/(b+g) \end{aligned}$$

Although the case $m \rightarrow \infty$ is not obtainable in practice, it is still a useful case as it gives us the maximum throughput gain that can be achieved using frequency-hopped modulation on a given fading channel. In the next section, we look at the improvement in throughput efficiency that can be obtained using a frequency-hopped ARQ system with two and three independent frequencies, as compared to the case of infinitely many independent frequencies ($m \rightarrow \infty$) and the trivial case of no frequency hopping.

4.3 Performance of Frequency-Hopped ARQ Systems

The manner in which throughput efficiency of ARQ schemes can be improved by the use of frequency-hopped modulation was briefly discussed in Section 4.1. In this section, we present performance plots of frequency-hopped ARQ schemes in light of the expressions for throughput performance derived in Section 4.2. Throughout this presentation, we consider the cases of no frequency hopping, hopping on two frequencies, hopping on three frequencies, and hopping on an infinite number of independent frequencies.

Performance plots showing ARQ throughput efficiency for both the SR and GBN protocols are given in Figures 4.5 and 4.6, for fading channels with $b = 1 \cdot 10^{-4}, g = 3 \times 10^{-4}$ and $b = 1 \times 10^{-6}, g = 3 \times 10^{-6}$, respectively. In both figures, one can see a substantial improvement in throughput efficiency when using the $m = 2$ (dashed line) and $m = 3$ ('dash-dotted' line) frequency hopping schemes, over the no frequency hopping scheme (solid line). The case of frequency hopping over an infinite number of independent frequencies is also shown (dotted line) in the figures. For $m \to \infty$, one achieves the maximum possible improvement in throughput for this particular channel. The plots show throughput efficiency versus average bit error rate. The range in average bit error rate is obtained by keeping p_0 constant ($p_0 = 1 \times 10^{-5}$), and varying p_1. By using frequency hopping, an additional gap between codewords at the same frequency is introduced. For $m \to \infty$, one obtains the maximum possible improvement in throughput for this particular channel.

The improvement in throughput efficiency shown in Figures 4.5 and 4.6 comes partly from the introduction of an additional gap between consecutive codewords transmitted at the same frequency. By transmitting a data stream alternatively on, say, three independent channels, the gap between codewords at the same frequency becomes $(3N - 1)n$ bits, instead of $(N - 1)n$ bits, if no frequency hopping were used (see Figure 4.4). However, this increased gap between codewords does not translate in a delay in the transmission of data, as would be the case if N were increased (see Section 3.4). To the transmitter and receiver, the delay between consecutive codewords is the same with or without frequency-hopped modulation. However, by dividing up transmitted codewords onto several independent channels, the gap between codewords on a particular channel is increased, and thus, the memory between CSV's on that particular channel is decreased. Such decrease in channel memory results in an improvement in ARQ performance, as discussed in Section 3.4.

For the case of hopping on an infinite number of frequencies, Figures 4.5 and 4.6 show that the throughput curves approach an asymptotic value as the average bit error rate is increased towards unity. For the SR protocol, $\eta_{SR} \to 0.75$ as $p_1 \to 1$ for the given channel parameters. This behaviour can be explained as follows: As mentioned in Section 3.4, the memory between channel state bits is quite large for highly bursty channels. Hence, a given codeword tends to be either entirely in the 'good' state, or entirely in the 'bad' state. Now, since for the channels in Figure 4.5 and 4.6, the probability of being in the 'good' state is $P_g = g/(b + g) = 0.75$, and the BER in the 'good' state is very low ($p_0 = 1 \times 10^{-5}$), the probability of successfully transmitting

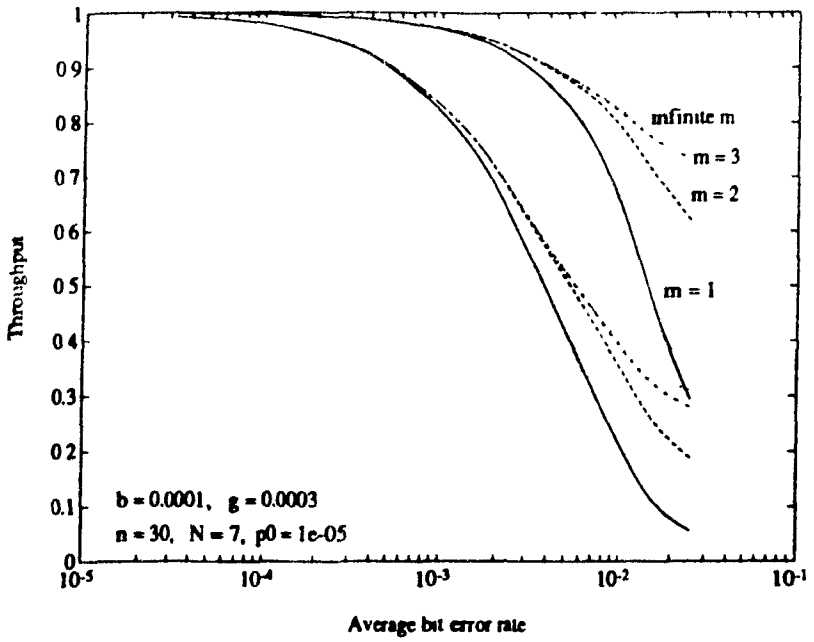

Figure 4.5: **Throughput efficiency for several frequency-hopped ARQ schemes over a fading channel with $b = 1 \times 10^{-4}, g = 3 \times 10^{-4}$**

a codeword tends to 0.75 as well. Also, as $m \rightarrow \infty$, consecutive codewords become totally uncorrelated, and thus ARQ throughput for the SR scheme also tends to 0.75, no matter how large p_1 is made, as shown in both Figure 4.5 and 4.6.

From the above argument, the extent to which ARQ throughput can be improved by the use of frequency hopping depends highly on the steady-state probability of being in a 'good' channel state. Indeed, Figure 4.7 and 4.8 show the improvement in throughput efficiency obtainable for $P_g = 0.5$ and $P_g = 0.25$, respectively. As can be seen from these figures, the smaller the steady-state probability of being in the 'good' state, the smaller the improvement that can be achieved with frequency hopping. However, one must bear in mind that the channel parameters b, g, p_0, and p_1 are related to each other, as shown in Section 2.2; one cannot vary b and g without effect on p_0 and p_1. Thus, one must use parameters from real channels in order to fairly analyse the improvement in throughput brought about by the use of frequency hopping. In this research, we analyse the performance improvement that can be brought about by the use of frequency hopping for a given Gilbert-Elliott channel, regardless of how it applies to physical channels.

We assumed in this section that m independent channels are readily obtained.

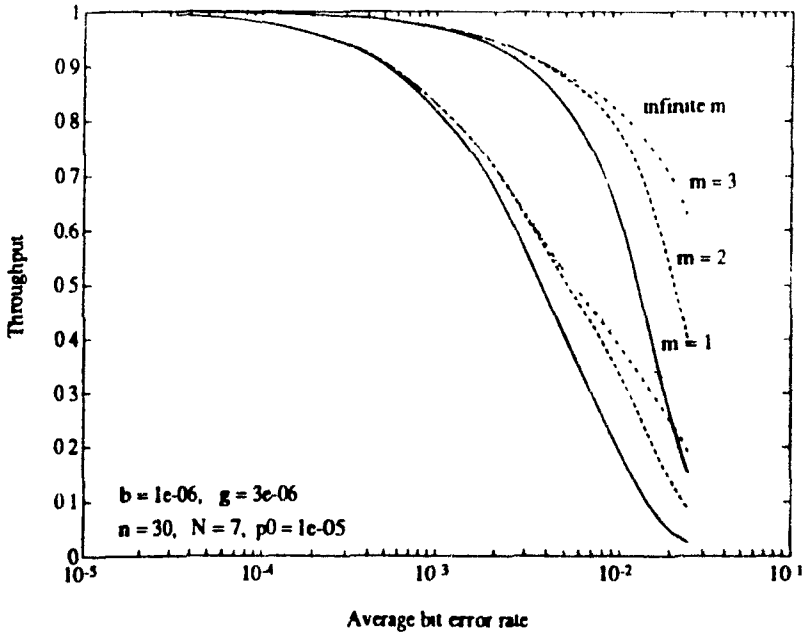

Figure 4.6: Throughput efficiency for several frequency-hopped ARQ schemes over a fading channel with $b = 1 \times 10^{-6}, g = 3 \times 10^{-6}$

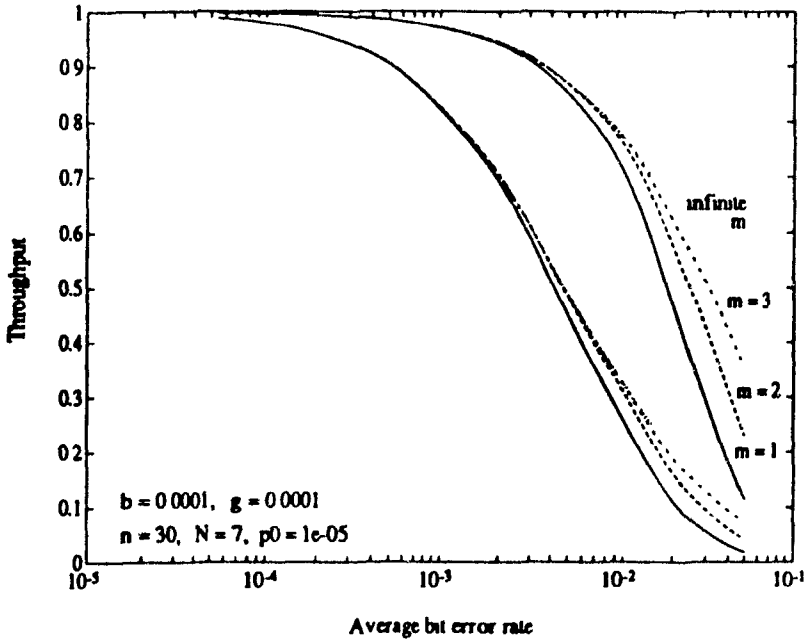

Figure 4.7: Throughput efficiency for several frequency-hopped ARQ schemes over a fading channel with $b = 1 \times 10^{-6}, g = 1 \times 10^{-6}$

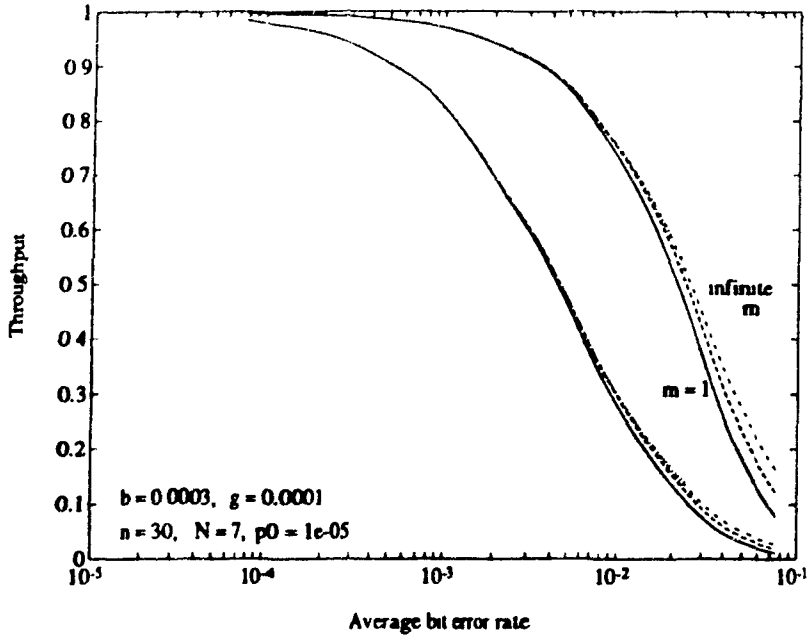

Figure 4.8: **Throughput efficiency for several frequency-hopped ARQ schemes over a fading channel with $b = 3 \times 10^{-4}$, $g = 1 \times 10^{-4}$**

However, in real communication scenarios, totally independent channels can be difficult or even impossible to obtain. For instance, communication links affected by the so called shadowing phenomenon (i.e. when the transmitter and receiver are separated by a large obstacle, say a hill or a large building) all channels are equally degraded. In such event, the real performance of the frequency-hopped ARQ system should be close to the $m = 1$ case shown in the figures. In other words, communication links hampered by shadowing are not expected to benefit much from frequency hopping. On the other hand, communications suffering from multi-path interference are expected to benefit substantially from frequency hopping. Indeed, multi-path fading is highly frequency dependent, as discussed in Chapter 1, and thus independent channels can be easily obtained.

In light of the above discussion, and taking into account the fact that a non-optimal frequency hopping protocol may have to be used, the results displayed in this section should be interpreted with some caution. These results represent the throughput efficiency of ideal automatic frequency-hopped ARQ schemes. In other words, the plots of frequency-hopped ARQ system performance shown in Figures 4.5 - 4.8 should be viewed as upper bounds on throughput efficiency. The extent to which

real system performance approaches the upper bounds in the figures depends on the availability of m truly independent channels and on the optimality of the automatic hopping protocol used

Chapter 5

Conclusions

This research examines the throughput performance of ARQ systems over fading channels modelled by the two-state Gilbert-Elliott Markov chain. Chapter 2 shows the necessity of extending the Gilbert-Elliott to represent channel states over an entire n-bit codeword In Chapter 3, we show that the average number of transmissions T of the same codeword can be written as a series expansion of joint probabilities $P(A^l = I^l)$. From these joint probabilities, the throughput efficiency for the three conventional ARQ schemes is derived for Gilbert-Elliott fading channels. Although the expressions thus obtained are only approximations of the throughput performance of ARQ systems, these approximations are accurate for slow fading or high memory channels. Due to the complexity of the throughput expressions, only graphical results are useful in giving an insight into the effects of varying channel parameters and codeword length on throughput performance. Curves of throughput efficiency versus codeword length n are given. These curves confirm that there exists an optimal codeword length for data transmission using the Stop-and-Wait protocol, as is the case for memoryless channels. This optimum value of n depends highly on the average bit error rate and on the fading channel parameters.

Performance curves showing throughput versus average bit error rate are also given in Chapter 3. These plots show that throughput efficiency decreases with slower channel fading, or alternatively, with higher channel memory. From this, one can imply that superior ARQ performance can be achieved by reducing channel memory. One way of doing this is to delay the retransmission of erroneous codewords. A better way of reducing channel memory between consecutive data packets is to use frequency-hopped codeword modulation, i.e. transmitting consecutive data

packets alternatively on m independent channels. Expressions for the throughput performance of frequency-hopped ARQ systems are obtained for an assumed optimal frequency hopping protocol. These expressions are derived for a general number m of independent frequencies. The resulting throughput performance is plotted as a function of average bit error rate for $m = 1, 2, 3$ and the ideal case $m \to \infty$. From the plots, one can deduce that a significant improvement in throughput efficiency can be achieved by using frequency hopping for slow fading channels with high average bit error rates. The improvement in throughput performance is a function of the number of independent channels available; the larger the number of channels, the better the performance. The improvement brought about by frequency hopping is also highly dependent on the steady state probabilities in the Gilbert-Elliott model. The higher the probability of being in the 'good' state, the larger the improvement in throughput efficiency brought about by frequency hopping. One must keep in mind that the results obtained for frequency-hopped ARQ schemes are only upper bounds on their performance. The extent to which the performance of real frequency-hopped ARQ systems approaches the upper bounds is contingent on the availability of m truly independent channels and on the optimality of the frequency hopping protocol used.

This research dealt with the estimation of throughput performance of various ARQ schemes over the Gilbert-Elliott channel. Further areas for research would be to explore ARQ system performance on other Markov channel models such as McCullough's bit regenerative model or Fritchman's partitioned Markov chain model, for instance. One may also be interested in analysing the performance of ARQ systems using real error detecting codes. In this research a perfect error detecting code was assumed; real codes would give more realistic performance estimates for a given channel. Another area for further study is the performance evaluation of hybrid feedback communication schemes on fading channels. Only conventional ARQ schemes were examined in this research, whereas hybrid schemes are frequently encountered in applications. For instance, Type - II hybrid schemes are often well suited for high bit error rate non-stationary channels.

Appendix A

Transition Matrix for Slow Fading Channels

This appendix shows the derivation of the transition matrix $\tilde{P}^{(n)}$ for slow fading channels, i.e. for small b and g parameters in the Gilbert-Elliott model. The matrix $\tilde{P}^{(n)}$ provides the transition probabilities between n-bit CSV's, taking into account a gap $\delta = M - 1$ due to channel round-trip delay between consecutive codewords.

First of all, we recognize that $P(\underline{S}_{k+1}|\underline{S}_k) = P(\underline{S}_{k+1}|S_k(n))$ from the Markov property. $P(\underline{S}_{k+1}|S_k(n) = 0)$ are the elements occupying the even rows of the transition matrix (rows $0, 2, 4, \cdots, 2^n - 2$), and $P(\underline{S}_{k+1}|S_k(n) = 1)$ occupy the odd rows of this same transition matrix $P^{(n)}$. Here, for the sake of clarity, we analyze the elements of the even and odd rows separately.

- **Even rows:**

The non-zero CSV transitions can be depicted as follows:

$$\begin{array}{ccc} \times\times\times\cdots\times\times 0 & \longrightarrow & 000\underbrace{11\cdots1111}_{\text{m ones}} \\ S_k(n) = 0 & & \underline{S}_{k+1} = 2^m - 1, \quad m = 1, 2, \cdots, n \end{array}$$

Now, in order to describe a given CSV, we give it a value in decimal notation which is equivalent to the binary sequence it represents. For instance, the binary sequence $\underline{S}_k = (01001)$ is denoted here by its decimal equivalent, $\underline{S}_k = 9$. Now, from the above

bit pattern, the non-zero CSV transition probability is

$$P(\underline{S}_{k+1} = 2^m - 1 | S_k(n) = 0) = \begin{cases} (1-g)^{m-1}(1-b)^{n-m}b, & \underline{S}_{k+1} \neq 0 \\ (1-b)^n, & \underline{S}_{k+1} = 0 \end{cases} \quad (A.1)$$

- **Odd rows:**

Again, we depict the non-zero probability CSV transitions, this time for odd rows of the transition matrix $P^{(n)}$, as follows:

$$\times\times\times\cdots\times\times 1 \longrightarrow 1\,1\,1\underbrace{00\cdots0000}_{\text{m zeroes}}$$

$$S_k(n) = 1 \qquad \underline{S}_{k+1} = 2^n - 2^m, \quad m = 1, 2, \cdots, n$$

For the above CSV transitions, the transition probability is easily seen to be

$$P(\underline{S}_{k+1} = 2^n - 2^m | S_k(n) = 1) = \begin{cases} (1-b)^{m-1}(1-g)^{n-m}g, & \underline{S}_{k+1} \neq 2^n - 1 \\ (1-g)^n, & \underline{S}_{k+1} = 2^n - 1 \end{cases} \quad (A.2)$$

So far, the above transition probabilities $P(\underline{S}_{k+1}|\underline{S}_k)$ assumed no channel idle time ($\delta = 0$ or $M = 1$). In order to account for channel round-trip delay, slight modifications have to be made to the probabilities expressed in Equations A.1 and A.2. The resulting transition matrix is denoted by $\tilde{P}^{(n)}$, with the tilde (˜) symbol emphasizing the fact that channel idle time is taken into account.

The modifications needed to the transition probabilities $P(\underline{S}_{k+1}|\underline{S}_k)$ to account for channel idle time can be easily deduced by considering the CSV transition diagram of Figure 3.2. In this figure, the $\delta = M - 1$ bits of idle time between two consecutive CSV's are shown. Instead of the transition probability P_{ij} between $S_k(n)$ and $S_{k+1}(1)$, one must use an M^{th} order transition probability $P_{ij}^{(M)}$ to take into account the fact that the transition between state i and state j is done in M steps.

Now, the M^{th} order 2×2 transition matrix P^M can be shown [41] to be

$$P^M = \begin{pmatrix} P_{00}^{(M)} & P_{01}^{(M)} \\ P_{10}^{(M)} & P_{11}^{(M)} \end{pmatrix} = \begin{pmatrix} \frac{g}{b+g} + \frac{b(1-b-g)^M}{b+g} & \frac{b}{b+g} - \frac{b(1-b-g)^M}{b+g} \\ \frac{g}{b+g} - \frac{g(1-b-g)^M}{b+g} & \frac{b}{b+g} + \frac{g(1-b-g)^M}{b+g} \end{pmatrix}. \quad (A.3)$$

Substituting the elements of P^M in Equations A.1 and A.2 gives the following probabilities:

$$P(\underline{S}_{k+1}|S_k(n) = 0) = \begin{cases} P_{00}^{(M)}(1-b)^{n-1}, & \underline{S}_{k+1} = 0 \\ P_{00}^{(M)}(1-b)^{n-m-1}(1-g)^{m-1}b, & \underline{S}_{k+1} = 2^m - 1, \quad m = 1, 2, \quad , n-1 \\ P_{01}^{(M)}(1-g)^{n-1}, & \underline{S}_{k+1} = 2^n - 1 \\ 0, & \text{otherwise} \end{cases}$$

(A.4)

$$P(\underline{S}_{k+1}|S_k(n)=1)=\begin{cases} P_{10}^{(M)}(1-b)^{n-1}, & \underline{S}_{k+1}=0 \\ P_{11}^{(M)}(1-g)^{n-m-1}(1-b)^{m-1}g, & \underline{S}_{k+1}=2^n-2^m, \quad m=1,2,\quad ,n-1 \\ P_{11}^{(M)}(1-g)^{n-1}, & \underline{S}_{k+1}=2^n-1 \\ 0, & \text{otherwise} \end{cases} \tag{A.5}$$

The above probabilities can be further approximated by using a binomial expansion and retaining only the first order terms of this expression. A binomial of the form $(1-x)^n$ can be written as a series expansion as follows:

$$(1-x)^n = 1 - nx + \frac{n(n-1)}{2}x^2 - \frac{n(n-1)(n-2)}{3!}x^3 + \cdots.$$

For nx small ($nx \ll 1$), the above series expansion can be safely truncated to

$$(1-x)^n \simeq 1 - nx.$$

Thus, for small nb and ng, the above first order approximation can be used in Equations A.4 and A.5, giving the following transition probabilities:

$$P(\underline{S}_{k+1}|S_k(n)=0)=\begin{cases} P_{00}^{(M)}[1-(n-1)b], & \underline{S}_{k+1}=0 \\ P_{00}^{(M)}b, & \underline{S}_{k+1}=2^m-1, \quad m=1,2,\cdots,n-1 \\ P_{01}^{(M)}, & \underline{S}_{k+1}=2^n-1 \\ 0, & \text{otherwise} \end{cases} \tag{A.6}$$

$$P(\underline{S}_{k+1}|S_k(n)=1)=\begin{cases} P_{10}^{(M)}, & \underline{S}_{k+1}=0 \\ P_{11}^{(M)}g, & \underline{S}_{k+1}=2^n-2^m, \quad m=1,2,\cdots,n-1 \\ P_{11}^{(M)}[1-(n-1)g], & \underline{S}_{k+1}=2^n-1 \\ 0, & \text{otherwise} \end{cases} \tag{A.7}$$

In Equation A.6, $P_{00}^{(M)}(1-b)^{n-1}$ is approximated by $P_{00}^{(M)}[1-(n-1)b]$, $P_{00}^{(M)}(1-b)^{n-m-1}(1-g)^{m-1}b$ by $P_{00}^{(M)}b$, and $P_{01}^{(M)}(1-g)^{n-1}$ by $P_{01}^{(M)}$. These particular approximations are done in such a way as to preserve the stochastic nature of the ensuing transition matrix, as will be shown later.

From the above transition probabilities, the resulting transition matrix $\tilde{P}^{(n)}$ is of the form:

$$\tilde{P}^{(n)}=\begin{pmatrix} P_{00}^{(M)}[1-(n-1)b] & P_{00}^{(M)}b & 0 & P_{00}^{(M)}b & 0 & \cdots & 0 & 0 & 0 & P_{01}^{(M)} \\ P_{10}^{(M)} & 0 & 0 & 0 & \cdots & 0 & P_{11}^{(M)}g & 0 & P_{11}^{(M)}g & P_{11}^{(M)}[1-(n-1)g] \\ & \vdots & & & & & & & \vdots & \\ P_{10}^{(M)} & 0 & 0 & 0 & \cdots & 0 & P_{11}^{(M)}g & 0 & P_{11}^{(M)}g & P_{11}^{(M)}[1-(n-1)g] \end{pmatrix} \tag{A.8}$$

Appendix B

Deriving $P(A^l = I^l)$ for Slow Fading Channels

A derivation of the joint probabilities $P(A^l = I^l)$ for the slow fading channel case is given here.

It is shown in Section 3.1 that $P(A^l = I^l)$ can be written as:

$$P(A^l = I^l) = \beta_1^0 \frac{g}{b+g} + \beta_1^1 \frac{b}{b+g},$$

with the variables β_k recursively computed according to the following

$$\beta_k(S_{k-1}(n)) = \sum_{\underline{S}_k} \beta_{k+1}(\underline{S}_k) P(A_k = 1|\underline{S}_k) P(\underline{S}_k|S_{k-1}(n)).$$

Now, the variable β_k is a function of channel state bit $S_{k-1}(n)$ only, and can thus take one of two possible values:

$$\begin{aligned} \beta_k^0 &= \sum_{\underline{S}_k} \beta_{k+1}(\underline{S}_k) P(A_k = 1|\underline{S}_k) P(\underline{S}_k|S_{k-1}(n) = 0) \\ \beta_k^1 &= \sum_{\underline{S}_k} \beta_{k+1}(\underline{S}_k) P(A_k = 1|\underline{S}_k) P(\underline{S}_k|S_{k-1}(n) = 1). \end{aligned}$$

$P(\underline{S}_k|S_{k-1}(n) = 0)$ and $P(\underline{S}_k|S_{k-1}(n) = 1)$ are elements of the transition matrix $\tilde{P}^{(n)}$ given in Equation 3.20. Replacing these elements in β_k^0 and β_k^1 above gives:

$$\begin{aligned} \beta_k^0 = \; & \beta_{k+1}^0 P_{00}^{(M)} [1 - (n-1)b] P(A_k = 1|\underline{S}_k = 0) \\ & + \beta_{k+1}^1 P_{00}^{(M)} b \sum_{\underline{S}_k = 2^m - 1} P(A_k = 1|\underline{S}_k) \\ & + \beta_{k+1}^1 P_{01}^{(M)} P(A_k = 1|\underline{S}_k = 2^n - 1), \quad m = 1, 2, \cdots, n-1 \end{aligned} \tag{B.1}$$

and

$$\begin{aligned}\beta_k^0 &= \beta_{k+1}^0 P_{10}^{(M)} P(A_k = 1|\underline{S}_k = 0) \\ &\quad + \beta_{k+1}^0 P_{11}^{(M)} g \sum_{\underline{S}_k = 2^n - 2^m} P(A_k = 1|\underline{S}_k) \\ &\quad + \beta_{k+1}^1 P_{11}^{(M)} [1 - (n-1)g] P(A_k = 1|\underline{S}_k = 2^n - 1), \quad m = 1, 2, \cdots, n-1\end{aligned} \tag{B.2}$$

The expressions above show that $P(A^l = I^l)$ can be written in terms of the channel parameters b, g, δ, and the conditional probabilities $P(A_k = 1|\underline{S}_k)$. The latter are obtained from Equation 3.15, repeated here with $N_0^{(n)}(k)$ and $N_1^{(n)}(k)$ rewritten as Hamming weights $W_H(\underline{S}_k)$:

$$\begin{aligned}P(A_k = 1|\underline{S}_k) &\simeq 1 - (1-p_1)^{W_H(\underline{S}_k)}(1-p_0)^{n - W_H(\underline{S}_k)} \\ &= 1 - (1-p_0)^n \left(\frac{1-p_1}{1-p_0}\right)^{W_H(\underline{S}_k)}.\end{aligned}$$

Hence, we can write the $P(A_k = 1|\underline{S}_k)$ terms in Equation B.1 and B.2 as follows:

$$P(A_k = 1|\underline{S}_k = 0) \simeq 1 - (1-p_0)^n$$

$$P(A_k = 1|\underline{S}_k = 2^n - 1) \simeq 1 - (1-p_1)^n$$

$$\begin{aligned}\sum_{\underline{S}_k = 2^m - 1} P(A_k = 1|\underline{S}_k) &\simeq n - 1 - (1-p_0)^n \sum_{\underline{S}_k = 2^m - 1} \left(\frac{1-p_1}{1-p_0}\right)^{W_H(\underline{S}_k)}, \quad m = 1, 2, \cdots, n-1 \\ &= n - 1 - (1-p_0)^n [r + r^2 + r^3 + \cdots + r^{n-1}],\end{aligned}$$

where $r = (1-p_1)/(1-p_0)$. The above sum can be seen as a geometric series expansion:

$$1 + r + r^2 + r^3 + \cdots + r^{k-1} = \frac{1 - r^k}{1 - r}.$$

Thus,

$$\sum_{\underline{S}_k = 2^m - 1} P(A_k = 1|\underline{S}_k) \simeq n - 1 - (1-p_0)^n r \frac{1 - r^{n-1}}{1 - r}, \quad m = 1, 2, \cdots, n-1$$

Similarly,

$$\begin{aligned}\sum_{\underline{S}_k = 2^n - 2^m} P(A_k = 1|\underline{S}_k) &\simeq n - 1 - (1-p_0)^n \sum_{\underline{S}_k = 2^n - 2^m} r^{W_H(\underline{S}_k)}, \quad m = 1, 2, \cdots, n-1 \\ &= n - 1 - (1-p_0)^n [r + r^2 + r^3 + \cdots + r^{n-1}] \\ &= n - 1 - (1-p_0)^n r \frac{1 - r^{n-1}}{1 - r} \\ &= \sum_{\underline{S}_k = 2^m - 1} P(A_k = 1|\underline{S}_k) \\ &\equiv \alpha(n).\end{aligned}$$

We are now in a position to summarize the procedure for obtaining $P(A^l = I^l)$ for a slow fading channel.

We compute the variables β_k^0, β_k^1, recursively for $k = l, l-1, l-2, \cdots, 2, 1$ as follows:

$$\begin{aligned}\beta_k^0 &\simeq \beta_{k+1}^0 P_{00}^{(M)}[1-(n-1)b][1-(1-p_0)^n] \\ &\quad +\beta_{k+1}^1 P_{00}^{(M)} b\alpha(n) \\ &\quad +\beta_{k+1}^1 P_{01}^{(M)}[1-(1-p_1)^n]\end{aligned}$$

$$\begin{aligned}\beta_k^1 &\simeq \beta_{k+1}^0 P_{10}^{(M)}[1-(1-p_0)^n] \\ &\quad +\beta_{k+1}^0 P_{11}^{(M)} g\alpha(n) \\ &\quad +\beta_{k+1}^1 P_{11}^{(M)}[1-(n-1)g][1-(1-p_1)^n]\end{aligned}$$

where

$$\alpha(n) = n - 1 - (1-p_0)^n r \frac{1-r^{n-1}}{1-r}, \qquad r = \frac{1-p_1}{1-p_0},$$

and

$$\beta_{l+1}^0 = \beta_{l+1}^1 = 1.$$

Then, $P(A^l = I^l)$ is simply

$$P(A^l = I^l) = \beta_1^0 \frac{g}{b+g} + \beta_1^1 \frac{b}{b+g}.$$

The above procedure can be further simplified by writing the expressions for β_k^0 and β_k^1 in matrix form. Letting,

$$\underline{\beta}_k = \begin{pmatrix} \beta_k^0 \\ \beta_k^1 \end{pmatrix} \qquad , \qquad \underline{\beta}_{k+1} = \begin{pmatrix} \beta_{k+1}^0 \\ \beta_{k+1}^1 \end{pmatrix},$$

and

$$Q = \begin{pmatrix} P_{00}^{(M)}[1-(n-1)b][1-(1-p_0)^n] & P_{00}^{(M)} b\alpha(n) + P_{01}^{(M)}[1-(1-p_1)^n] \\ P_{11}^{(M)} g\alpha(n) + P_{10}^{(M)}[1-(1-p_0)^n] & P_{11}^{(M)}[1-(n-1)g][1-(1-p_1)^n] \end{pmatrix},$$

we can rewrite the above recursion as

$$\underline{\beta}_k = Q\underline{\beta}_{k+1}.$$

Using the same notation, we have:

$$\begin{aligned}
\underline{\beta}_l &= Q\begin{pmatrix}1\\1\end{pmatrix} \\
\underline{\beta}_{l-1} &= Q\underline{\beta}_l = QQ\begin{pmatrix}1\\1\end{pmatrix} = Q^2\begin{pmatrix}1\\1\end{pmatrix} \\
\underline{\beta}_{l-2} &= Q\underline{\beta}_{l-1} = Q^3\begin{pmatrix}1\\1\end{pmatrix} \\
&\vdots \\
\underline{\beta}_1 &= Q^l\begin{pmatrix}1\\1\end{pmatrix} \qquad \text{(B.3)}
\end{aligned}$$

Hence,

$$\begin{aligned}
P(A^l = I^l) &= \beta_1^0\frac{g}{b+g} + \beta_1^1\frac{b}{b+g} \\
&= \left(\frac{g}{b+g} \quad , \quad \frac{b}{b+g}\right)\underline{\beta}_1.
\end{aligned}$$

Replacing the vector $\underline{\beta}_1$ given by Equation B.3 in the above gives

$$P(A^l = I^l) = \left(\frac{g}{b+g} \quad , \quad \frac{b}{b+g}\right)Q^l\begin{pmatrix}1\\1\end{pmatrix}.$$

Appendix C

Deriving $P(A^l = I^l)$ for Fast Fading Channels

It was shown in Section 3.1 that the conditional probability $P(A_k = 1|\underline{S}_k)$ can be written as

$$P(A_k = 1|\underline{S}_k) \simeq 1 - (1 - p_1)^{N_1^{(n)}(k)}(1 - p_0)^{N_0^{(n)}(k)} \tag{C.1}$$

Now, the two factors in Equation C.1 can be approximated with only the first order terms of the binomial series expansion, as shown in Appendix A. Thus, for $N_1^{(n)}(k)p_1, N_0^{(n)}(k)p_0 \ll 1$, (C.1) can be rewritten as:

$$\begin{aligned} P(A_k = 1|\underline{S}_k) &\simeq 1 - [1 - p_1 N_1^{(n)}(k)][1 - p_0 N_0^{(n)}(k)] \\ &\simeq p_1 N_1^{(n)}(k) + p_0 N_0^{(n)}(k), \end{aligned} \tag{C.2}$$

provided np_1 is small.

Substituting (C.2) in variable $\beta_l(\underline{S}_{l-1})$ of Equation 3.8, we have

$$\begin{aligned} \beta_l(\underline{S}_{l-1}) &\simeq \sum_{\underline{S}_l} [p_1 N_1^{(n)}(l) + p_0 N_0^{(n)}(l)] P(\underline{S}_l|\underline{S}_{l-1}) \\ &= E_{\underline{S}_l|\underline{S}_{l-1}} \left\{ p_1 N_1^{(n)}(l) + p_0 N_0^{(n)}(l) \right\}. \end{aligned}$$

This is an averaging operation over $\underline{S}_l$ conditioned on $\underline{S}_{l-1}$ and gives

$$\beta_l(\underline{S}_{l-1}) \simeq p_1 \overline{N}_{1|\underline{S}_{l-1}}^{(n)} + p_0 \overline{N}_{0|\underline{S}_{l-1}}^{(n)} \tag{C.3}$$

$$\text{where,} \quad \overline{N}_{j|\underline{S}_{l-1}}^{(n)} = E_{\underline{S}_l|\underline{S}_{l-1}} \left\{ N_j^{(n)}(l) \right\}, \quad j = 0, 1$$

Now, the mean occupation times for the two-state Markov chain are derived in [11], and the results are summarized here.

$$\begin{aligned}
\mu_{00}^{(m)} &= \frac{mg}{b+g} + \frac{b(1-b-g)[1-(1-b-g)^m]}{(b+g)^2} \\
\mu_{01}^{(m)} &= \frac{mb}{b+g} - \frac{b(1-b-g)[1-(1-b-g)^m]}{(b+g)^2} \\
\mu_{10}^{(m)} &= \frac{mg}{b+g} - \frac{g(1-b-g)[1-(1-b-g)^m]}{(b+g)^2} \\
\mu_{11}^{(m)} &= \frac{mb}{b+g} + \frac{g(1-b-g)[1-(1-b-g)^m]}{(b+g)^2}.
\end{aligned}$$

The variable $\mu_{ij}^{(m)}$ is the mean number of times the Markov chain visits state j, in m steps, given that it was initially in state i. The mean occupation times $\overline{N}_{j|\underline{S}_{l-1}}^{(n)}$ can then be written in terms of the above mean occupation times $\mu_{ij}^{(m)}$ as follows:

$$\overline{N}_{j|\underline{S}_{l-1}}^{(n)} = \overline{N}_{j|S_{l-1}(n)}^{(n)} = \mu_{S_{l-1}(n)j}^{(n+\delta)} - \mu_{S_{l-1}(n)j}^{(\delta)},$$

where $\delta = R\tau$ is the channel round-trip delay time which must be included to account for the idle time between consecutive codeword transmissions.

Replacing the appropriate $\mu_{ij}^{(m)}$ in the above equation, one finds that

$$\begin{aligned}
\overline{N}_{0|S_{l-1}(n)=0}^{(n)} &= \mu_{00}^{(n+\delta)} - \mu_{00}^{(\delta)} \\
&= n\frac{g}{b+g} + b\rho,
\end{aligned}$$

where

$$\rho = \frac{(1-b-g)^{(\delta+1)}}{(b+g)^2}[1-(1-b-g)^n].$$

Similarly,

$$\begin{aligned}
\overline{N}_{0|S_{l-1}(n)=1}^{(n)} &= n\frac{g}{b+g} - g\rho \\
\overline{N}_{1|S_{l-1}(n)=0}^{(n)} &= n\frac{b}{b+g} - b\rho \\
\overline{N}_{1|S_{l-1}(n)=1}^{(n)} &= n\frac{b}{b+g} + g\rho
\end{aligned}$$

Replacing the above expressions in Equation C.3 gives

$$\beta_l = \begin{cases} np_0\frac{g}{b+g} + np_1\frac{b}{b+g} - b\rho(p_1-p_0), & S_{l-1}(n)=0 \\ np_0\frac{g}{b+g} + np_1\frac{b}{b+g} + g\rho(p_1-p_0), & S_{l-1}(n)=1 \end{cases}. \qquad \text{(C.4)}$$

Note that the values of β_l for $S_{l-1}(n) = 1$ is larger than that for $S_{l-1}(n) = 0$. Now, looking at the recursive relation of (3.8) in Section 3.1, one can see that an upper bound on β_k can be obtained by replacing $\beta_{k+1}(\underline{S}_k)$, $\beta_{k+2}(\underline{S}_{k+1})$, $\cdots$, $\beta_l(\underline{S}_{l-1})$ by $\beta_{max} = \beta_l(S_{l-1}(n) = 1)$ as follows

$$\begin{aligned}
\beta_{l-1}(\underline{S}_{l-2}) &= \sum_{\underline{S}_{l-1}} \beta_l(\underline{S}_{l-1}) P(A_{l-1} = 1|\underline{S}_{l-1}) P(\underline{S}_{l-1}|\underline{S}_{l-2}) \\
&\leq \beta_{l-1,max} \\
&\leq \sum_{\underline{S}_{l-1}} \beta_{l,max} P(A_{l-1} = 1|S_{l-1}) P(\underline{S}_{l-1}|\underline{S}_{l-2}) \\
&\leq \beta_{l,max} \sum_{\underline{S}_{l-1}} P(A_{l-1} = 1|S_{l-1}) P(\underline{S}_{l-1}|\underline{S}_{l-2}) \\
&\leq \beta_{l,max}^2 \\
&\leq [np_0 \frac{g}{b+g} + np_1 \frac{b}{b+g} + g\rho(p_1 - p_0)]^2 \\
&\vdots
\end{aligned}$$

and hence,

$$\beta_k \leq \beta_{k,max} = [np_0 \frac{g}{b+g} + np_1 \frac{b}{b+g} + g\rho(p_1 - p_0)]^{l-k+1}.$$

Similarly the lower bound is obtained by replacing $\beta_{k+1}(\underline{S}_k)$, $\beta_{k+2}(\underline{S}_{k+1})$, $\cdots$, $\beta_l(\underline{S}_{l-1})$ by $\beta_{min} = \beta_l(S_{l-1}(n) = 0)$. This gives

$$\beta_k \geq \beta_{k,min} = [np_0 \frac{g}{b+g} + np_1 \frac{b}{b+g} - b\rho(p_1 - p_0)]^{l-k+1}$$

Carrying the argument further, one gets upper and lower bounds on $P(A^l = I^l)$

$$P(A^l = I^l) \leq \beta_{1,max} = [np_0 \frac{g}{b+g} + np_1 \frac{b}{b+g} + g\rho(p_1 - p_0)]^l \qquad \text{(C.5)}$$

$$P(A^l = I^l) \geq \beta_{1,min} = [np_0 \frac{g}{b+g} + np_1 \frac{b}{b+g} - b\rho(p_1 - p_0)]^l \qquad \text{(C.6)}$$

Bibliography

[1] R. G. Gallager, *Information Theory and Reliable Communication*, John Wiley & Sons, New York, 1968

[2] J. M. Wozencraft, I. M. Jacobs, *Principles of Communication Engineering*, John Wiley & Sons, New York, 1965

[3] E. A. Lee, D. G. Messerschmitt, *Digital Communications*, Kluwer Academic Publishers, Boston, 1988

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

[5] S. Lin, D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, Englewood Cliffs, N.J.: Prentice Hall, ch. 15, 1983

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

[7] M. Schwartz, *Telecommunication Networks*, Reading, Mass.: Addison-Wesley, ch. 4, 1987

[8] A. S. Tanenbaum, *Computer Networks*, Englewood Cliffs, N.J.: Prentice Hall, ch. 4, 1988

[9] R. J. Benice, A. H. Frey, Jr., "An Analysis of Retransmission Systems", *IEEE Transactions on Communication Technology*, Dec. 1964, pp. 135-145

[10] V. K. Oduol, S. D. Morgera, "Performance Evaluation of the Generalized Type - II Hybrid ARQ Scheme with Noisy Feedback on Markov Channels", accepted for publication in the *IEEE Transactions on Communications*, 1991

[11] S. Lin, P. S. Yu, "A Hybrid ARQ Scheme With Parity Retransmission For Error Control of Satellite Channels", *IEEE Transactions on Communications*, vol. 30, pp. 1701-1719, July 1982

[12] H. Krishna, S. D. Morgera, "A New Error Control Scheme for Hybrid ARQ Systems", *IELE Transactions on Communications*, vol. 35, no. 10, pp. 981-990, Oct. 1987

[13] S. Kallel, "Analysis of a Type II Hybrid ARQ Scheme with Code Combining", *IEEE Transactions on Communications*, vol. 38, pp. 1133-1137, August 1990

[14] S. Kallel, D. Haccoun, "Generalized Type II Hybrid ARQ Scheme Using Punctured Convolutional Coding", *IEEE Transactions on Communications*, vol 38, pp. 1938-1946, November 1990

[15] E. N. Gilbert, "Capacity of a Burst Noise Channel", *Bell System Technical Journal*, vol. 39, pp. 1253-1265, Sept. 1960

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

[17] L. N. Kanal, A. R. K. Sastry, "Models for Channels with Memory and Their Applications to Error Control", *Proceedings of the IEEE*, vol. 66, no. 7, pp. 724-744, July 1978

[18] R. H. Deng, D. J. Costello, Jr., "Reliability and Throughput Analysis of a Concatenated Coding Scheme", *IEEE Transactions on Communications*, vol 35, no. 7, pp. 698-705, July 1987

[19] C. Fujiwara, *et al.*, "Evaluations of Error Control Techniques in Both Independent-Error and Dependent-Error Channels", *IEEE Transactions on Communications*, vol. 26, no. 6, pp. 785-793, June 1978

[20] M. Mushkin, I. Bar-David, "Capacity and Coding for the Gilbert- Elliott Channels", *IEEE Transactions on Information Theory*, vol. 35, no. 6, pp. 1277-1288, Nov. 1989

[21] R. Krishnamurthi, S.C. Gupta, "Coding Methods for Portable Radios", *IEEE Vehicular Technology Conference*, 1989, pp. 660-665

[22] D Cygan M Dippold. F Finkenzeller. "Kanalmodelle für die satellitengestutzte Kommunikation landmobiler Teilnehmer". *Archiv für Elektronik und Übertragung-technik*. pp 329-339, Nov./Dec 1988

[23] P Beirouti, H Leib S D. Morgera, "ARQ on Fading Channels", *Proceedings of the Canadian Conference on Electrical and Computer Engineering*, 1991, pp. 66 1.1 - 66.1.6

[24] S Berkovits, E. L. Cohen. "A 3-state model for digital error distributions", Technical Report ESD-TR-67-73, The Mitre Corp., Bedford, MA, 1967

[25] R H McCullough, "The binary regenerative channel". *Bell System Technical Journal*, vol. 47, pp. 1713-1735. Oct. 1968

[26] B. D Fritchman, "A binary channel characterization using partitionned Markov chains", *IEEE Transactions on Information Theory*, vol. 13, pp. 221-227, April 1967

[27] J. H. Berger, B. Mandelbrot, "A new model for error clustering in telephone circuits", *IBM Journal of Research and Development*, vol. 7, pp. 224-236, July 1963

[28] E. O. Elliott. "A model for the switched telephone network for data communications", *Bell System Technical Journal*, vol. 44, pp. 89-119, Jan. 1965

[29] N. Muntner, J. K. Wolf, "Predicted performances of error-control techniques over real channels", *IEEE Transactions on Information Theory*, vol. 14, pp. 640-650, Sept. 1968

[30] J.-P. A. Adoul, "Error intervals and cluster density in channel modelling", *IEEE Transactions on Information Theory*, vol. 20, p. 125, Jan. 1974

[31] J. M. Morris, "Optimal Blocklengths for ARQ Error Control Schemes", *IEEE Transactions on Communications*, vol. 27, pp. 488-493, Feb. 1979

[32] W. W. Chu, "Optimal Message Block Size for Computer Communications with Error Detection and Retransmission Strategies", *IEEE Transactions on Communications*, vol. 22, pp. 1516-1525, Oct. 1974

[33] A R K Sastry, "Improving Automatic Repeat Request (ARQ) Performance on Satellite Channels Under High Error Rate Conditions", *IEEE Transactions on Communications* pp 436-439, April 1975

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

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

[36] J. A. C. Martins, J De Carvalho Alves, "ARQ Protocols with Adaptive Block Size Perform Better Over a Wide Range of Bit Error Rates", *IEEE Transactions on Communications*, vol 38, pp. 737-739, June 1990

[37] R. A. Comroe, 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

[38] C. H. C. Leung, Y. Kikumoto, S. A. Sorensen, "The Throughput Efficiency of the Go-Back-N Scheme Under Markov and Related Error Structures", *IEEE Transactions on Communications*, vol. 36, pp. 231-234, February 1988

[39] D. Towsley, "A Statistical Analysis of ARQ Protocols Operating in a Nonindependent Error Environment", *IEEE Transactions on Communications*, vol. 29, pp. 971-981, July 1981

[40] L. R. Lugand, D. J. Costello, Jr., R. H. Deng, "Parity Retransmission Hybrid ARQ Using Rate 1/2 Convolutional Codes on a Nonstationary Channel", *IEEE Transactions on Communications*, vol. 37, pp. 755-765, July 1989

[41] U. N. Bhat, *Elements of Applied Stochastic Processes*, New York: John Wiley and Sons, ch. 4, pp. 105-109, 1984

[42] L. R. Rabiner, B. H. Juang, "An Introduction to Hidden Hidden Markov Models", *IEEE ASSP Magazine*, pp. 4-16, January 1986

[43] P. R. Drouilhet, Jr., S. L. Bernstein,, "TATS - A Broadband Modulation - Demodulation System for Multiple Access Tactical Satellite Communication", *1969*

IEEE Electronics and Aerospace (EASCON) Conf. record, Washington, D C., pp. 126-132, October 27-29, 1969

[44] D G Smith, "Spread Spectrum for Wireless Phone Systems: The Subtle Interplay between Technology and Regulation", *IEEE Communications Magazine*, February 1991

[45] R C. Dixon, *Spread Spectrum Techniques*, IEEE Press, New York, 1976