PERFORMANCE EVALUATION OF ERROR CONTROL METHODS WITH NOISY FEEDBACK:

The Generalized Type-II Hybrid ARQ Scheme and the Selective Repeat ARQ Scheme on Markov and Gilbert Channels

BY

Vitalice K. Oduol,
B.Eng. (Hons), M. Eng., (McGill)

A Thesis Submitted to the Faculty of Graduate Studies and
Research in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

Table of Contents

Table of Figures

Table of Tables

ABSTRACT

Noise affects all channels used for communication, causing errors in the transmitted messages, thereby creating the need for error control. With a few exceptions, the methods of error control that use a feedback channel have assumed this channel to be error-free in their analysis. This thesis considers the effect of feedback channel errors in two methods of error control, the Generalized Type-II Hybrid ARQ Scheme and the Selective ARQ Scheme. Expressions are derived for the throughput efficiency, the probabilities of undetected error, correct delivery, and packet loss. It is found that noise in the feedback channel lowers the throughput efficiency, causes the loss of packets, and increases the number of undetected errors. Methods are suggested for reducing the degradations caused by feedback channel errors. Simulation results are also provided. From the results and the analysis, the conclusion drawn is that feedback channel noise has an important effect on the performance of ARQ schemes, and should be taken into account to provide a more realistic assessment of their performance.

Résumé

Le bruit affecte tous les canaux de communication, ce qui se traduit par des erreurs dans les méssages transmis. Il y a donc un besoin pour les techniques de correction d'erreurs. À quelques exceptions, près toutes les techniques de correction d'erreurs qui utilisent la rétroaction dans le canal font l'hypothèse que celui ci est sans erreurs pour en faire l'analyse. Cette thèse étudie l'effet de la rétroaction dans le canal sur deux techniques de correction d'erreurs, la méthode hybride ARQ type II généralisé et la méthode du ARQ sélectif. On y présente des expressions pour l'efficacité du débit, la probabilité des erreurs non-détectées, des méssages reçues correctement et des paquets perdues. On y montre que le bruit dans la boucle de contre-réaction réduit l'efficacité du débit, cause des pertes de paquets et augmente le nombre d'erreurs non-détectées. Des moyens sont présentés pour réduire la dégradation causée par les erreurs dans le canal avec rétroaction. Des résultats obtenues par simulation sont présentés. À partir des résultats, et de l'analyse de ceux-ci, on conclut que le canal avec rétroaction a un effet important sur la performance des méthodes de type ARQ et que ceci doit être pri en compte afin d'évaluer de façon réaliste leurs performance.

ACKNOWLEDGEMENTS

In the writing of books, as in much else, no man is an island, entire to himself; we all must borrow from somewhere, consciously or otherwise. We all hope that when we do borrow, however widely, that we do so wisely. Whatever the errors of this work, they would have been far greater were it not for countless others, many of whom will remain unmentioned. I would like to thank Prof. S. D. Morgera for his support and for suggesting that I investigate the effects of feedback errors on ARQ performance. I thank the anonymous reviewers of the parts of this work that were released for publication[1]; their comments have been most useful. I am also indebted to Prof. P. E. Caines, who, although an outsider to the area of this work, gave a detailed review of an earlier draft of this thesis, and gave comments that have been very effective in shaping the final result.

I thank my fellow graduate students with whom I spent those trying hours of difficulty, and whose sense of humour and forbearance helped a great deal to make the task bearable. They are too many to mention by name here, in the limited space that is available. Finally, I thank my family for their boundless patience, love and understanding throughout my studies.

1 For the appropriate title, see reference [95].

CLAIM OF ORIGINALITY

To the author's best knowledge:

1. The treatments of the GH-II ARQ Scheme and the Selective Repeat ARQ Scheme with noisy feedback are given here for the first time.

2. The analysis of the above two schemes as presented here is new; the expressions obtained here for the throughput efficiency, the probabilities of undetected error, and correct delivery are given here for the first time.

3. The demonstration of the loss of packets in both the ARQ schemes is also new. The discussion of the transmissions (spurious deliveries) that are made of packets that have already been delivered is new here, and so are the expressions derived for these events.

4.The use of a predictor with the GH-II ARQ technique is an original contribution of this thesis.

5. The use of codes with invertible subblocks to avoid packet loss is an original contribution of this work.

6. Finally, the main contribution of this work is that it has demonstrated that the performance evaluation of error control methods is more realistic when the feedback noise is taken into account. Besides, the work has shown that expressions can be derived for the relevant performance parameters.

CHAPTER I
INTRODUCTION

Communications channels contain errors that necessitate some form of error control to ensure acceptable reliability. Shannon's theorems [1] demonstrate the existence of codes that can provide error-free communication, provided the transmission rate is below the capacity of the channel in question. Since Shannon's theorems are existence statements, they do not give suggestions on how to construct (or implement) the codes that yield the promised performance. As a result of Shannon's work, many people have been involved in seeking error control methods that would satisfy the promises of the theory. Many methods of error control have been devised. In general, a specific signal is assigned to one of M possible messages. The selecting rule for making this assignment is the *code*. The first codes that were developed were block codes, in which a sequence of k information symbols is encoded into a longer sequence of n symbols ($n > k$). The added n-k symbols are called parity symbols. There are good block codes for which the post-decoding error rate decreases exponentially with the length n of the block.

The other class is that of convolutional codes, which have been studied greatly. The applications of these codes to communication systems can be found in [2]. Briefly, in convolutional codes the present information symbol is combined with a certain number of previous symbols to compute the channel outputs. In this way the data stream is encoded continuously, rather than on a block basis. Even with the most powerful codes, block or convolutional, it is found that no practical coding scheme has so far been able to achieve the capacity and error-free transmission. In the hope of obtaining improved performance with low complexity, methods that use a feedback channel have been proposed. Feedback does not increase the capacity of the channel [3]; it serves as a means of lowering the probability of undetected error.

All these methods aim at matching the error control method to the channel. The *optimality principle* given later in the thesis is a statement of the criterion used to develop the methods. While the simplicity of nonadaptive methods is practically beneficial, better performance is provided by adaptive methods. This point is discussed in more detail later.

Those methods that use a feedback channel have almost invariably assumed it to be error-free in their analysis, ignoring the effects that errors in this channel have on the results. In reality, no channel is error-free. It is therefore worthwhile to inquire into the performance degradations that will arise due to the errors in the feedback channels. Among the exceptions is Turin, who in Chapter 5 of his book [4] gives a brief discussion that assumes the possibility of errors in the feedback channel.

The justification for ignoring the effects of feedback channel errors is that the data rate in this channel is much lower than in the forward channel; so it is possible to find a long enough code to guarantee virtually error-free performance in the return channel. Indeed, there are many cases in which this applies. But there are yet many other practical communication systems that are largely two-way, requiring as much information transmission in the feedback channel as in the forward channel. The acknowledgement from the receiver is usually made part of a frame sent with data in the return channel. The return channel will be no better than the forward one. Therefore, the amount of error correction coding that can be placed in the return channel is as limited as in the forward channel. The present thesis takes into account the effect of these feedback channel errors. Expressions are derived for the relevant performance parameters on a Markov channel.

There are two fundamental error control techniques in digital communication systems: forward error correction (FEC) and automatic repeat request (ARQ) schemes [5]. In FEC schemes, an error correcting code is used at the transmitter. The receiver decodes the arriving data, and assumes that the code has corrected any errors that may have occurred. In an ARQ system, a single high-rate error-detecting linear block code is used. When a message of k bits is ready for transmission, n-k parity bits are added to it to form a codeword, which is then transmitted in the channel. If the receiver detects errors in this transmission, a request is made for a retransmission; otherwise the next packet is transmitted. In principle, this process continues until the packet is correctly received. When FEC and ARQ are combined, the result is what is now known as type-I hybrid ARQ, in anticipation[1] of type-II hybrid ARQ, which combines the two methods with some redundancy retransmission. A full discussion of these ARQ methods is given in Section 2.7 (entitled *Hybrid ARQ Schemes*).

The performance of the type-II ARQ scheme depends very much on the choice of the error correcting code C_{EC}. Both the performance of the type-II ARQ scheme and its advantage over the type-I ARQ scheme are given in [6]. A generalized version of this is presented in Chapter III, where a method of analysis is developed and used to obtain expressions for the throughput efficiency, the probability of undetected error, and the other parameters needed in evaluating the performance of

1 We say "in anticipation" here only for the order in which we present the concepts. Chronologically, the type-I scheme was known as "hybrid ARQ scheme" until it was found that another (the present type-II hybrid ARQ) could be developed. The latter was named type-II and its predecessor, type-I.

the GH-II ARQ scheme. In a parallel fashion, Chapter IV analyses the selective repeat ARQ scheme with noisy feedback. To provide a means of comparison, Chapter V discusses the situation without noise in the feedback channel. Chapter VI looks at the performance on the Gilbert channel. It is established in this chapter that the expressions derived for the Markov channel cannot be used on the Gilbert channel, since the resulting error process in the Gilbert channel is not first-order Markov. Chapter VII presents the simulations on the Gilbert and Markov channels. One channel captures the statistical dependence at the bit level, the other captures the dependence at the block level. Besides, either method can be particularized to cover the independent-error channel.

In Chapter VIII, some methods are presented for improving the performance under noisy feedback. These are the use of a predictor with the GH-II ARQ scheme, the use of invertible codes to prevent packet loss, and the use of a threshold at the transmitter to reduce spurious transmissions. The corresponding improvements are also discussed. Finally, Chapter IX gives the summary and conclusions. Material that was considered necessary, but whose details seemed a great digression from the objective of the thesis has been relegated to an appropriate appendix. The KM codes, which are used in the GH-II ARQ scheme discussed here, are presented in Appendix A. This discussion of the codes is only a brief one; fuller knowledge of these codes can be obtained from [7] and [8]. Other appendices are referred to in the body of the thesis.

On channels where the error bursts (of maximum length B) are separated by a gap (of minimum length G), *burst trapping* techniques can be used. Early contributions to this area are found in the works of Gallager [9], Forney [10], Burton, Tong and Sullivan [11-13]. In the basic form, an information vector is transmitted two times, the first time in the information part of a codeword, and the second time it is added to the parity part of the next codeword. When the first transmission is received in error, it is assumed that the second will be error-free, hence the requirement that the gap length G be longer than some minimum. The information part of the second transmission is then extracted and given to a local encoder, which produces the parity bits. These parity bits are then used to recover the previously transmitted information. Burst trapping methods have been generalized [12-14] to cope with a wider variety of channel types. These techniques differ markedly from the GH-II ARQ scheme described here, in that they require certain statistical properties in the bursts, requirements that are not needed in ARQ systems [15].

1.1 Where Can We Use These Methods ?

Some practical channels where the methods of this thesis are applicable are satellite channels [5] and HF troposcatter channels [16,17]. The satellite channel varies from a very noisy state (during rain and bad weather) to a more benign state (during good weather). The variation is slow enough for the GH-II ARQ to provide a reasonable performance improvement. The parameters used to model the HF channel can be found in [16,17]. Since these models specify a Markovian dependence, the methods of this thesis would be applicable.

In variable rate ARQ systems, the return channel is used to adaptively control the forward transmission rate [18]. This form of ARQ may be attractive for satellite links where the feedback delays are large and the changes in channel characteristics, such as deep fades, are slow. In the same paragraph of the cited reference[2], it is said that the principal drawback of such adaptive rate systems is an increase in system complexity, since each allowed rate may require a distinct modulator and demodulator.

However, there now exist some codes that can have variable rates while maintaining nearly the same complexity. With these codes, the same modulator-demodulator pair can be used and the transmission rate altered by varying the code parameters.

Examples of transmission systems where the GH-II ARQ can be used are: HF, land mobile, cellular radio, terrestrial digital radio, satellite communication systems. The corresponding frequency bands are given below.

1.1.1 HF Communication Channels

Highly reliable, long-distance communication can be realized using HF radio waves (3-30 MHz) [19]. These links take advantage of the fact that HF signals can be reliably reflected by the ionosphere and returned to the earth at great distances from the transmitter. Distances of 6000 km are realizable.[3] As the frequency increases, tropospheric impairments typically increase while ionospheric effects decrease [19].

The dominant effects at HF frequencies are ionospheric. The E Layer (90-140 km) of the ionosphere exhibits sporadic activity, requiring appropriate precautions for reliable link operation. For example, the operating frequency may have to be altered a few times over a 24-hour period to coincide with the ionospheric changes [19]. The remaining drawback is from multipath fading due to reflections from buildings and other electrically conducting objects within a few miles from the receiving antenna.

---

2 Bhargava et al., p.337.

3 Microwave tropospheric scatter techniques in which line-of-sight transmissions are scattered by refraction and reflection in the troposphere (sea level to about 11km) fit somewhere between LOS and HF.

1.1.2 Mobile Radio Communication Systems

Mobile radio is used for transmitting weather, road and traffic information. The frequencies used here are:

VHF (25-50 MHz)
VHF (150-174 MHz)
UHF (450-470MHz)[4]
800 (806-947 MHz)

The US Weather Bureau provides weather information using 160 MHz. Small suburban airports and others interested in timely weather information can monitor these transmissions. A number of cities broadcast road and traffic advisory information using low-power transmitters.

The main impairment is *shadowing*, or substantial transmission loss due to hills, buildings, and trees in the transmission path. It increases with frequency and is to some extent offset by reflections from objects not located in the direct path. Troposcatter is seldom a cause of interference in mobile services, but can be used for communication between fixed stations [20].

1.1.3 Cellular Radio Systems

The frequencies used are:

UHF (825-848 MHz) transmitter
UHF (868-895 MHz) receiver

The main propagation impairments are:

1. multipath propagation due to scatter from nearby obstructions
2. multipath propagation due to large echoes from distant reflectors
3. shadowing of the direct path.

The effect of multipath propagation is a standing-wave pattern in space determined by the amplitude and phase relationship of the direct and reflected and/or scattered energy components. A vehicle moving through this pattern will experience short-term fluctuations in signal intensity, typically with a period of a fraction of a second.

Shadowing, by contrast, produces an overall reduction in signal intensity which changes more slowly with time as the vehicle moves about.

4 The allocated frequency range is actually 150-512 MHz.

1.1.4 Terrestrial Digital Radio

This type of system uses microwave frequencies. Here, since the transmission is fully digital, ARQ methods can provide performance gains for all the transmissions, the level of performance improvement being determined by the channel conditions. This can be said also for satellite communications links, described below.

The main impairments are multipath fading, sky noise, radio interference, and rain attenuation. These impairments vary sufficiently slowly for the predictor described in the sequel to improve the performance of the GH-II ARQ scheme. The radio interference will vary from one region to another, peaking around urban centres.

1.1.5 Satellite Communication Links

Examples of satellite communication systems given in [18] has a signalling rate of $R_S = 10^4$ bits/s and a round trip delay of $\Delta T = 512$ ms. If we take a packet size of $B_P = 504$ bits (as used in the present thesis), the round-trip delay D_P (in packet times) will be equivalent to

$$D_P = \Delta T\left(\frac{R_s}{B_p}\right) = 0.512\left(\frac{10^4}{500}\right) = 10.24 \text{ packet times}$$

This number is well below the threshold of $D_P = 15$, beyond which the predictor improvement becomes increasingly negligible. However, the scheme without a predictor is still applicable.

1.1.6 Remarks

The above list of applications, although not exhaustive, covers the essential portion of the spectrum used for communications. The methods of this thesis are applicable in each case. The only note of caution that must be sounded is that when the links get too long, the method with prediction may not yield sufficient performance improvement. Table 1.1 lists some qualitative summary of what one might expect when the predictor is used.

Table 1.1 Summary of Predictor Performance

Application	Propagation Impairments	GH-II ARQ with predictor
HF	E-Layer absorption	The E-Layer effects are slowly-varying and the propagation delays are reasonably short. Consequently, sufficient performance improvements are possible.
Mobile Radio & Cellular Radio	shadowing multipath radio interference	Currently, most of the traffic in this application is analogue; the control signals are digital. It is for these signals that ARQ can be used; therefore, the improvement in performance is limited. But in future, when the transmissions become increasingly digital, there will be greater gains when GH-II ARQ is used.
Terrestrial Microwave	multipath rain attenuation sky noise radio interference	Some improvement is possible, due to the short distances between repeaters. The dominant limitation will result from multipath fading.
Satellite Links	multipath rain attenuation sky noise	Increased buffering will be required at both ends if the link is too long. Predictor Improvement diminishes with the length of the link. For long links, the scheme with predictor is no better than the one without.

CHAPTER II
BACKGROUND

As mentioned at the end of the preceding chapter, there are two areas where the applicability of the methods of this work is obvious—the satellite channels and the HF troposcatter channels. There are other areas that can be studied in order to determine the applicability of these methods. This will require data collection to determine the model parameters. It is clear, however, that whenever the channels do not vary too rapidly[5], these methods will yield an improvement.

A look at the recent issues of the IEEE Transactions on Communications indicates that error control is receiving increasing attention. A component of error control is channel coding, which is fast becoming a widespread technique for improving the reliability of data in digital communication systems. However, the design and fabrication of decoders remains difficult. The reason for this is that the extensive memory required takes much chip area. Lavoie, Haccoun and Savaria [21,22] have considered architectures that would facilitate the implementation of CODECs for convolutional codes[6]. Continuing work in code design, and advances in high speed processing technology will soon make it possible to implement fast CODECs for both convolutional codes and block codes.

Errors in communication channels occur due to noise and other channel impairments whose combined effect is to generally cause errors to occur in bursts separated by long error-free gaps. Over a given period of time, the channels tend to be either very good or very bad. Since neighbouring time intervals have similar error rates, this biased behaviour has been termed memory [23]. This memory can be exploited to control the errors in some manner to ensure a tolerable error rate. The goal of error control is to achieve an acceptable error rate in the most cost-effective manner. An indispensable tool in this is the channel model.

2.1 Channel Models

The manifestations of nature are so abundant in facts that we must all the time find some orderly way of representing the facts so as to make any meaningful statement about them. We are forced toabstract from nature what we call representations or "models." These models are understood to be adequate representations of those aspects of nature in which we are at present interested. Any model will therefore have its scope and obvious limitations, as specified in the assumptions in its

5 That is, whenever the channel noise level stays fairly constant within a period of at least 15 packet transmission times.
6 A CODEC is a two-way device that performs the operations of coding and decoding.

formulation. A general discussion of models, with very entertaining examples, can be found in Casti's [24] book.[7] For our purposes here we shall focus our attention on the models of channels with memory, and their use in developing error control methods.

In principle, it is possible to evaluate the performance of error control methods without recourse to a model. For example, raw data from a channel can be used to test the performance of a code. An approach of this type normally requires an enormous amount of computation time. As an example [25], a tape drive specifies bit error rates as low as 10^{-12}. With an average transfer rate of 5 Mb/s, it may take 20 days to collect enough data for a reliable bit error rate estimate. All this time is spent on one code. On the other hand, a model can be used to evaluate the performance of a code on a channel in *minutes*. From such an evaluation, it is possible to predict the post-decoding bit error rates. Furthermore, when many codes are to be tested, using the model it is possible to select the one most appropriate for a given channel and specified constraints.

Many models have been proposed for channels with memory. In an effort to develop models that adequately represent the behaviour of real channels and are mathematically tractable, the search for models took many directions. According to Kanal and Sastry [23], the models fall into two categories: *generative* and *descriptive*. Generative models are those that specify an underlying mechanism that produces error sequences similar to those being modelled; descriptive models are those that give the structure of the error sequence in terms of some statistics. The utility of models in this area is exemplified by the work of several authors. Knowles and Drukarev [26] considered estimating bit error rates on channels with memory. They used a model developed by Fritchman [27], called the Fritchman model. The authors report that they used this model "because of its simplicity and fairly good agreement with experimentally observed channel statistics." Tsai [16] applied some known models to data obtained on an HF channel. He obtained the corresponding values for the parameters of the models. These values can now be used by others interested in applying the models to their studies. MacManamon [17] obtained a similar set of parameter values assuming a bit-level Markov model.

The models assumed by these authors have been those whose parameters can be uniquely determined from the observed channel data [23]. Yet another contribution to the field is that of Vucetic and others [28], who proposed a quasistationary model which they analyzed and from which they obtained expressions for the performance parameters. The conclusion drawn from the survey

7 This book does not have the final word on models; it offers discussions, examples, and exercises intended to provoke the reader to develop his own view (his own alternate reality).

of the work on models is that concentration has been on applying existing models to given data in order to find the corresponding parameter values. From these values, the experimenters attempt to operate the models to see if the observed output is in agreement with the data. In this way, it has been possible to find the models most appropriate to a given channel. The problem faced by these workers is that while the model gives a representation of the entire time behaviour of the error process, extending into the remote future, the data available is finite. A further problem is determining the length of the data record that will give adequate parameter estimates for the models. These problems notwithstanding, people have tried to obtain working models. Analytic models for some real channels can be found in [29] and [30].

2.2 The Error Process

The error process in a communication link can be viewed as a discrete-time binary stochastic process [20]. That is, it can be viewed as a family of binary random variables $\{\mathbf{E}_t;\ t \in \mathbf{Z}\}$, where $\mathbf{Z}$ is the set of all integers. A realization of the error process is represented by the sequence $\ldots, e_{-1}, e_0, e_1, \ldots$ This can be related to the channel output sequence as follows. Let $\{a_t\}$ be the input sequence and let $\{b_t\}$ be the corresponding output sequence. Then $b_t = a_t + e_t$. The addition is done modulo-2. The characterization of the error process as a binary stochastic process in discrete time is assumed almost universally in the analysis of error control methods. Some approaches incorporate modulation in ARQ error control analysis; notable among are the works of Benneli [31], and Kasami and others [32], who consider a coded modulation scheme with code concatenation.

Here, we use the binary process approach. We shall consider three channel models in this work: the partitioned Markov Chain Model with one error state, the Gilbert Model (a special case of the former), and the Markov Error Model, developed from the descriptive category.

2.3 The Partitioned Markov Chain Model

This model is characterized by a set of states $\{1,2,3,\ldots,N\}$ and a state transition probability matrix $\mathbf{P} = (p_{ij})$. The state transitions of the chain are synchronized with the error digit transitions in the channel. Let q_k be the probability of error when the chain is state k. Then in the most general setting, one would have $q_k \neq 0, \quad \text{for} \quad k \in \{1,2,3,\ldots,N\}$. This would yield the unpartitioned Markov chain model. Unfortunately, determining the parameters of the model from channel data is quite costly [30]. To create a model that is not as costly, and still represents the real channel reasonably

$$\mathbf{P} = \begin{pmatrix} 0.66 & 0.00 & 0.34 \\ 0.00 & 0.9991 & 0.0009 \\ 0.44 & 0.34 & 0.22 \end{pmatrix}. \tag{2.2}$$

Turin [4] gives a discussion of single-error-state models. He provides numerical results for the matrix P for such channels as the troposcatter and T1 digital repeatered line. The results for the T1 line can also be found in the work of Brilliant [33]. By setting N=2 in the general single-error-state model, we obtain the Gilbert model, first proposed by Gilbert [34][8]. Gilbert used this model to model the data supplied by Alexander and others [37]. Figure 2.2 illustrates the model proposed by Gilbert. If in this model we set $q = r$, we obtain the random error channel, usually called the binary symmetric channel (BSC).

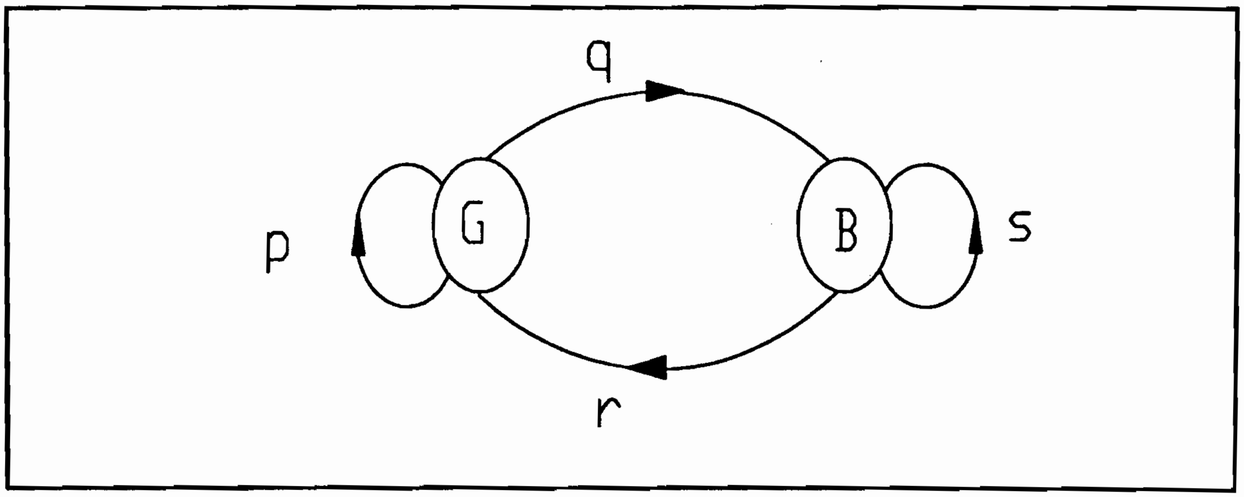

Figure 2.2 The Gilbert Channel Model.

2.4 The Markov Error Model

Following the analysis given by Leung, Kikumoto and Sørensen [38], in which a Markov error model is assumed, we envisage here the number of errors in a block of size N as a realization of a Markov chain. Specifically, let $e_k(N)$ be the number of errors in such a block at a general time k. Then the model we seek is such that the sequence $\{e_k(N); k=\ldots,-1,0,1,\ldots\}$ is a Markov chain. Since

8 This was later modified by Elliott [32,33] to form what is called the Gilbert-Elliott model, which has errors in each state.

we are giving here the specification of the error sequence, this would be termed a descriptive model, in Kanal and Sastry's terminology, referred to earlier.[9] As it stands, the underlying space of states is N-dimensional. If N is about 500, as it usually is, then the state transition probability matrix will have $N^2 = 250000$ entries. This will be too large. To trim down this number, the state space is partitioned into three classes. Figure 2.3 shows the state transition diagram of this model, where p_i, $i=1,2,3$ is the probability of error in the ith class.

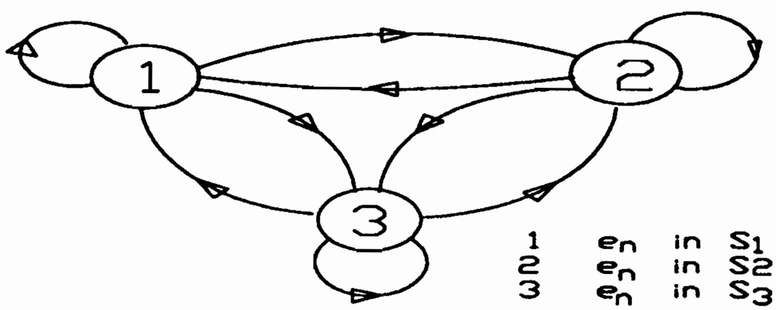

Figure 2.3: Transition Diagram For the Markov Error Model.

The use of a Markov model is also found in the work of Meyer and Sastry [39], who considered an end-to-end burst error link, generating results that, as they report, are of particular interest to ISDN. Since their work does not address ARQ schemes, we will not discuss their details here. For the same type of link, Lai and others [40] have analyzed the undetected error probability. In a practical situation, the decision to use a Markov model can be made by examining the autocovariance function of the error process [38]. The autocovariance function can be estimated from the observations according to the equation

$$\hat{\rho}(t) = \frac{c(t)}{c(0)}, \tag{2.3}$$

where $c(t) = \frac{1}{\text{M-t}}\left[\sum_{n=1}^{M-t}(e_n - \hat{\mu})(e_{n+t} - \hat{\mu})\right]$, and $\hat{\mu} = \frac{1}{M}\sum_{n=1}^{M} e_n$.

9 See Channel Models above (Section 2.1).

If this estimate is an exponentially decaying function of the lag, the channel can be taken as a Markov channel, and the model of this section will be most appropriate. An example is given later in which a process is constructed that has a decaying autocovariance function of the type required here. If the autocovariance function closely resembles a delta function, the channel is best represented by a random noise model.

An autocovariance function of the above form is reported in [41] to be very common and to be a very good approximation to many random phenomena encountered in practice. More detailed studies of the modelling of errors on real channels are found in [42].

2.5 Compound Models

Models that combine two or more simpler ones have also been considered. The hope is that the compound model will have the strengths of its constituents in giving a more realistic representation of the channel. One method models the channel noise as a sum of a random Gaussian noise component and an impulsive noise component[10]. The impulsive component is a series of impulses whose occurrence times are taken to be the arrival times of a Poisson process. The ith impulse $\delta(t-t_i)$, which occurs at a random time t_i, is considered to have a random "tail" area A_i. The areas are considered to be statistically independent of each other and of the arrival instants. The noise is characterized by the number of impulses per symbol duration. This number is denoted by γ[11]. Highly impulsive environments will have $\gamma \gg 1$. Several applications with large values of γ have been reported [43]. The value of γ will restrict the choices of error control coding method to be used. For example [44], sequential detection does not work well for very large values of γ.

To characterize the clustering phenomenon, a compound Poisson model has been proposed. The error cluster arrivals are represented by a Poisson distribution with parameter m_1, and the number of errors within each cluster also have a Poisson distribution with parameter m_2. The chi-square method is used to test the validity of the model in representing the actual error process in a digital transmission system. Results of tests conducted in France on 2 Mb/s and 140 Mb/s digital links, from which a lot of data has been gathered, suggest that this compound Poisson model (sometimes called the Neyman model) does not adequately represent the bit errors, but represents

10 The intention here is to represent the presence of the random and the impulsive components in one model. The task then is to determine the parameters that are relevant to this type of model, and then to devise experiments that can be used to find their values.

11 The symbol γ as used here has no connection with its use later in characterizing the Gilbert Channel (in Chapter 6).

well the occurrence of errored seconds in a one-minute interval. Another approach models the error process at the bit level. As yet no model has been found that encompasses all cases of errors on a digital communication channel [45].

2.6 Error Control Methods

It is assumed in this thesis that the messages have been broken into packets. A packet is a standard unit used in communication systems. The messages may be of arbitrary length. For purposes of coding, each channel will tend to have its own optimum packet length. For example, for an error rate of 10^{-4} on an additive white Gaussian noise channel, Bhargava and others [18] give a length of 500 bits for the packets. The same value can be inferred from Martins and others [46], who adaptively measured the channel bit error rate and used it to adjust the packet size to yield the optimum throughput. All this is in agreement with the derivations of the optimum packet length found in [47].

There are two fundamental error control techniques in digital communication systems: forward error correction (FEC) and automatic repeat request (ARQ) schemes [5]. In any ARQ system, a single high-rate error-detecting linear block code is used. When a message of k bits is ready for transmission, n-k parity bits are added to it to form a codeword, which is then transmitted in the channel. If the receiver detects errors in this transmission, a request is made for a retransmission; otherwise the next packet is requested. In principle, this process continues until the packet is correctly received. In FEC, the transmitter employs an error-correcting code (block or convolutional) to combat transmission errors. The difference between this method and ARQ is that here there are no retransmissions, and so no feedback channel is required. Since the probability of a decoding error is always higher than the probability of undetected error (using the same code), FEC schemes cannot provide as high a system reliability as do ARQ schemes. For FEC schemes to have the same undetected error probability as ARQ, a longer, powerful error-correcting code is used to correct most of the error patterns. Owing to these factors, ARQ schemes are generally preferred.

To evaluate the performance of error-control methods, two parameters usually considered are the throughput efficiency and the data reliability. The throughput efficiency is the proportion of the transmission time spent in transmitting useful information; the reliability is the probability that a received message contains no errors. This is called the probability of correct delivery, which is complementarily related to the probability of undetected error.

As a consequence of feedback channel errors, two other parameters to examine are the probability of packet loss and the probability of a packet being transmitted unnecessarily. A packet is lost when the errors in the feedback channel cause a retransmission (R) request to be converted into an acknowledgement (A). This (R→A) error causes the transmission to begin sending the next packet instead of retransmitting the present one. The receiver considers this packet as a retransmission and proceeds with decoding. This discussion is treated in greater detail in Sections 3.1 and 4.1.

Since there are no retransmissions in FEC, the throughput efficiency is constant, and is set by the code rate, independent of the channel conditions. The reliability of FEC schemes is highly sensitive to the channel conditions, and degrades as the channel gets poorer. Although ARQ schemes have a lower throughput efficiency, they can be made to have increased reliability, regardless of the channel conditions. FEC and ARQ, therefore, are two methods which tend to affect one of the two parameters and leave the other one more or less unchanged as the channel error rate fluctuates. In situations where the error rate is too high to guarantee acceptable throughput using ARQ alone, a combination of FEC and ARQ is used. Such a scheme is appropriately termed hybrid ARQ [5]. Naturally, this scheme inherits the advantages of both techniques.[12] High reliability is achievable as a result of error-detection and retransmission (the ARQ contribution), and high throughput efficiency is possible through the correction of frequent error patterns (the FEC contribution).

Most channels used for communication have error rates that vary with time from a very poor state to a very good state, as mentioned earlier. The methods of error control have tended to select codes that provide acceptable performance when the channel is in its worst state. Such worst-case codes must have low rates. Communications channels generally hit their worst state only a small fraction of the time. A code matched to the worst-case channel conditions will, therefore, result in a low efficiency. This is stated more formally by the optimality principle that is soon to follow. We first define $T_i = \{ x: t_{i-1} < x \leq t_i \}$, where t_i is the maximum number of errors that can be detected by the ith code.

12 It is quite fortunate here that we have two methods whose combination retains their individual strengths and overcomes their individual weaknesses.

The Optimality Principle: If on the average, the error process stays in T_i with probability π_i, then given a set of codes with rates R_i, i=1,2,...,m, where the ith code is suited to the ith channel state (T_i), the optimum code rate R_{eq} is given by

$$R_{eq} = \sum_{i=1}^{m} \pi_i R_i, \tag{2.4}$$

where m is the ratio of the length of the longest code in the set to the number of information bits. In Chapter III, this is referred to as the depth of the code.

Proof: Suppose the code assignment probabilities are instead b_1, b_2,..., with some n_j for which $b_{n_j} \neq \pi_{n_j}$. Then let us form the difference

$$\Delta R = \sum_{i=1}^{m} b_i R_i - \sum_{i=1}^{m} \pi_i R_i. \tag{2.5}$$

This gives

$$\Delta R = \sum_j (b_{n_j} - \pi_{n_j}) R_{n_j} + \sum_l (b_{n_l} - \pi_{n_l}) R_{n_l}, \tag{2.6}$$

where the sum over j is for those terms for which $b_{n_j} - \pi_{n_j} > 0$ and the sum over l is for terms for which $b_{n_l} - \pi_{n_l} < 0$.

Let ΔN_1 be the change in the number of errors correctable in those cases corresponding to the first term above, and ΔN_2 the change in the number of errors correctable in the second set of cases. Since the rate of the resulting code is higher in each term of the first sum, the resulting error correcting power will be lower for that sum. Therefore, $\Delta N_1 < 0$. Now let us consider the second sum. The rate of the code is lower in each term of the second sum. The number of errors correctable would be higher. But since the number of errors that occur will not exceed the level correctable by the original assignment (using π_i's) we must have $\Delta N_2 = 0$, because although we have a more powerful code assignment, the channel does not need the extra power. If we now denote by ΔN the change in the number of errors correctable, then

$$\Delta N = \Delta N_1 + \Delta N_2 \tag{2.7}$$

from which we see that $\Delta N < 0$ so long as there is at least one case for which $b_i \neq \pi_i$. In other words, the optimum assignment probabilities for any set of codes with rates R_i is the one for which $b_i = \pi_i$. That completes the proof of the optimality theorem.

Corollary 2.1: A code designed for the worst case channel condition will have a lower throughput efficiency than the optimal code.

Proof: The worst-case assignment is done by choosing $b_m = 1$, and $b_i = 0$ otherwise. This implies that

$$\Delta R = (1-\pi_m)R_m - \sum_{i=1}^{m-1} \pi_i R_i \quad (2.8a)$$

$$= R_m - R_{eq}, \quad (2.8b)$$

which is necessarily positive, since $R_m > R_{eq}$. Therefore the throughput efficiency will be lower than for the optimal assignment, which completes the proof.

One channel that has a varying signal-to-noise ratio (SNR)—and therefore a varying error rate—is the meteor burst channel, which has an SNR that decays exponentially with time, i.e.,

$$\text{SNR} = (SNR)_0 e^{-a(t-t_0)} \qquad t > t_0, \quad (2.9)$$

where t_0 is the starting time for signal transmission [48]. The parameters $(\text{SN R})_0$ and a may be random variables. One task in applying the methods of error control would be to collect enough data to characterize the statistical nature of these parameters. The author of this thesis had the opportunity to survey the work in this area, some of whose findings and projections are reported in [49]. Metzner [48] and Mui [50] have addressed the issue of the possible coding strategies for the meteor burst channel. Pursely and Sandberg [49] have discussed the use of incremental redundancy in meteor burst channels. This area is still new, and a lot of work is still needed before the appropriate error control methods can be established.

At present, the techniques of error control that approximate the optimality best are the generalized ARQ techniques (discussed later). These perform far better than many alternative approaches. Fantacci [52] gives some discussion of this fact, with some convincing details.

Many attempts have been made to improve the performance of ARQ schemes. The use of soft-decision decoding with the GH-II ARQ scheme appears in [53], where the advantage of soft-decision decoding is observed in the throughput efficiency. Other methods using soft-decision are presented in [54] and [55]. Fantacci [56] has demonstrated the advantage of soft-decision decoding in a generalized compound go-back-N (CGBN) ARQ technique. More recently, Martins and Alves [46] have shown that by monitoring the bit error rate in the channel, it is possible to adaptively vary block size (packet size) to bring about a performance improvement in ARQ schemes. They have shown that around the optimum block length, the throughput curve is fairly flat; this fact allows even a sub-optimal packet length to yield a practically high throughput efficiency, provided the length used is not too far from the optimal. There are other methods such as code combining, which was proposed by Chase [57], in which packets received in error are stored and combined with later transmissions, while a running measure is computed each time. When the measure exceeds a threshold, a decision is made regarding the packet. In another paper [58], code combining and the selective repeat ARQ are combined, and a performance improvement is found. Kallel [59] has shown that the merits of Type-II Hybrid ARQ[13] can be combined with those of code combining to produce even greater performance improvements. Earlier, Kallel and Haccoun [60] analyzed an error control system that involved sequential decoding and code combining, from which they showed that the throughput efficiency of the ARQ schemes were much higher with code combining than without it.

In addition to the above methods, other adaptive ARQ methods have been developed to approximate the optimality referred to above as nearly as possible. Vucetic and others [61] have given an algorithmic approach to the synthesis of an adaptive ARQ system. They also proposed a quasistationary model, for whose parameters they were able to find values. The method of Vucetic [61], although not a hybrid ARQ method, is an alternative that can be used to handle time-varying channels. Drajic and Vucetic [62] evaluated several adaptive ARQ methods, giving simulation results to support their conclusion that adaptive methods perform better. In [63] and [64] Feldman and Li presented and analyzed an adaptive ARQ method, estimating the number of errors in a transmission slot and using this estimate to choose the coding parameters for the next transmission slot. Chase's code combining [57], already referred to, is an adaptive ARQ method since it uses retransmissions and combines a number of packets determined by the prevailing channel conditions.

13 A description of the generalized type-II ARQ Scheme will given shortly.

Another adaptive ARQ method is the Generalized Type-II Hybrid ARQ Scheme (GH-II ARQ) introduced [8] as a generalization of variable redundancy transmission first proposed by Mandelbaum [65]. The method uses the redundant information available upon successive retransmissions of a codeword to provide high throughput during poor channel conditions. A class of codes developed by Krishna and Morgera [7] is used to provide the adaptive feature. The codes can be made to have a nested structure in which longer codes contain smaller ones as subcodes. The longer codes necessarily have a greater minimum distance and can, therefore, correct more errors. The adaptation is effected by selecting a longer code as the channel gets poorer. In this way, the system can correct more errors. The main advantage of this class of codes is that the encoder/decoder configuration stays essentially the same as the code length is varied. This allows the receiver to maintain the same complexity even with a longer code.[14]

Regarding punctured convolutional codes, codes of a similar nature have been constructed by Du and others [64]. These have been called separable codes, implying that the redundancy symbols of the code can be isolated from the code, and within the redundancy part itself, further divisions are possible. When applied to GH-II ARQ, more redundancy is transmitted as the number of retransmissions increases. In this context, the term separable codes seems to encompass the block codes used in this thesis and the convolutional codes in [66].

This thesis takes an approach that can be used to analyze the performance of ARQ schemes when the noise in the feedback channel is taken into account. The analysis assumes that the noise is such that the number of errors in a packet forms a sequence that can be taken as a Markov chain. Since the Gilbert channel is used quite often, it is also considered, and it is shown that the number of errors in a packet does not form a Markov chain. Therefore the results derived under the Markov assumption cannot be used for it. The thesis concentrates on the GH-II ARQ scheme and the selective repeat ARQ. The reason for considering the selective ARQ scheme here is that it may be considered as a special case of the GH-II ARQ. As a preliminary step towards the GH-II ARQ, we present the ARQ schemes from which it developed.

14 A discussion of KM codes is given in Appendix A. From the discussion in this Appendix, and the description of the ARQ scheme that uses them (to be given in Chapter II), these points will be evident.

2.7 Hybrid ARQ Schemes

Hybrid ARQ schemes are those that combine pure ARQ with forward error correction (FEC). As has already been said, these methods retain the strengths of ARQ and FEC and do not have the disadvantages peculiar to either method. The hybrid ARQ schemes are broadly classified as type-I hybrid ARQ and type-II ARQ systems [6].

2.7.1 Type-I Hybrid ARQ Schemes

Type-I hybrid ARQ schemes employ one code for forward error correction (FEC) and another for error detection as shown in Figure 2.4. At the receiver, the decoder first attempts to correct any errors it finds. It then tries to detect if any errors have been left in the received block. If uncorrectable errors are detected, the received block is rejected and a request is sent to the transmitter for a retransmission. This request is usually called a negative acknowledgment (NACK). This procedure is repeated until no post-decoding errors are detected, at which time the receiver sends an acknowledgement (ACK) to the transmitter, to ask for the next block to be transmitted. Since hybrid ARQ uses codes for error detection and error correction, it requires more parity bits than pure ARQ, which uses codes for error detection only. For good channels, therefore, the throughput efficiency of a hybrid ARQ may be lower than that of the corresponding pure ARQ scheme [6]. When the channel error rate is high, a type-I hybrid ARQ method provides a higher throughput, because its error-correcting code corrects the most frequently occurring error patterns, thereby reducing the number of transmissions. Due to this, type-I hybrid ARQ schemes are suited for use on channels whose characteristics are known *a priori* to be fairly constant.

When the channel characteristics change, the type-I hybrid ARQ schemes may not be as efficient. When the channel is "quiet", that is, when the error rate is practically zero, no error correction is required. The parity bits for error correction are not needed at this time, and their presence results in a lower throughput. When the channel is noisy, that is, when errors occur frequently, the error-correcting code may be inadequate. As a result, the number of retransmissions will be high, resulting in a lower throughput. It is difficult to design error correcting codes for channels with varying characteristics. An example of such a channel is the satellite channel, which is very quiet in good weather and behaves rather poorly during rain [67]. The variable characteristics of communication channels lead one to ask whether it would be possible to design a system that adapts to the channel fluctuations so as to maximize the throughput. Such a method would use an algorithm that implements the optimality theorem discussed earlier. The type-II hybrid ARQ scheme is one of the best such methods presently available.

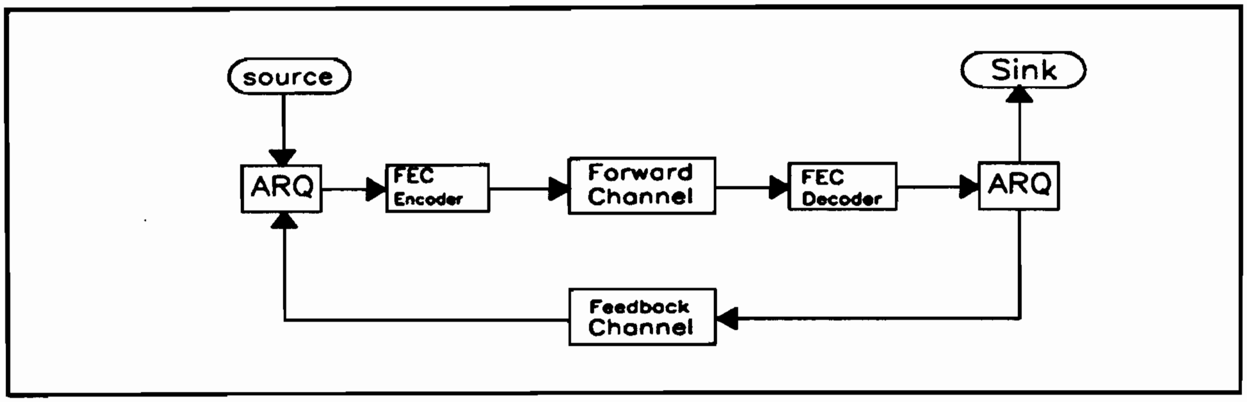

Figure 2.4 Type-I Hybrid ARQ Scheme

2.7.2 Type-II Hybrid ARQ Systems

As mentioned above, when the channel has extremely variable characteristics, an adaptive hybrid ARQ scheme is desirable. Error detection with retransmission is adaptive. Retransmission of redundancy is increased when errors occur. The concept of variable redundancy transmission was introduced by Mandelbaum [65], and based on this, the first type-II hybrid ARQ scheme was proposed by Metzner [68]. In the type-II ARQ scheme, the parity bits are transmitted only when needed. Two linear codes C_{ED} and C_{EC} are used for error detection and error correction, respectively. C_{EC} is a half-rate invertible code used for error correction. A rate-1/2 code is said to be invertible if, knowing the parity bits of a codeword, the associated information bits can be uniquely determined by an inversion process.

A message D is encoded by C_{ED} to give a block of length n. Denote this block by I. A corresponding parity block P(I), also of length n, is formed from I. At first, the transmitter sends I to the receiver. If the receiver detects any errors, it sends a NACK to the transmitter, which then sends the parity block P(I). The receiver combines this block with the previous reception (called $\hat{I}$) for decoding with the error correcting code C_{EC}. The result of this decoding is then checked for errors using C_{ED}. If errors are still found, the receiver again sends a NACK to transmitter. The transmitter, on receiving the NACK sends the original block I. This alternate transmission of I and P(I) continues until the receiver detects no errors. The receiver keeps the latest value of I in a buffer, updating as new values arrive. The same is done for the parity block P(I).

One method that uses the retransmissions of type-II ARQ scheme with diversity combining and majority-logic decoding is that of Wicker [69]. In this method, the first transmission is the normally encoded packet (information block plus a parity block). The second transmission consists of two copies of the information block, and the third transmission, two copies of the parity block. Subsequent even numbered transmissions will be two copies of the information block, while odd ones will be two copies of the parity block. In this way, there will be an odd number copies of the information portion of the codeword, and an odd number of copies of the parity portion of the codeword. So in a voting procedure, used in majority-logic decoding, there will be no ties. This method uses diversity combining, which differs from code combining, in that it combines a fixed number of transmissions each time. The author admits that diversity combining is not as efficient as code combining; it is, however, much simpler to implement [69].

Kallel and Haccoun [70] discuss the construction of rate-compatible convolutional codes. They present an algorithm that is very simple to implement. They report that, despite its simplicity, the algorithm produces good codes. After this, they proceed to use these codes in a generalized type-II hybrid ARQ scheme (GH-II ARQ) that combines the benefits of code combining with those of the ARQ scheme of Hagenauer [71]. Hagenauer, of course, has made contributions to this field in another paper [72] where the treatment of hybrid ARQ systems is given for fading channels. The work is restricted to the use of rate-compatible convolutional codes.

The GH-II ARQ scheme of Kallel and Haccoun [73] employs a set of convolutional codes designed from a common structure. As requests are made for retransmission, the method sends more sections of the convolutional code. The end result is that as more and more sections of this code are transmitted, it is soon possible to correct the errors in the received packet.

There is similarity between their approach and the one taken this thesis. The methods share the use of increasing redundancy transmission (using a code of lower rate as the channel gets poorer). The differences are that where they use convolutional codes, this thesis uses block codes, and where they use a noiseless feedback channel, this thesis pays great attention to a noisy feedback channel.

Shiozaki and others [67] present a method of error control that uses codes of varying error correcting power. Their transmission uses the HDLC-REJ protocol, which has a frame check sequence (FCS) that is used to determine whether or not a frame is error-free. When the FCS indicates the presence of errors, the receiver sends a NACK to the transmitter. The transmitter then uses a more powerful code to encode the information sequence. The decision as to which code to use is based on a bit error rate estimate. The identity of the code has to be transmitted to the receiver.

There are two differences between this method and that of this thesis: where these authors use a different code for each transmission, the method of the thesis uses variable redundancy, and where they use a noiseless feedback channel, the thesis uses a noisy feedback channel. In the next chapter, we present the generalized type-II ARQ scheme using a special type of block codes.

CHAPTER III
THE GENERALIZED TYPE-II ARQ SCHEME

In the type-II hybrid ARQ scheme, the first transmission of the parity provides the receiver with n redundant digits. Consider now the situation in which the first retransmission is the first parity block $P_1(I)$ and the second retransmission is another parity block $P_2(I)$. These two parity blocks can be used with the information block I to form a codeword for a (3n,n) error correcting code. As more parity equations can be formed, this code can correct more errors. In principle, such a scheme can be generalized to any number of retransmissions before the transmitter resends the information block. A scheme so generalized is called the Generalized Type-II Hybrid ARQ Scheme (GH-II ARQ). Of the two codes used, the code C_{EC}, which is used adaptively for error correction, is designed in such a way that its generator matrix can be partitioned into m sub-blocks. The integer m is called the depth of the code used [7]. The code rate is $1/m$. the generator matrix **G** of the code can be partitioned into sub-blocks G_i, to give

$$\mathbf{G} = [\mathbf{G}_1 | \mathbf{G}_2 | \mathbf{G}_3 | \ldots | \mathbf{G}_m]. \tag{3.1}$$

For the code C_{EC} to be useful for adaptive error correction, it is designed so that the sub-code C_i, with the generator matrix $\mathbf{G}^{(i)}$ given by

$$\mathbf{G}^{(i)} = [\mathbf{G}_1 | \mathbf{G}_2 | \mathbf{G}_3 | \ldots | \mathbf{G}_i], \qquad i \leq m, \tag{3.2}$$

has minimum distance d_i such that $d_i < d_j$ for $1 \leq i \leq j \leq m$. The code C_m is the final code C_{EC} itself. In the GH-II ARQ scheme, the information codeword produced from the original information block can be represented as

$$\mathbf{c} = [\mathbf{c}_1 | \mathbf{c}_2 | \mathbf{c}_3 | \ldots | \mathbf{c}_m], \tag{3.3}$$

where each $\mathbf{c}_i$ corresponds to $\mathbf{G}_i$, i=1,2,...,m. The data block I can be recovered from $\mathbf{c}_i$ if and only if $\mathbf{G}_i$ is invertible. The sequence of transmitted blocks until successful decoding is $\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3, \ldots, \mathbf{c}_m$. When $\mathbf{c}_i$ is received, the receiver inverts it and determines if there are any errors in the result. If errors are found, $\mathbf{c}_i$ is combined with previously received blocks for decoding followed by error detection. If errors are found, a NACK is sent to the transmitter to request the block $\mathbf{c}_{i+1}$, and the procedure continues. If no errors are found, an ACK is sent to the transmitter and the next packet is transmitted. With the sequence of blocks above, the receiver performs error correction based on the codes $C_1, C_2, \ldots, C_m$, having minimum distances $d_1, d_2, \ldots, d_m$. With each retransmission, therefore, a code of a larger minimum distance is used for error correction until the code C_m is reached. It is clear from this description that the type-II ARQ scheme is a special case of the GH-II ARQ scheme, obtained by setting m=2. We consider the use of KM codes in the GH-II ARQ scheme. The generator matrix of a KM code is block structured as $\mathbf{M} = [\mathbf{M}_1 | \mathbf{M}_2 | \ldots | \mathbf{M}_m]$. The matrix $\mathbf{G}_i$ is related to $\mathbf{M}_i$ as

$$\mathbf{G}_i = \begin{pmatrix} \mathbf{M}_i & \mathbf{0} & \mathbf{0} & \cdot & \cdot & \cdot & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_i & \mathbf{0} & \cdot & \cdot & \cdot & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{M}_i & \mathbf{0} & \cdot & \cdot & \mathbf{0} & \mathbf{0} \\ \cdot & \cdot & \mathbf{0} & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdot & \cdot & \cdot & \mathbf{0} & \mathbf{M}_i \end{pmatrix} = \mathrm{diag}(\mathbf{M}_i, \mathbf{M}_i, \ldots, \mathbf{M}_i) \qquad (3.4a)$$

$$\mathbf{G}_i^{-1} = \mathrm{diag}(\mathbf{M}_i^{-1}, \mathbf{M}_i^{-1}, \ldots, \mathbf{M}_i^{-1}). \qquad (3.4b)$$

The generator matrix $\mathbf{G}$ is then formed as $\mathbf{G} = [\mathbf{G}_1 | \mathbf{G}_2 | \ldots | \mathbf{G}_m]$. The code with generator matrix $[\mathbf{G}_1 | \mathbf{G}_2 | \ldots | \mathbf{G}_i]$ is a depth-i code. For example, the (12,4,5) KM code has the generator matrix

$$\mathbf{M} = \begin{pmatrix} 1011 | 0111 | 1000 \\ 0110 | 1101 | 1000 \\ 0001 | 0110 | 1011 \\ 0000 | 1111 | 0101 \end{pmatrix} = [\mathbf{M}_1 | \mathbf{M}_2 | \mathbf{M}_3] \qquad (3.5)$$

and is a depth-3 code. In the generator matrix, the first two blocks form the generator matrix for the (8,4,3) KM code. It is evident that the matrices $\mathbf{M}_1$, $\mathbf{M}_2$ and $\mathbf{M}_3$ are invertible. This means that for an information vector $\mathbf{d}$, if $\mathbf{B}^{(i)} = \mathbf{d}\mathbf{G}_i$, then $\mathbf{d}$ can be recovered from $\mathbf{B}^{(i)}$ by the inversion process $\mathbf{d} = \mathbf{B}^{(i)}\mathbf{G}_i^{-1}$.

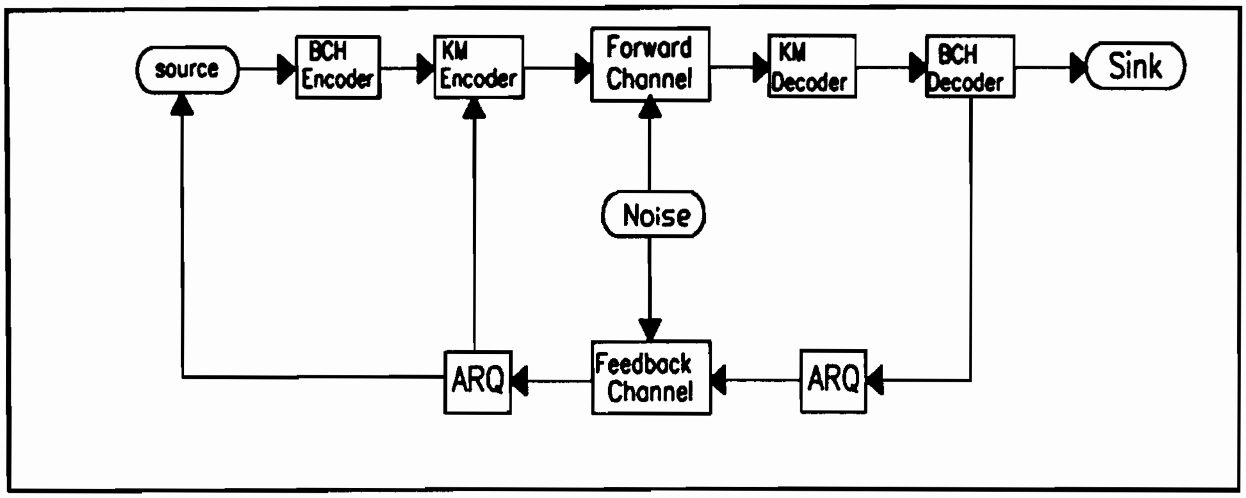

Figure 3.1 The GH-II ARQ Scheme

Code Concatenation

A block diagram of the GH-II ARQ scheme is shown in Figure 3.1, from which it is evident that the method uses code concatenation, a topic that has received considerable attention. Code concatenation was first proposed by Forney [74] as a practical technique for implementing a code with a very long length and a higher error correcting power. This is further discussed by Mokrani and Solimani [75], who point out that as the error rate in a channel increases, a longer, more powerful code is needed. The longer code, however, may have a greater encoder/decoder complexity, creating doubt as to whether there would be any overall gain. They suggest (in agreement with Forney) that code concatenation can provide the required error correcting power with a lower complexity. This fact is also stated by Clark and Cain [76].

The CCSDS[1] has recommended a concatenated coding scheme as a standard for telemetry channel coding [77]. This coding system uses, as outer code, the (255,223) Reed-Solomon code and a rate-1/2 convolutional code as an inner code. These concatenated coding systems have even been used by ESA[2] in the Halley's Comet mission [77].

1 The Consultative Committee for Space Data Systems (Commité Consultative des Systémes de Donnés Spatiales), a constituent of the CCITT.

2 The European Space Agency (ESA) is an organization that is committed to developing more efficient space communications systems, and also space experimentation stations. It has as its technical branch the European Space and Technology Centre (ESTEC).

Kasami, Fujiwara, and Lin [78] have given an excellent analysis of a concatenated coding scheme for error control using block codes. They obtained expressions for the probability of successful decoding and showed how to find the complexity for the computation of this probability. Kløve and Miller [79] showed that there is a condition that the minimum distances of the two codes must satisfy for concatenation to improve reliability. This condition (hereafter called the Kløve-Miller condition) is stated as follows:

The Kløve-Miller Condition: Given a code **C**, with codewords **c**, each of length n. Denote the distance between any two codewords **c** and **y** by d(**y**,**c**). Further, define the following:

$$m(\mathbf{y}) = \min\{d(\mathbf{y},\mathbf{c}) \mid \mathbf{c} \in \mathbf{C}\} \text{ for all } \mathbf{y} \in \{0,1\}^n \quad (3.6a)$$

$$d_C = \min\{m(\mathbf{y}) \mid \mathbf{y} \in \{0,1\}^n\}, \quad (3.6b)$$

where m(**y**) is the distance to the nearest codeword from **y**, and d_C is called the covering radius of the code. If two codes **C** and **D** are concatenated so that **D** is the outer code and **C** is the inner code, then for there to be an improvement due to concatenation, it is necessary and sufficient that

$$d_C \; > \; 2d_D. \quad (3.7)$$

This condition is stated here without proof, as its proof is found in [79]. Therefore, good codes cannot be constructed by arbitrary concatenation. We show below that the minimum distances used in the thesis do satisfy the Kløve-Miller condition.

For the BCH code used here $d_D \geq 5$, since the code was obtained by shortening a distance-5 code. Experimentally, the minimum distance was found to be 13. For the derived variable depth code, $d_C > 3n_1$, where n_1 is the number of sub-blocks into which the BCH codeword has been divided, and 3 is the minimum distance of the depth-2 (14,7,3) KM code. Since n_1=72 (giving $d_C > 216$), we see that these numbers satisfy the Kløve-Miller condition.

Deng and Costello [80], in their discussion of code concatenation, also discussed the channel models that are appropriate for the errors handled by the inner and outer codes. They found that concatenation can reduce the number of undetected errors and, at the same time, provide a high throughput. Code concatenation can yield a much more powerful coding scheme using relatively simple codes. It must be mentioned here that the overall complexity is not reduced by using concatenation. However, when compared to other coding schemes of equal power, it is found that the complexity is much lower when concatenation is used.

The GH-II ARQ Scheme

The matrices G_i and G_i^{-1} may suggest that we will be handling rather large matrices, but their implementation is considerably simplified when we consider the partitioning of the data into blocks that are convenient for encoding by a KM code. In the GH-II ARQ scheme, the information vector is first encoded by the error detecting code (which is a shortened BCH code here). The output of this encoder is the input to the KM encoder. To effect this encoding, the BCH encoder output is first partitioned into sub-blocks of n_l bits each. Each sub-block is then encoded to produce mn_l bits. Since the KM code has a generator matrix with invertible blocks, the n_l input bits can be recovered from the blocks by an inversion process. For transmission, the first blocks are selected as shown in Figure 3.2.

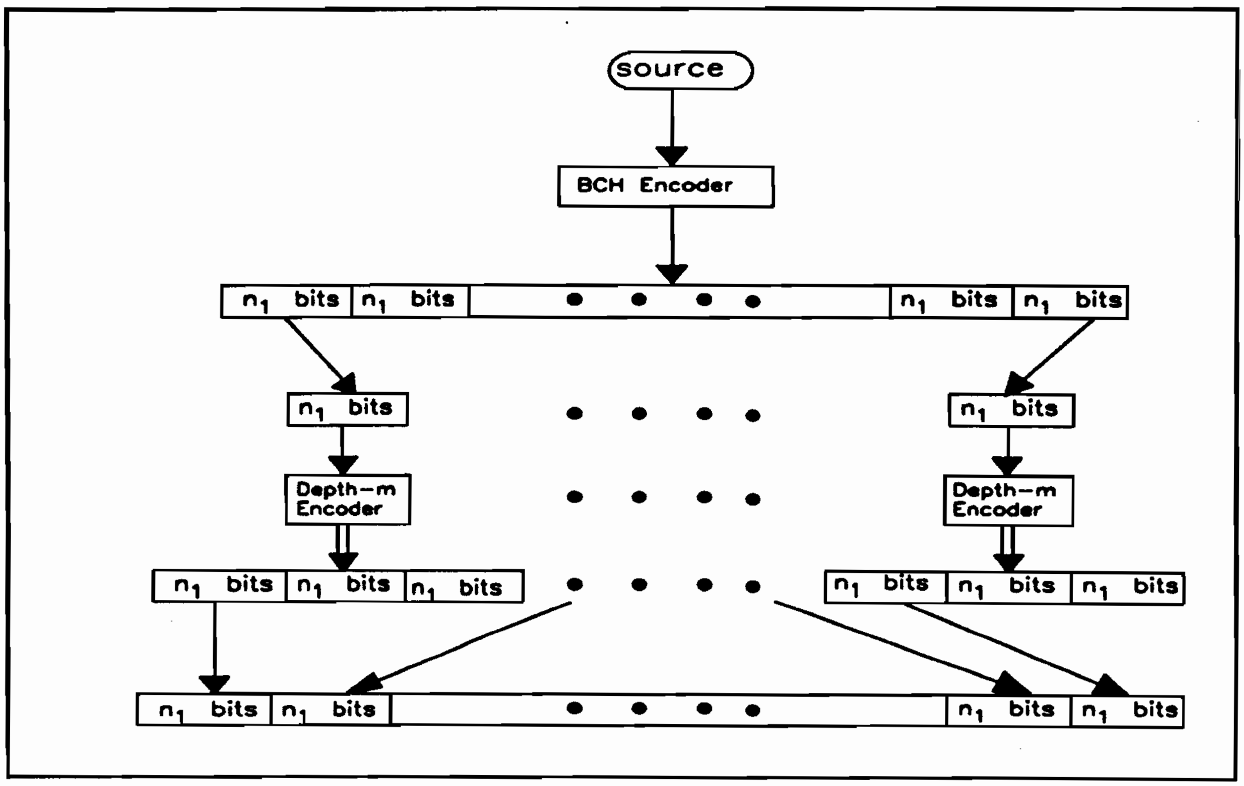

Figure 3.2 Creating the First Transmission Block

The receiver inverts the first transmission and applies BCH decoding to the result for error detection. When the receiver detects errors in this first transmission, the transmitter is asked (by sending a NACK signal) to send the second block. The sub-blocks for the second block are selected in a manner similar to that of the first. The receiver inverts the block, and follows this with error detection. If errors are detected, it combines these blocks as shown in Figure 3.3 and proceeds with decoding using the first two blocks of the KM code. This is referred to as depth-2 decoding.

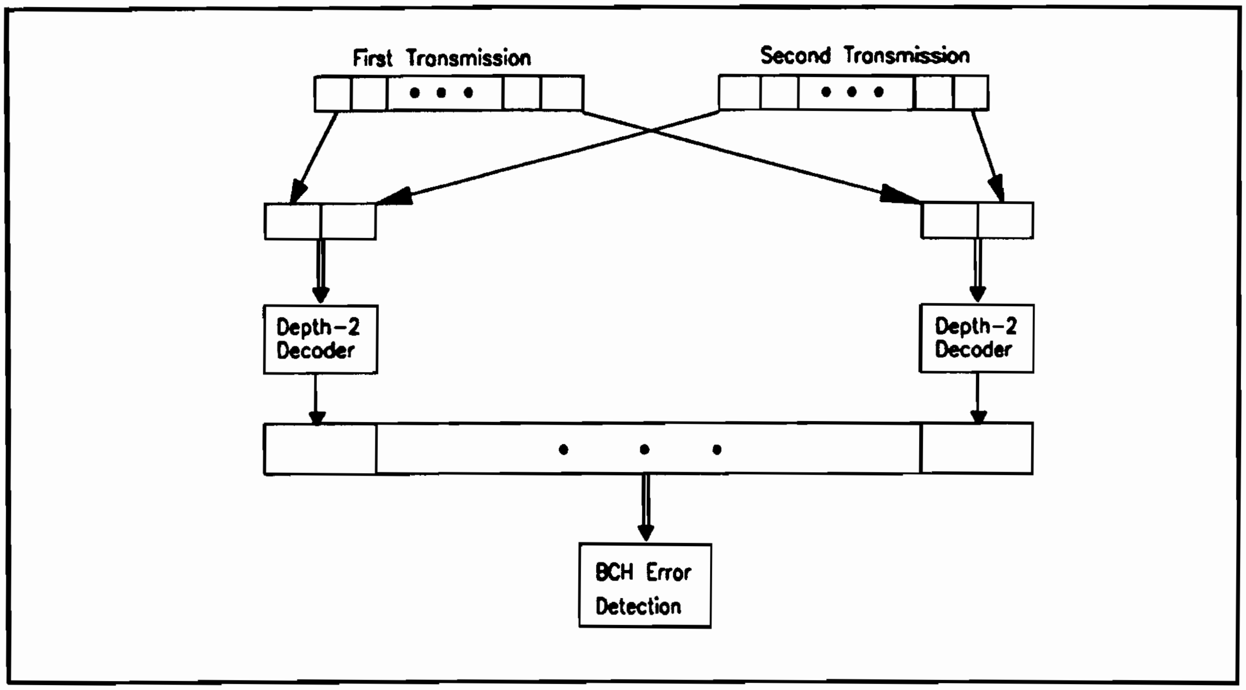

Figure 3.3 Selection of Blocks For Depth-2 Decoding

The BCH code is then used for error detection. If errors are still detected, the transmitter sends the third block. The selection of which is the similar to that of the first. This block is then first inverted. It is then checked for errors. If errors are found, this block is combined with the previous two for decoding with the first three blocks of the KM decoder. This is called depth-3 decoding. After this decoding, error detection follows. The procedure is as shown in Figure 3.4. If errors are still found, the transmitter is requested to transmit the fourth block, and the procedure repeated. In an infinite order system, the blocks transmitted will keep on coming from later sections of the KM encoder. Practically, it is found that there is little to be gained by going beyond depth-4. In fact, there is very little advantage in going from depth-3 to depth-4 [53].

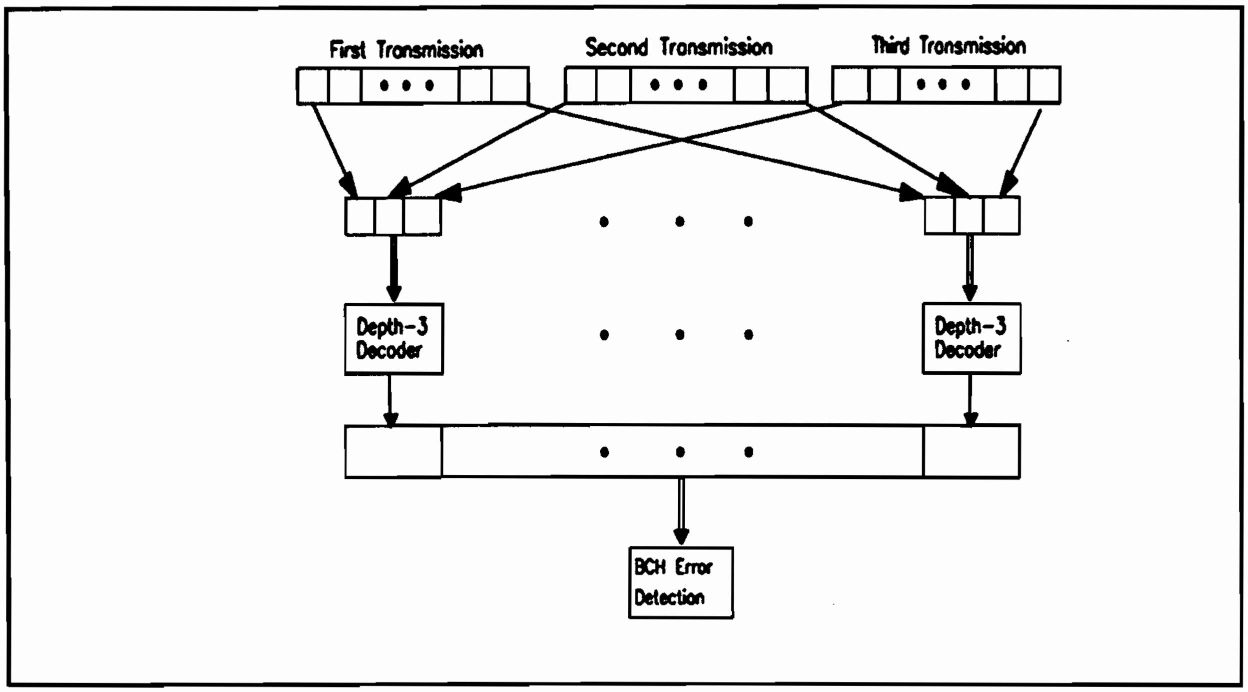

Figure 3.4 Selection of Blocks For Depth-3 Decoding

The flow chart of Figure 3.5 shows the overall procedure. The decoder divides the packets into sub-blocks of length n_l bits, and decodes each one. The entire packet is considered polluted if at least one sub-block is found to contain uncorrectable errors. Similarly, if the errors are undetectable in one of them, the whole packet is considered to contain undetectable errors. Therefore, there are three states for the decoder. The manner in which the decoder undergoes transitions among these states during decoding is given in Appendix B.

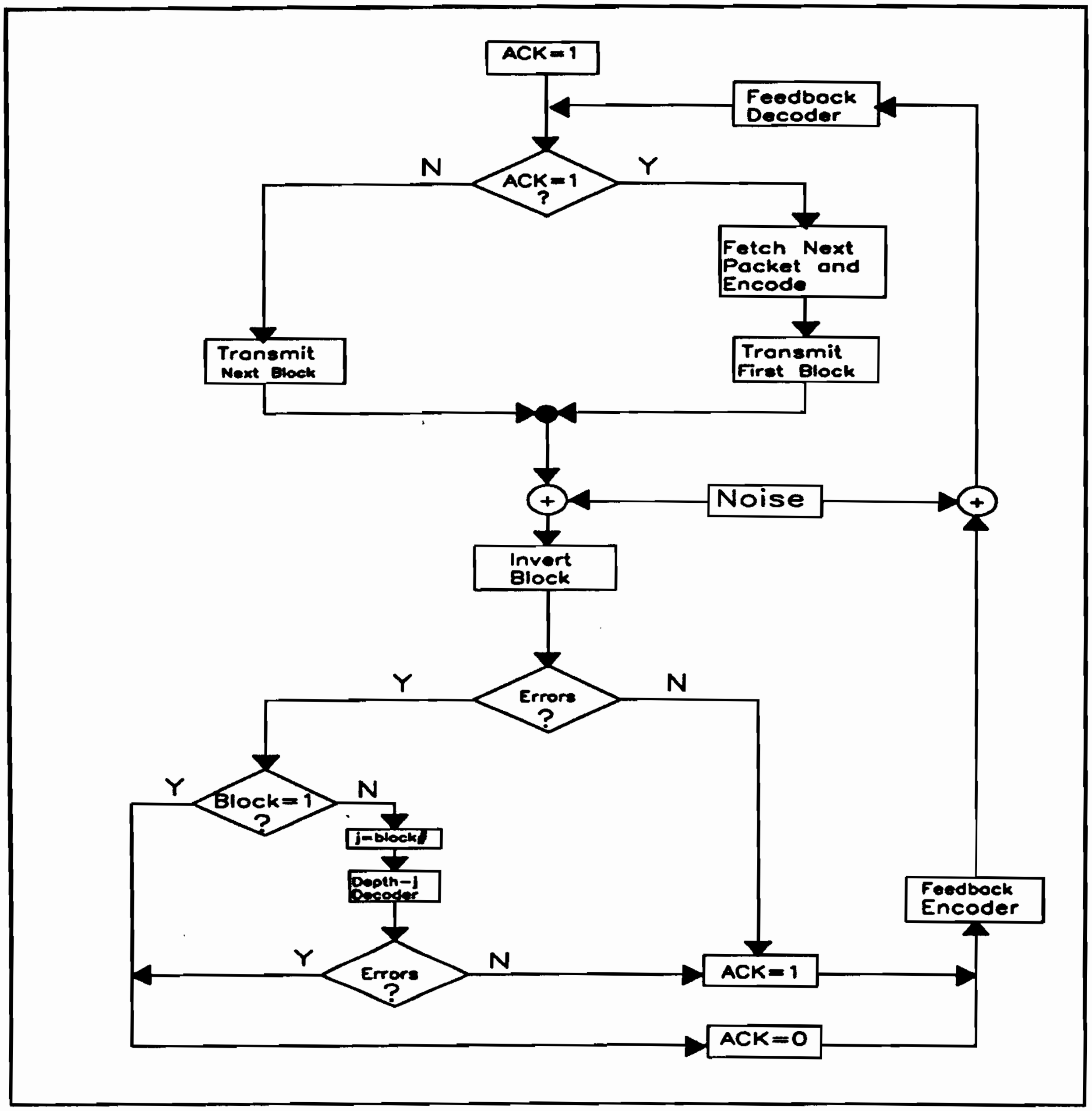

Figure 3.5 The GH-II ARQ Operation

3.1 Performance Evaluation

In this section, it is assumed that the error process in the channel can be taken as a Markov chain. Specifically, let $e_k(N)$ be the number of errors in a block of length N bits at time k. the sequence $e_1(N)$, $e_2(N)$, $e_3(N)$,..., is taken as a Markov chain. The way in which this is an adequate representation of several real channels is discussed in [21,79]. With this assumption, we can derive the expressions for the probabilities of correct delivery, undetected error, and packet loss. In addition, we can also derive the expression for the throughput efficiency. These calculations are the subject of the pages that follow.

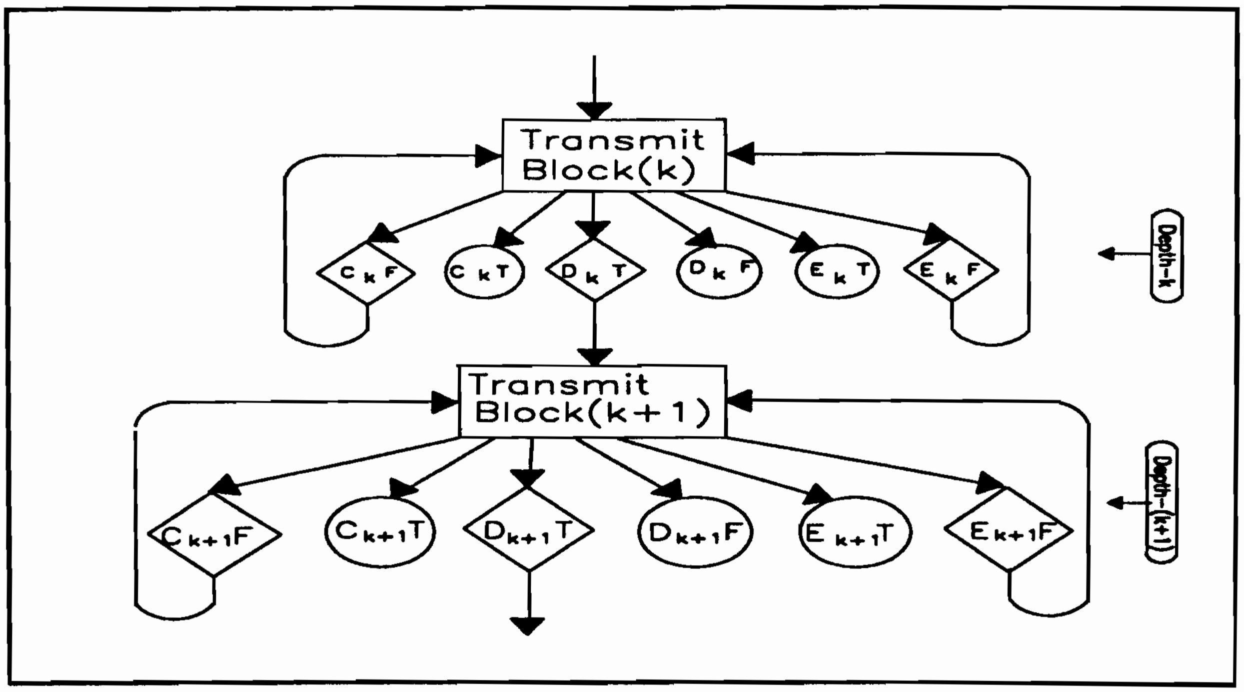

Figure 3.6 State Transition from One Transmission to the Next.

Let C_k, D_k, and E_k, be the events that the errors at depth-k are, respectively, *correctable, detectable,* or *undetectable*. Further, let T and F be the events, respectively, that the feedback message is received as *true* or *false*. A packet transmitted may be retransmitted. The decision to retransmit is made after the corresponding feedback message has been received. The feedback message can be true (T) or false (F). The reception of a packet can result in one of three possibilities. The errors may be correctable, detectable, or undetectable. Referring to Figure 3.6, the packet may end in one of the circled states, C_4T, C_3T, etc, which are the absorbing states. These states C_kT are actually obtained by concatenating states, so that C_kT means C_k followed by T. Figure 3.6 shows how the depth changes from one event to the next, until the final depth (depth-4) is reached. The state transitions at the final depth are illustrated in Figure 3.7, where the depth subscripts are omitted for simplicity.

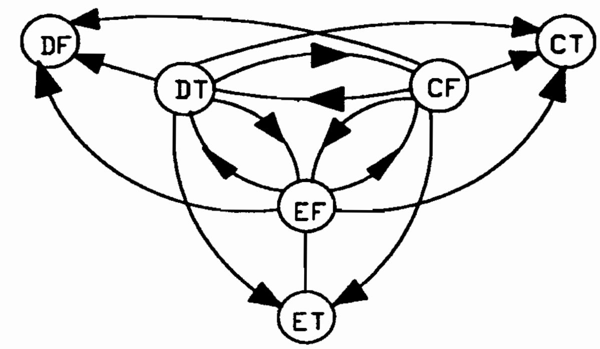

Figure 3.7 Transitions After Reaching the Maximum Depth

The absorbing states are now CT, DF, and ET, as these indicate the termination of the transmission of a packet. When the transmission of a packet ends, it can be viewed as having been absorbed into one of these states. One way to evaluate the performance of the ARQ scheme is, therefore, to first obtain the first entry probabilities, and then use them to obtain the other required expressions. If we represent the transition probabilities as p_{ij}, and the probability of going from state i to state j in n steps, as $p_{ij}^{(n)}$ then

$$p_{ij}^{(n)} = \sum_{k=0}^{n} f_{ij}^{(k)} p_{ii}^{(n-k)}, \qquad (f_{ii}^{(0)} = 1, \; f_{ij}^{(0)} = 0 \quad \text{for} \quad i \neq j), \tag{3.8}$$

where $f_{ij}^{(k)}$ is the probability that starting in state i, the first entry into state j occurs at the kth step. Defining the moment generating functions $P_{ij}(z)$, *and* $F_{ij}(z)$ as,

$$P_{ij}(z) \triangleq \sum_{k=0}^{\infty} z^k p_{ij}^{(k)} \quad \text{and} \tag{3.9}$$

$$F_{ij}(z) \triangleq \sum_{k=0}^{\infty} z^k f_{ij}^{(k)}, \tag{3.10}$$

it can be shown that

$$F_{ij}(z) = \frac{P_{ij}(z)}{P_{jj}(z)} \quad \text{and} \tag{3.11}$$

$$F_{ii}(z) = 1 - \frac{1}{P_{ii}(z)}. \tag{3.12}$$

The first entry probabilities can be found by inverting the above moment generating functions. Since the determination of the performance parameters necessitates the determination of $P_{ij}(z)$, a process which requires a lot of computation, this work does not use this approach, and instead finds the $f_{ij}^{(k)}$'s directly.

3.1.1 Correct Delivery at the *k*th Transmission

The probability of correct delivery at the *k*th transmission is given by the following lemma.

Lemma 3.1: Let $P_c(k)$ be the probability that a packet is correctly delivered for the first time at its *k*th transmission. Then

$$P_c(k) = \begin{cases} P(C_1)P(T/C_1) & k = 1 \\ P(D_1)P(T/D_1)\beta_2 & k = 2 \\ P(D_1)P(T/D_1)\beta_3\alpha_2 & k = 3 \\ P(D_1)P(T/D_1)\beta_4\alpha_3\alpha_4^{k-4} & k \geq 4, \end{cases} \tag{3.14}$$

where
$$\alpha_2 = P(D_2/T)P(T/D_2)$$
$$\alpha_3 = P(D_2/T)P(T/D_2)P(D_3/T)P(T/D_3)$$
$$\alpha_4 = P(D_4/T)P(T/D_4), \quad \text{and}$$
$$\beta_k = P(C_k/T)P(T/C_k).$$

Proof: With the help of Figure 3.8, we can see that the probability $P_c(1)$ of correct delivery on the first transmission is given by $P_c(1) = P(C_1) \cdot P(T/C_1)$. The possible sequences of events leading to the departure of a packet at the second transmission are as illustrated in Figure 3.9

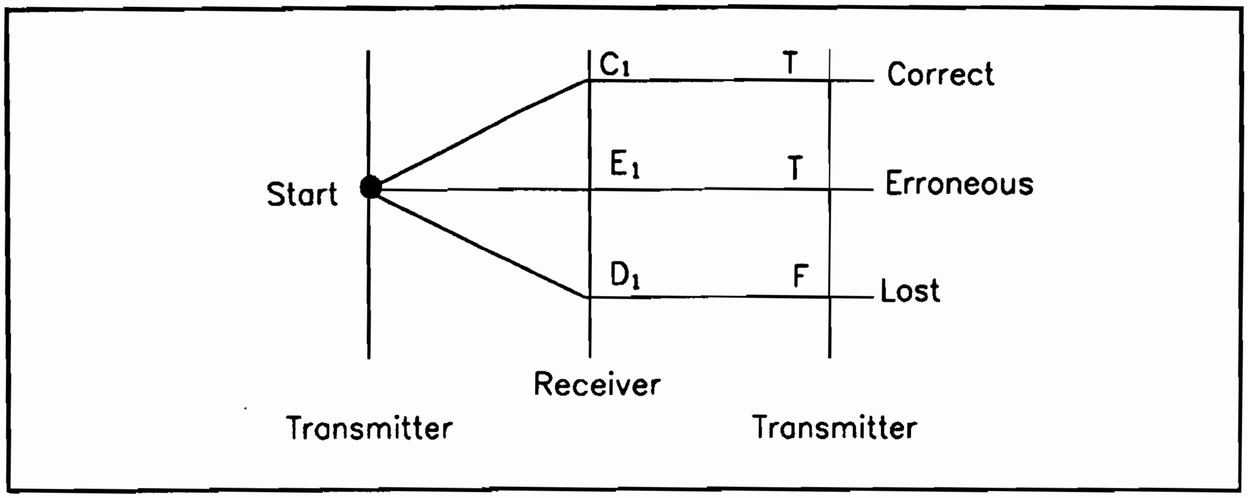

Figure 3.8 Departure of a Packet on the First Transmission

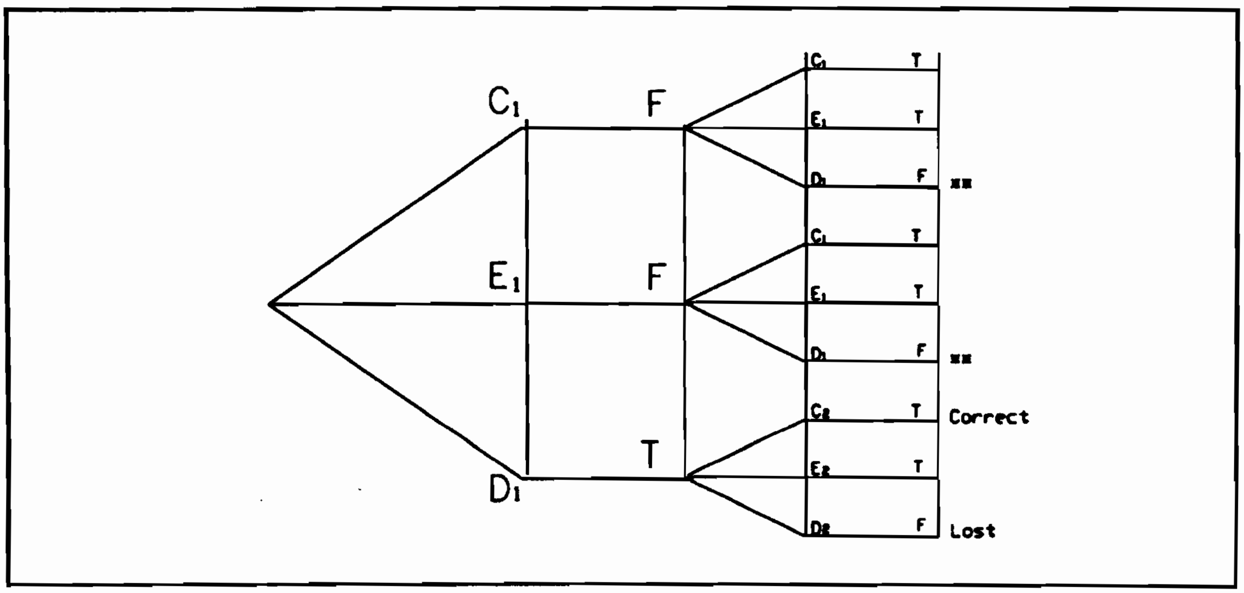

Figure 3.9 Departure of a Packet on the Second Transmission[3]

3 In Figure 3.9, and in others to come, the points labelled ** denote the loss of a packet that has already been delivered. This is not a real loss, since there has already been a delivery. It is really a spurious transmission that has turned into a "loss."

using the Markovian assumption, we can find the probability $P_c(2)$ of correct delivery at the second transmission as

$$P_c(2) = P(D_1)P(T/D_1)P(C_2/T)P(T/C_2). \tag{3.15}$$

We proceed to find the corresponding expressions for the third transmission. We use Figure 3.10, where we use the fact that $\overline{D}_k = C_k \cup E_k$ in order to simplify the diagram. The probability of correct delivery at the third transmission is given by

$$P_c(3) = P(D_1)P(T/D_1)P(D_2/T)P(T/D_2)P(C_3/T)P(T/C_3). \tag{3.16}$$

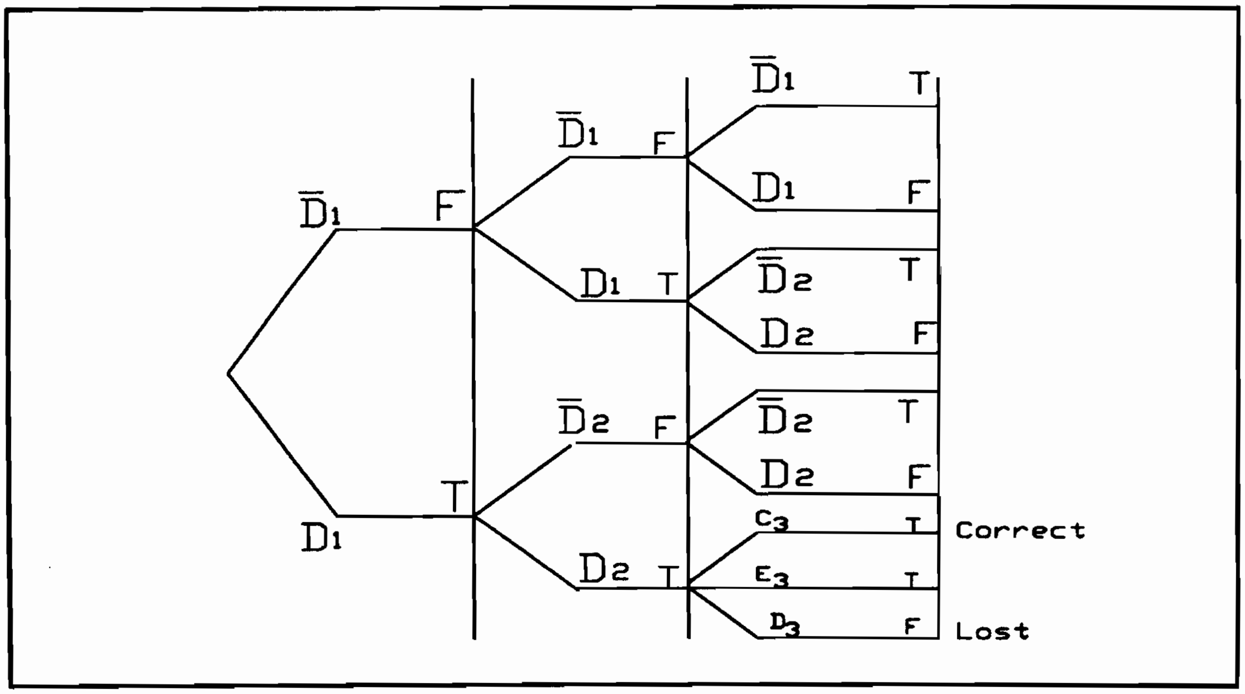

Figure 3.10 Departure of a Packet at the Third Transmission

We notice that in Figures 3.8, 3.9, and 3.10, the depth changes in some cases, while in others it does not change. For example, when $\overline{D}_1$ is followed by a false feedback (F), the transmission of a new packet begins. The transmission of a packet begins at depth-1; therefore, there is no depth change when $\overline{D}_1$ is followed by a false feedback (F). If D_k is followed by a true feedback (T), however, the depth increases, since the requested transmission is used in a higher depth (D_{k+1}) upon its arrival. This action of depth change is summarized in the illustration of Figure 3.11.

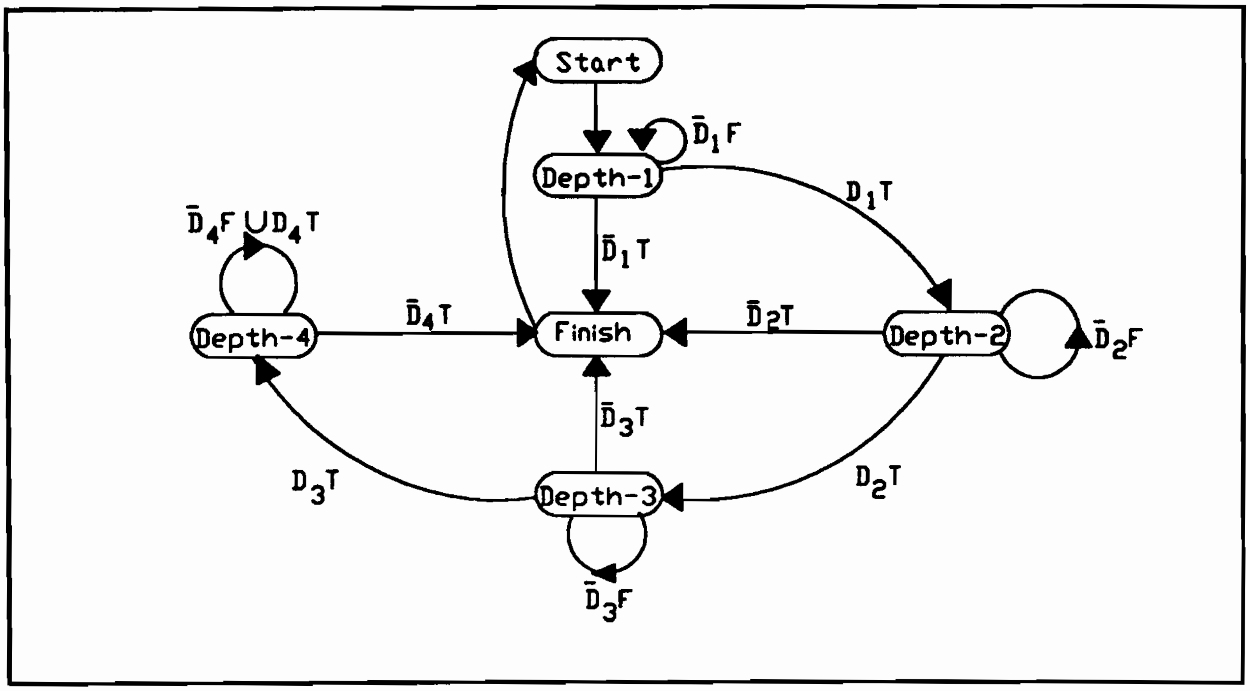

Figure 3.11 Change of Depth For the GH-II ARQ scheme

The labels on the arrows in Figure 3.11 indicate the events causing the changes. From this figure, it can be seen that extension to higher depth systems is quite possible. This can be done by appropriately modifying the transitions from depth-4. Since the procedure is assumed clear at this point, several steps are omitted here, and only the results are given. We have

$$P_c(4) \;=\; P(D_1)P(T/D_1)P(D_2/T)P(T/D_2)P(D_3/T)P(T/D_3)P(C_4/T)P(T/C_4). \qquad (3.17)$$

At this point it is fitting to introduce some variables that will simplify the expressions. We define

$$\alpha_2 \;=\; P(D_2/T)P(T/D_2) \qquad (3.18a)$$

$$\alpha_3 \;=\; P(D_2/T)P(T/D_2)P(D_3/T)P(T/D_3) \qquad (3.18b)$$

$$\alpha_4 \;=\; P(D_4/T)P(T/D_4), \quad \text{and} \qquad (3.18c)$$

$$\beta_k \;=\; P(C_k/T)P(T/C_k), \qquad (3.18d)$$

and proceeding as above, we obtain the general expression as

$$P_c(k) = \begin{cases} P(C_1)P(T/C_1) & k=1 \\ P(D_1)P(T/D_1)\beta_2 & k=2 \\ P(D_1)P(T/D_1)\beta_3\alpha_2 & k=3 \\ P(D_1)P(T/D_1)\beta_4\alpha_3\alpha_4^{k-4} & k\geq 4, \end{cases} \tag{3.19}$$

which completes the proof (a rather long proof).

3.1.2 Incorrect Delivery at the *k*th Transmission

The probability of incorrect delivery at the *k*th transmission of a packet is given by the lemma that follows.

Lemma 3.2: Given $\beta = P(F/F), \delta = P(D_1)P(T/D_1)P(F/T)$, and $\sigma = P(D_1)P(T/D_1)P(E_4/T)$, we denote by $P_e(k)$ the probability that a packet is erroneously delivered at its *k*th transmission. Then

$$P_e(1) = P(E_1), \tag{3.20a}$$

$$P_e(2) = P_f + P(D_1)P(T/D_1)P(E_2/T), \tag{3.20b}$$

$$P_e(3) = P_fP(F/F) + P(D_1)P(T/D_1)P(F/T) + P(D_1)P(T/D_1)P(D_2/T)P(T/D_2)P(E_3/T), \tag{3.20c}$$

$$P_e(4) = P_fP(F/F)^2 + P(D_1)P(T/D_1)P(F/T)P(F/F) + P(D_1)P(T/D_1)P(D_2/T)P(T/D_2)P(F/T), + P(D_1)P(T/D_1)P(E_4/T)\alpha_3, \tag{3.20d}$$

$$P_e(k) = (P_f\beta^2 + \delta\beta + \delta\alpha_2)\beta^{k-4} + \sigma\alpha_3\alpha_4^{k-4} + \delta\alpha_3\beta^{k-5}\sum_{j=0}^{k-5}\left(\frac{\alpha_4}{\beta}\right)^j, \quad k\geq 5. \tag{3.20e}$$

Proof: The proof proceeds by reading the event sequences from the departure diagrams given above. Let $P_f = P(\overline{D}_1)P(F/\overline{D}_1)$ be the probability that a delivery on the first transmission is followed by a false feedback message. From the events leading to delivery with undetected errors, we have

$$P_e(1) = P(E_1), \tag{3.21a}$$

$$P_e(2) = P_f + P(D_1)P(T/D_1)P(E_2/T), \tag{3.21b}$$

$$\begin{aligned} P_e(3) &= P_fP(F/F) + P(D_1)P(T/D_1)P(F/T) \\ &+ P(D_1)P(T/D_1)P(D_2/T)P(T/D_2)P(E_3/T), \end{aligned} \tag{3.21c}$$

$$\begin{aligned} P_e(4) &= P_fP(F/F)^2 + P(D_1)P(T/D_1)P(F/T)P(F/F) + P(D_1)P(T/D_1)P(E_4/T)\alpha_3 \\ &+ P(D_1)P(T/D_1)P(D_2/T)P(T/D_2)P(F/T). \end{aligned} \tag{3.21d}$$

Define $\beta = P(F/F)$ to prevent the expressions from becoming too long. The probability of incorrect delivery at the fifth transmission is then given by

$$\begin{aligned} P_e(5) &= P_f\beta^3 + P(D_1)P(T/D_1)P(F/T)\beta^2 + P(D_1)P(T/D_1)P(F/T)\alpha_2\beta \\ &+ P(D_1)P(T/D_1)P(E_4/T)\alpha_3\alpha_4 \end{aligned} \tag{3.21e}$$

$$\begin{aligned} &= P_f\beta^3 + P(D_1)P(T/D_1)P(F/T)[\beta^2+\alpha_2\beta+\alpha_3] \\ &+ P(D_1)P(T/D_1)P(E_4/T)\alpha_3\alpha_4. \end{aligned} \tag{3.21f}$$

We further introduce $\delta = P(D_1)P(T/D_1)P(F/T)$ and $\sigma = P(D_1)P(T/D_1)P(E_4/T)$. Then

$$P_e(6) = P_f\beta^4 + \delta[\beta^3+\alpha_2\beta^2+\alpha_3\beta+\alpha_3\alpha_4] + \sigma\alpha_3\alpha_4^2, \tag{3.21g}$$

$$P_e(7) = P_f\beta^5 + \delta[\beta^4+\alpha_2\beta^3+\alpha_3\beta^2+\alpha_3\alpha_4\beta+\alpha_3\alpha_4^2] + \sigma\alpha_3\alpha_4^3. \tag{3.21h}$$

The general expression for $k \geq 5$ is

$$P_e(k) = (P_f\beta^2+\delta\beta+\delta\alpha_2)\beta^{k-4}+\sigma\alpha_3\alpha_4^{k-4}+\delta\alpha_3\beta^{k-5}\sum_{j=0}^{k-5}\left(\frac{\alpha_4}{\beta}\right)^j, \quad k \geq 5. \tag{3.22}$$

This completes the derivation of the probability of incorrect delivery at the kth transmission. This expression will be used later in determining the average probability of incorrect delivery. As for now, we proceed to derive a similar expression for the loss of a packet.

3.1.3 Packet Loss at the *k*th Transmission

A condition necessary for the loss of a packet is the detection of errors in a packet. This, by itself, is not sufficient to cause the loss of a packet, but if this is followed by a false feedback message then a packet will be lost if the subsequent feedback message is true. Thus, the loss of a packet occurs whenever the event sequence is

[Errors Detected] [False feedback message] [True feedback message]

So, in deriving the expressions for the loss of a packet, we will trace the event sequences that end with the sequence **DFT**. The loss occurs in this case for the following reason. If the feedback message following error detection is false, the transmitter will start the transmission of the next packet instead of a retransmission. The receiver will combine the two transmissions for decoding, and, since these transmissions belong to different packets, the decoding will be meaningless. The receiver will not be able to detect the problem since the equivalent number of errors will be greater than the error detecting power of the code used. A delivery will be made to the user and the transmitter informed accordingly. If this message reaches the transmitter unaltered (**true** feedback) the transmitter will begin transmitting the next packet. Two things have happened here. A packet has been delivered in error, and another has been lost. The scenario described here is illustrated in Figure 3.12.

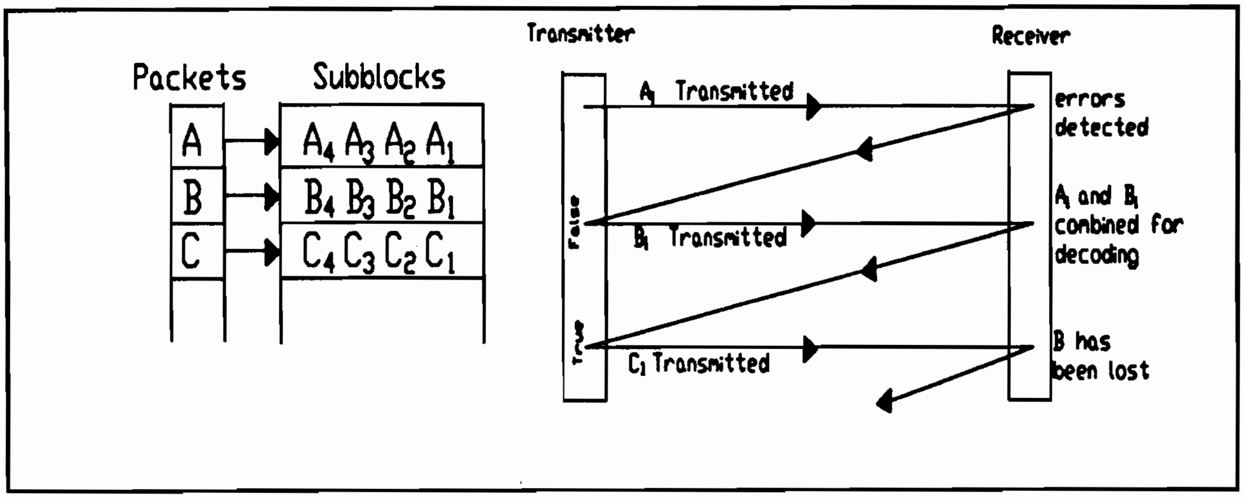

Figure 3.12 The Loss of a Packet

The probability $P_l(k)$ of packet loss at the kth transmission is given by the following lemma. The proof is omitted here because by tracing the event sequences ending with the sequence **DFT**, the results will be evident.

Lemma 3.3: The probability of packet loss after a number of transmission attempts is given by

$$P_l(1) = P(D_1)P(F/D_1)P(T/F) \tag{3.23a}$$

$$P_l(2) = P(D_1)P(F/D_2)P(T/F)P(T/D_1)P(D_2/T) \tag{3.23b}$$

$$P_l(3) = P(D_1)P(F/D_3)P(T/F)P(T/D_1)P(D_2/T)P(T/D_2)P(D_3/T) \tag{3.23c}$$

$$P_l(4) = P(D_1)P(F/D_4)P(T/F)P(T/D_1)P(D_2/T)P(T/D_2)P(D_3/T)P(T/D_3)P(D_4/T) \tag{3.23d}$$

$$P_l(k) = P_l(4)\alpha_4^{k-4}, \qquad k \geq 5 \tag{3.24}$$

We are now in a position to derive the expressions for the average probabilities of correct delivery and undetected error.

3.2 Average Probability of Undetected Error and Correct Delivery

Suppose we have P packets to transmit. We define the probabilities of undetected error and of correct delivery as the limits

$$P_{UD} = \lim_{P\to\infty}\left\{\frac{\text{number of erroneous deliveries for P packets}}{\text{total number of deliveries for P packets}}\right\}$$

$$= \lim_{P\to\infty}\left\{\frac{\overline{N}_e(P)}{\overline{N}_{\text{del}}(P)}\right\}.$$

$$P_C = \lim_{P\to\infty}\left\{\frac{\text{number of correct deliveries for P packets}}{\text{total number of deliveries for P packets}}\right\}$$

$$= \lim_{P\to\infty}\left\{\frac{\overline{N}_c(P)}{\overline{N}_{\text{del}}(P)}\right\}.$$

We consider here the number of deliveries made, rather than the number of packets, because multiple deliveries of the same packet are possible. The following definitions are needed in the theorem that is soon to follow.

$$\omega = P(D_1)P(T/D_1)\alpha_3\alpha_4, \tag{3.25}$$

$$Q_2 = P_f\beta^2 + \delta\beta + \delta\alpha_2, \tag{3.26}$$

$$S_e = \sum_{k=1}^{4} P_e(k), \quad \text{and} \quad S_c = \sum_{k=1}^{4} P_c(k). \tag{3.27}$$

Theorem 3.1: Given ω, Q_2, S_e and S_c as defined above, the probabilities of undetected error and of correct delivery are given by the equations

$$P_{UD} = \frac{S_e(1-\beta)(1-\alpha_4) + Q_2(1-\alpha_4)\beta + \delta\alpha_3\beta}{(S_e+S_c)(1-\alpha_4)(1-\beta)+\alpha_4(\omega+\sigma\alpha_3)(1-\beta)+\delta\alpha_3\beta+Q_2(1-\alpha_4)\beta}. \tag{3.28}$$

$$P_C = \frac{S_c(1-\beta)(1-\alpha_4) + \omega\alpha_4(1-\beta)}{(S_e+S_c)(1-\alpha_4)(1-\beta)+\alpha_4(\omega+\sigma\alpha_3)(1-\beta)+\delta\alpha_3\beta+Q_2(1-\alpha_4)\beta}. \tag{3.29}$$

Proof: Since $\overline{N}_e(P) = P\sum_{k=1}^{\infty} P_e(k)$, and $\overline{N}_{del}(P) = P\sum_{k=1}^{\infty} P_{del}(k)$, we have

$$P_{UD} = \left(\frac{\sum_{k=1}^{\infty} P_e(k)}{\sum_{k=1}^{\infty} P_{del}(k)}\right) = \frac{\sum_{k=1}^{\infty} P_e(k)}{\sum_{k=1}^{\infty}[P_e(k)+P_c(k)]} \tag{3.30}$$

The following sums are needed in evaluating P_{UD}. The first sum is

$$\sum_{k=1}^{\infty} P_e(k) = S_e + \sum_{k=5}^{\infty} P_e(k), \quad \text{where} \quad S_e = \sum_{k=1}^{4} P_e(k), \quad \text{and}$$

using Lemma 3.2, we have

$$\sum_{k=5}^{\infty} P_e(k) = (P_f\beta^2+\delta\beta+\delta\alpha_2)\sum_{k=5}^{\infty}\beta^{k-4} + \sigma\alpha_3\sum_{k=5}^{\infty}\alpha_4^{k-4} + \delta\alpha_3\beta\sum_{k=5}^{\infty}\beta^{k-5}\sum_{j=0}^{k-5}\left(\frac{\alpha_4}{\beta}\right)^j \tag{3.31}$$

There are two ways of evaluating the double sum $\sum_{k=5}^{\infty}\beta^{k-5}\sum_{j=0}^{k-5}\left(\frac{\alpha_4}{\beta}\right)^j$. One way is to evaluate the inner sum and then the outer one. This gives

$$\sum_{k=5}^{\infty}\beta^{k-5}\sum_{j=0}^{k-5}\left(\frac{\alpha_4}{\beta}\right)^j = \sum_{k=5}^{\infty}\beta^{k-5}\left(\frac{1-(\alpha_4/\beta)^{k-4}}{1-\alpha_4/\beta}\right). \tag{3.32a}$$

Evaluating this, we obtain

$$\sum_{k=5}^{\infty}\beta^{k-5}\left(\frac{1-(\alpha_4/\beta)^{k-4}}{1-\alpha_4/\beta}\right) = \sum_{k=5}^{\infty}\frac{\beta^{k-4}-\alpha_4^{k-4}}{\beta-\alpha_4} = \frac{1}{\beta-\alpha_4}\left(\frac{\beta}{1-\beta}-\frac{\alpha_4}{1-\alpha_4}\right) \tag{3.32b}$$

$$= \frac{1}{\beta-\alpha_4}\left(\frac{(\beta-\alpha_4\beta)-(\alpha_4-\alpha_4\beta)}{(1-\alpha_4)(1-\beta)}\right) = \frac{1}{(1-\alpha_4)(1-\beta)}. \tag{3.32c}$$

The other way is to change the indices and write the sum as

$$\sum_{k=0}^{\infty}\beta^k\sum_{j=0}^{k}\left(\frac{\alpha_4}{\beta}\right)^j = \sum_{k=0}^{\infty}\sum_{j=0}^{k}\alpha_4^j\beta^{k-j}, \tag{3.33}$$

which immediately shows that the double sum is actually a product of two sums

$$\sum_{k=0}^{\infty}\sum_{j=0}^{k}\alpha_4^j\beta^{k-j} = \sum_{k=0}^{\infty}\beta^k\sum_{j=0}^{\infty}\alpha_4^j = \left(\frac{1}{1-\beta}\right)\left(\frac{1}{1-\alpha_4}\right), \tag{3.34}$$

which is the same as above. The result is then

$$\sum_{k=5}^{\infty}P_e(k) = (P_f\beta^2+\delta\beta+\delta\alpha_2)\left(\frac{\beta}{1-\beta}\right) + \frac{\sigma\alpha_3\alpha_4}{1-\alpha_4} + \frac{\delta\alpha_3\beta}{(1-\beta)(1-\alpha_4)} \tag{3.35}$$

The other sum needed is

$$\sum_{k=1}^{\infty}P_c(k) = \sum_{k=1}^{4}P_c(k) + \sum_{k=5}^{\infty}P_c(k) \tag{3.36a}$$

$$= S_c + \sum_{k=5}^{\infty}P_c(k), \tag{3.36b}$$

where S_c represents the first sum. Using the expressions of Lemma 3.1, we have

$$\sum_{k=1}^{\infty}P_c(k) = S_c + \left(\frac{\omega\alpha_4}{1-\alpha_4}\right). \tag{3.37}$$

The probability of undetected error is then found by substituting the above quantities in (3.30), which leads to

$$P_{UD} = \frac{S_e(1-\beta)(1-\alpha_4) + Q_2(1-\alpha_4)\beta + \delta\alpha_3\beta}{(S_e+S_c)(1-\alpha_4)(1-\beta)+\alpha_4(\omega+\sigma\alpha_3)(1-\beta)+\delta\alpha_3\beta+Q_2(1-\alpha_4)\beta}. \tag{3.38}$$

where $\omega = P(D_1)P(T/D_1)\alpha_3\alpha_4$, and $Q_2 = P_f\beta^2 + \delta\beta + \delta\alpha_2$. The probability of correct delivery is similarly found from

$$P_C = \frac{\sum_{k=1}^{\infty}P_c(k)}{\sum_{k=1}^{\infty}P_{del}(k)} = \frac{\sum_{k=1}^{\infty}P_c(k)}{\sum_{k=1}^{\infty}P_e(k)+P_c(k)}. \tag{3.39}$$

Substituting from previous expressions, we obtain

$$P_C = \frac{S_c(1-\beta)(1-\alpha_4) + \omega\alpha_4(1-\beta)}{(S_e+S_c)(1-\alpha_4)(1-\beta)+\alpha_4(\omega+\sigma\alpha_3)(1-\beta)+\delta\alpha_3\beta+Q_2(1-\alpha_4)\beta}. \tag{3.40}$$

3.3 Throughput Efficiency

The preceding expressions can now be used to derive the expression for the throughput efficiency for the ARQ scheme. The result is given in the theorem that follows.

Theorem 3.2:

$$\text{Given } \omega = P(D_1)P(T/D_1)\alpha_3\alpha_4, \tag{3.41}$$

$$N_1 = \sum_{k=1}^{3} k[P_e(k)+P_c(k)+P_l(k)] + 4[P_l(4)+P_e(4)] - \frac{\omega}{1-\alpha_4}, \tag{3.42}$$

$$Q_1 = \omega + \sigma\alpha_3\alpha_4 + \alpha_4 P_l(4), \tag{3.43}$$

$$Q_2 = P_f\beta^2 + \delta\beta + \delta\alpha_2, \tag{3.44}$$

$$Q_3 = \frac{5(1-\beta)^2(1-\alpha_4)^2+(\beta+\alpha_4-2\alpha_4\beta)}{(1-\alpha_4)^2(1-\beta)^2}. \tag{3.45}$$

Then the throughput efficiency is given by the expression

$$\eta = \frac{1}{N_1 + \frac{(5-4\alpha_4)}{(1-\alpha_4)^2}Q_1 + \frac{(5-4\beta)\beta}{(1-\beta)^2}Q_2 + \frac{\delta\alpha_3}{(1-\beta)^2(1-\alpha_4)^2}Q_3} \tag{3.46}$$

Proof: Defining $P_{dep}(k) = P_c(k)+P_e(k)+P_l(k)$, the average number of transmissions per packet is

$$\overline{N} = \sum_{k=1}^{\infty} kP_{dep}(k) = \sum_{k=1}^{\infty} k[P_c(k)+P_e(k)+P_l(k)]. \tag{3.47}$$

Using $\omega = P(D_1)P(T/D_1)\alpha_3\alpha_4$, and taking the sums separately, we have

$$\sum_{k=1}^{\infty} kP_c(k) = P_c(1)+2P_c(2)+3P_c(3) + \omega\sum_{k=4}^{\infty} k\alpha_4^{k-4} \tag{3.48a}$$

$$= P_c(1)+2P_c(2)+3P_c(3) + \omega\frac{4-3\alpha_4}{(1-\alpha_4)^2} \tag{3.48b}$$

$$= P_c(1)+2P_c(2)+3P_c(3) - \frac{\omega}{1-\alpha_4} + \frac{\omega(5-4\alpha_4)}{(1-\alpha_4)^2} \tag{3.48c}$$

The second sum is

$$\sum_{k=1}^{\infty} kP_e(k) \;=\; \sum_{k=1}^{4} kP_e(k) \;+\; \sum_{k=5}^{\infty} kP_e(k). \tag{3.49}$$

Substituting from (16), we obtain

$$\sum_{k=1}^{\infty} kPe(k) = (P_f\beta^2+\delta\beta+\delta\alpha_2)\frac{\beta(5-4\beta)}{(1-\beta)^2}+\frac{\sigma\alpha_3\alpha_4(5-4\alpha_4)}{(1-\alpha_4)^2}+\delta\alpha_3\sum_{k=5}^{\infty} k\beta^{k-5}\sum_{j=0}^{k-5}\left(\frac{\alpha_4}{\beta}\right)^j \tag{3.50}$$

The double sum can be written as

$$\sum_{k=5}^{\infty} k\beta^{k-5}\sum_{j=0}^{k-5}\left(\frac{\alpha_4}{\beta}\right)^j \;=\; \sum_{k=0}^{\infty}(5+k)\beta^k\sum_{j=0}^{k}\left(\frac{\alpha_4}{\beta}\right)^j \tag{3.51a}$$

$$= \frac{5}{(1-\beta)(1-\alpha_4)} + \frac{1}{\beta-\alpha_4}\left(\frac{\beta^2}{(1-\beta)^2}-\frac{\alpha_4^2}{(1-\alpha_4)^2}\right)$$

$$+ \frac{5}{(1-\beta)(1-\alpha_4)} + \frac{\beta+\alpha_4-2\alpha_4\beta}{(1-\beta)^2(1-\alpha_4)^2}. \tag{3.51b}$$

Putting these together, we obtain

$$\sum_{k=1}^{\infty} kPe(k) \;=\; Q_2\frac{\beta(5-4\beta)}{(1-\beta)^2} + \frac{\sigma\alpha_3\alpha_4(5-4\alpha_4)}{(1-\alpha_4)^2}$$

$$+ \frac{5\delta\alpha_3}{(1-\beta)(1-\alpha_4)} + \frac{\sigma\alpha_3(\beta+\alpha_4-2\alpha_4\beta)}{(1-\beta)^2(1-\alpha_4)^2} \tag{3.52}$$

The last sum we need is

$$\sum_{k=1}^{\infty} kP_l(k) \;=\; P_l(1)+2P_l(2)+3P_l(3)+P_l(4) \;+\; \frac{P_l(4)(5-4\alpha_4)}{(1-\alpha_4)^2} \tag{3.53}$$

Let us define N_l, Q_l, Q_2, and Q_3 as follows:

$$N_1 = \sum_{k=1}^{3} k[P_c(k) + P_e(k) + P_l(k)] + 4[P_l(4) + P_e(4)] - \frac{\omega}{1-\alpha_4} \tag{3.54a}$$

$$Q_1 = \omega + \sigma\alpha_3\alpha_4 + \alpha_4 P_l(4), \tag{3.54b}$$

$$Q_2 = P_f\beta^2 + \delta\beta + \delta\alpha_2, \quad \text{(already defined)}, \tag{3.54c}$$

$$Q_3 = \frac{5(1-\alpha_4)^2(1-\beta)^2 + (\beta + \alpha_4 - 2\alpha_4\beta)}{(1-\alpha_4)^2(1-\beta)^2} \tag{3.54d}$$

Then, the average number of transmissions per packet may be written as

$$\overline{N} = N_1 + \frac{(5-4\alpha_4)}{(1-\alpha_4)^2}Q_1 + \frac{(5-4\beta)\beta}{(1-\beta)^2}Q_2 + \frac{\delta\alpha_3}{(1-\beta)^2(1-\alpha_4)^2}Q_3. \tag{3.55}$$

The throughput efficiency $\eta = 1/\overline{N}$ is then given by

$$\eta = \frac{1}{N_1 + \frac{(5-4\alpha_4)}{(1-\alpha_4)^2}Q_1 + \frac{(5-4\beta)\beta}{(1-\beta)^2}Q_2 + \frac{\delta\alpha_3}{(1-\beta)^2(1-\alpha_4)^2}Q_3}. \tag{3.56}$$

3.4 Spurious Transmissions

A packet delivered may be transmitted again if the feedback message incurs undetectable errors, a problem that is caused by an $(A \rightarrow R)$ error.[4] This unnecessary transmission is called here a *spurious transmission*, and, in this section, an expression is derived for the probability of its occurrence. It turns out that the preceding work has eased the derivation. In deriving the probability of undetected error, this event was included. That is, spurious transmissions were assumed to automatically result in undetected errors.

Lemma 3.4: With the quantities already defined, the probability of a spurious transmission at the kth attempt is given by

$$P_{sp}(k) = \begin{cases} 0 & k=1 \\ P_f & k=2 \\ P_f\beta+\delta & k=3 \\ P_f\beta^2+\delta\beta+\delta\alpha_2 & k=4 \\ (P_f\beta^2+\delta\beta+\delta\alpha_2)\beta^{k-4} \quad + \quad \delta\alpha_3\beta^{k-5}\sum_{j=0}^{k=5}\left(\frac{\alpha_4}{\beta}\right)^j & k\geq 5 \end{cases} \tag{3.57}$$

Proof: All that is needed in the proof of this lemma is to identify the terms in $P_e(k)$ that correspond to spurious transmissions. When the term $\sigma\alpha_3\alpha_4^{k-4}$ is subtracted from $P_e(k)$ in Lemma 3.2, the result is the probability of spurious delivery at the kth attempt. The term removed here, $\sigma\alpha_3\alpha_4^{k-4}$, represents erroneous delivery without any unnecessary transmissions. We can therefore write

$$P_{sp}(k) = P_e(k) - \sigma\alpha_3\alpha_4^{k-4}. \tag{3.58}$$

Substituting from Lemma 3.2 gives the probability that the delivery of a packet at the kth transmission is spurious, i.e.,

4 That is, the receiver has accepted a packet but the feedback channel has changed the Accepted message into a Retransmit message. There is another problem that arises from an $(R \rightarrow A)$ error in the return channel. This is the loss of packets, which has already been discussed.

$$P_{sp}(k) = \begin{cases} 0 & k=1 \\ P_f & k=2 \\ P_f\beta+\delta & k=3 \\ P_f\beta^2+\delta\beta+\delta\alpha_2 & k=4 \\ (P_f\beta^2+\delta\beta+\delta\alpha_2)\beta^{k-4} \quad + \quad \delta\alpha_3\beta^{k-5}\sum_{j=0}^{k=5}\left(\frac{\alpha_4}{\beta}\right)^j & k\geq 5 \end{cases} \tag{3.59}$$

The result can then be used to derive the average probability of a spurious transmission. As transmission proceeds, some will be legitimate, and others will be spurious. For P packets, we can count the illegitimate transmissions, and also find the total number of transmissions. The ratio of these two numbers will give the relative frequency of illegitimate transmissions.

Given $\quad P_{SP} \;\underline{\Delta}\; \lim_{P\to\infty}\left\{\frac{\overline{N}_{SP}(P)}{\overline{N}(P)}\right\},$

where $\overline{N}_{SP}(P) \quad = \quad$ average number of spurious transmissions for P packets, and

$\overline{N}(P) \quad = \quad$ average number of transmissions for P packets.

With these definitions, we have

Theorem 3.3: The average probability of a spurious transmission is given by

$$P_{SP} = \eta\left(P_f+P_f\delta+\frac{Q_2}{1-\beta}+\frac{\delta\alpha_3}{(1-\beta)(1-\alpha_4)}\right). \tag{3.60}$$

Proof: The numbers $\overline{N}_{SP}(P)$ and $\overline{N}(P)$ are given by the equations

$$\overline{N}_{SP}(P) = P\sum_{k=1}^{\infty}P_{sp}(k) \qquad \text{and} \qquad \overline{N}(P) = P\sum_{k=1}^{\infty}kP_{dep}(k) = P\overline{N}. \tag{3.61}$$

Thus, using the definition of P_{SP} given above, we have

$$P_{SP} = (1/\overline{N})\sum_{k=1}^{\infty}P_{sp}(k) = \eta\sum_{k=1}^{\infty}P_{sp}(k), \tag{3.62}$$

where η is the throughput efficiency. Substituting from Lemma 3.4, we have

$$P_{SP} = \eta\left(P_f+P_f+\delta+Q_2\sum_{k=4}^{\infty}\beta^{k-4} \quad + \quad \delta\alpha_3\sum_{k=5}^{\infty}\beta^{k-5}\sum_{j=0}^{k-5}\left(\frac{\alpha_4{}^j}{\beta}\right)\right), \tag{3.63}$$

where $Q_2 = P_f + \delta\beta + \delta\alpha_2$ (as already previously defined). The result is then found by evaluating the sums in the expression. This gives

$$P_{SP} = \eta\left(P_f + P_f\delta + \frac{Q_2}{1-\beta} + \frac{\delta\alpha_3}{(1-\beta)(1-\alpha_4)}\right), \tag{3.64}$$

as required. This unwanted transmission is one important consequence of feedback errors in ARQ systems, and one which needs to receive more attention.

3.5 The Average Probability of Packet Loss

The average probability of packet loss is given by the following theorem:

Theorem 3.4: Define the average probability of packet loss as follows:

$$P_L = \lim_{P\to\infty}\left\{\frac{\text{average number of packets lost in P packets}}{\text{average number of packets transmitted in P packets}}\right\}$$

Using $S_L = \sum_{k=1}^{4} P_l(k)$, the average probability of packet loss is then given by

$$P_L = S_L + \left(\frac{\alpha_4 P_l(4)}{1-\alpha_4}\right) \tag{3.65}$$

Proof: With the average probability of loss as defined above, we have

$$P_L = \lim_{P\to\infty}\left\{\frac{\text{average number of packets lost in P packets}}{\text{average number of packets transmitted in P packets}}\right\}$$

$$= \lim_{P\to\infty}\left\{\frac{P\sum_{k=1}^{\infty}P_l(k)}{P}\right\}$$

$$= \sum_{k=1}^{\infty}P_l(k) = S_L + \sum_{k=5}^{\infty}P_l(k) \tag{3.66}$$

Substituting for $P_l(k)$ from Lemma 3.3, gives

$$P_L = S_L + \left(\frac{\alpha_4 P_l(4)}{1-\alpha_4}\right), \tag{3.67}$$

as required. The loss of packets is another undesirable consequence of feedback errors in ARQ schemes. We shall consider later (in the chapter on improvements) one method that reduces the number of lost packets.

3.6 Packet Delay

A packet entering the system is first queued (incurring queuing delay) then transmitted (incurring transmission delay). The total delay, $\overline{d}$, therefore, has two components. If the mean arrival rate is λ packets per second and the mean service time is $\overline{m}$ seconds, then for the M/G/1 queue, Hayes [81] shows that the mean delay is given by the following equation, which is known as the Pollaczek-Khintchin formula:

$$\overline{d} = \overline{m} + \frac{\lambda\overline{m^2}}{2(1-\rho)}, \tag{3.68}$$

where $\overline{m}$ is the mean time to transmit a packet once it is out of the queue, $\rho=\lambda\overline{m}$ is the load on the system, and $\overline{m^2}$ is the second moment of the service time. The mean service $\overline{m}$ is related to the throughput as follows:

$$\overline{m} = \overline{N}T_P = \frac{T_P}{\eta}, \tag{3.69}$$

where η is the throughput efficiency, $\overline{N}$ is the mean number of transmissions per packet, and T_p is the transmission time for one packet[5]. A lower bound to the delay can be obtained by noting that $\overline{m^2} \geq (\overline{m})^2$.

In this case, the delay is given by

$$\overline{d} = \frac{T_p}{\eta}\left(1 + \frac{\rho}{2(1-\rho)}\right). \tag{3.70}$$

It is evident from this equation that for constant load conditions, the delay is inversely proportional to the throughput efficiency. For very light loads (ρ very small), the second term in the large parentheses can be ignored; the delay can be inferred entirely from the throughput efficiency. As the load approaches unity, the queuing component of the delay dominates. The results presented in the thesis can be used to obtain the delay lower bound provided the load is known.

3.7 Summary

In this chapter, expressions have been derived for the performance parameters of the GH-II ARQ scheme. These are now gathered here as a summary in Table 3.1. Since the GH-II ARQ scheme as presented here is an improvement on the selective repeat ARQ scheme (SR-ARQ) scheme, it seems reasonable to examine the performance of the SR-ARQ with errors in the feedback channel and to determine if the advantages of GH-II ARQ still apply here, as they do in noiseless feedback situations. In the next chapter, the SR-ARQ scheme is given the same kind of treatment that the GH-II ARQ scheme has been given. In a later chapter, both methods will be examined under noiseless feedback and comparisons will be made.

5 $T_p = P_{size}/R$, where P_{size} (bits) is the packet size and R (bits/s) is the transmission rate.

Table 3.1 Performance Parameters of the GH-II ARQ with Noisy Feedback

Probability of Undetected Error	$P_{UD} = \dfrac{S_e(1-\beta)(1-\alpha_4) + Q_2(1-\alpha_4)\beta + \delta\alpha_3\beta}{(S_e+S_c)(1-\alpha_4)(1-\beta)+\alpha_4(\omega+\sigma\alpha_3)(1-\beta)+\delta\alpha_3\beta+Q_2(1-\alpha_4)\beta}.$
Probability of Correct Delivery	$P_C = \dfrac{S_c(1-\beta)(1-\alpha_4) + \omega\alpha_4(1-\beta)}{(S_e+S_c)(1-\alpha_4)(1-\beta)+\alpha_4(\omega+\sigma\alpha_3)(1-\beta)+\delta\alpha_3\beta+Q_2(1-\alpha_4)\beta}.$
Probability of Packet Loss	$P_L = S_L + \left(\dfrac{\alpha_4 P_l(4)}{1-\alpha_4}\right).$
Probability of Spurious Delivery	$P_{SP} = \eta\left(P_f + P_f\delta + \dfrac{Q_2}{1-\beta} + \dfrac{\delta\alpha_3}{(1-\beta)(1-\alpha_4)}\right),$
Throughput Efficiency	$\eta = \dfrac{1}{N_1 + \frac{(5-4\alpha_4)}{(1-\alpha_4)^2}Q_1 + \frac{(5-4\beta)\beta}{(1-\beta)^2}Q_2 + \frac{\delta\alpha_3}{(1-\beta)^2(1-\alpha_4)^2}Q_3}$

We further assume that the error process in the channel, as it affects each transmission, can be modelled as a Markov Chain. Finally, for the return channel, we define the events T and F to be

T = { *the feedback message as received is true* },
F = { *the feedback message as received is false* }.

The assumption made here is that we have only forward error correction in the feedback channel. Each transmitted packet may give rise to one of four events. The first three events are the following. The packet may be completely lost, delivered with errors, or delivered without errors. These events correspond to the packets departure from the queue at the transmitter. Finally, the packet may be retransmitted. In this case, it still remains in the queue at the transmitter. Our goal in analyzing this scheme is to finally be able to obtain expressions for the throughput efficiency, the probability of undetected error, and the probability of packet loss. To find the above performance parameters, we would need to first determine the probability that at its *k*th transmission, a packet

(1) leaves the queue at the transmitter **(Dep)**,
(2) is correctly delivered **(C)**,
(3) is delivered with errors **(E)**,
(4) is lost **(L)**.

These probabilities are $P_{dep}(k)$, $P_c(k)$, $P_e(k)$, and $P_l(k)$, respectively. The events given here are such that the last three are an exhaustive decomposition of the first, i.e., **Dep** = **C** ∪ **E** ∪ **L**. This allows us to write

$$P_{dep}(k) = P_c(k) + P_e(k) + P_l(k), \tag{4.1}$$

a fact which will be used later.

4.1 Packet Loss and Correct Delivery at the *k*th Transmission

We begin our analysis by seeking the expression for $P_l(k)$, the probability that a packet is lost at the *k*th transmission. When the packet is transmitted, it can end up with correctable errors (C), undetectable errors (E), or detectable, but not correctable errors (D)[1]. Following the arrival of a packet, the receiver determines its status, then sends a feedback message on the return channel. There are four possibilities, 1(a), 1(b), 2(a), and 2(b), as described below.

1 The number of errors detectable is larger than the number of errors that a code can correct. So there will be instances when a retransmission will have to be requested, in the hope that the extra information provided will help correct the errors detected.

1(a) If the arrival of a packet ends in either C or E, the feedback message should indicate that a delivery has been made. Thus a true feedback message will terminate the transmission of the packet on the first try.

1(b) However, if the packet's arrival results in either C or E as above, and the feedback message is false, another transmission of the packet will follow. The result is that transmission of the packet continues *beyond* the first try.

2(a) Suppose, on the one hand, that the arrival of the packet results in D, and the feedback message is false. The transmitter will begin sending the next packet, which means that the transmission of the original packet terminates with the first try.

2(b) On the other hand, if the arrival results in D and the feedback message is true, then another transmission of the same packet will follow, which means that transmission of the packet will continue *beyond* the first try.

We have in 1(a), a delivery (correct or incorrect); in 1(b), an unnecessary transmission; in 2(a), a packet loss; and in 2(b), a retransmission. Although explicit reference will not always be made to the items in the list given here, the scenarios they represent will be heavily used in the derivations that are soon to follow.

Let us introduce the variables σ_1, σ_2, and α as follows:

$$\sigma_1 = P(D)P(T/D)P(C/T)P(T/C), \quad (4.2a)$$

$$\sigma_2 = P(D)P(F/D), \quad \text{and} \quad (4.2b)$$

$$\alpha = P(T/D)P(D/T). \quad (4.2c)$$

Lemma 4.1: With the variables defined above, we let $P_c(k)$ be the probability that a packet is correctly delivered at its *k*th transmission, and $P_l(k)$, the probability that a packet is lost at its *k*th transmission; then,

$$P_c(1) = P(C)P(T/C), \quad (4.3a)$$

$$P_c(k) = \sigma_1\alpha^{k-2}, \quad k \geq 2 \quad (4.3b)$$

$$P_l(k) = \sigma_2\alpha^{k-1}, \quad k \geq 2 \quad (4.3c)$$

Proof: The proof continues by following the event sequences that terminate in correct delivery and the loss of packets. Taking the possibilities given by 1(a) and 2(a), we obtain the events leading to the termination of transmission on the first try. They are illustrated in Figure 4.1.

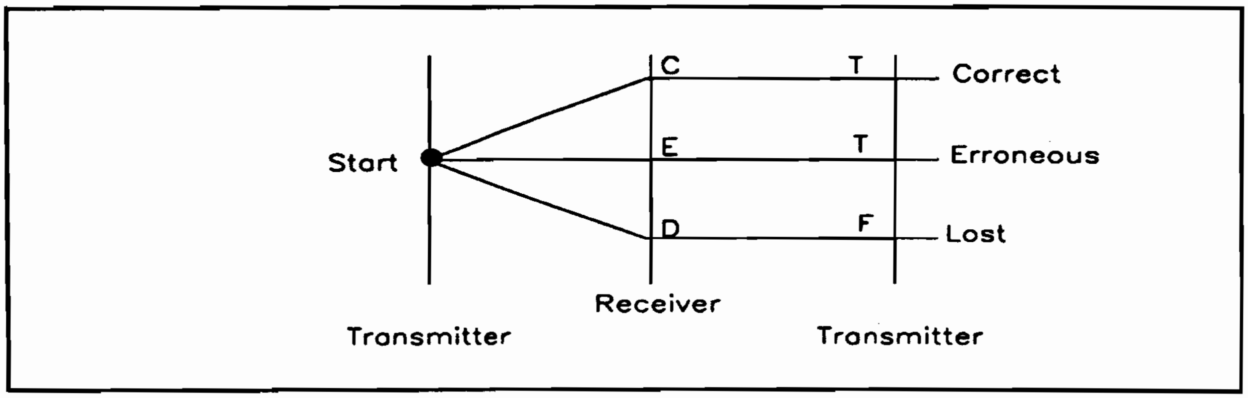

Figure 4.1 Departure of a Packet at the First Transmission

The events leading to the departure of a packet at the first transmission, as depicted in Figure 4.1, have probabilities

$$P_c(1) = P(T/C)P(C) \tag{4.4a}$$

$$P_e(1) = P(T/E)P(E) \tag{4.4b}$$

$$P_l(1) = P(F/D)P(D) = \sigma_2 \tag{4.4c}$$

Taking 1(b) and 2(b), and following the same line of reasoning, we obtain the events leading to the termination of transmission at the second try, the third try, etc. A packet's transmission will terminate when it is either delivered or lost. Figure 4.2 illustrates the loss of a packet at the first try. A loss after many transmissions is also possible, as will be apparent later.

In Figure 4.3, we illustrate the departure of a packet on the second transmission. From this diagram, we have

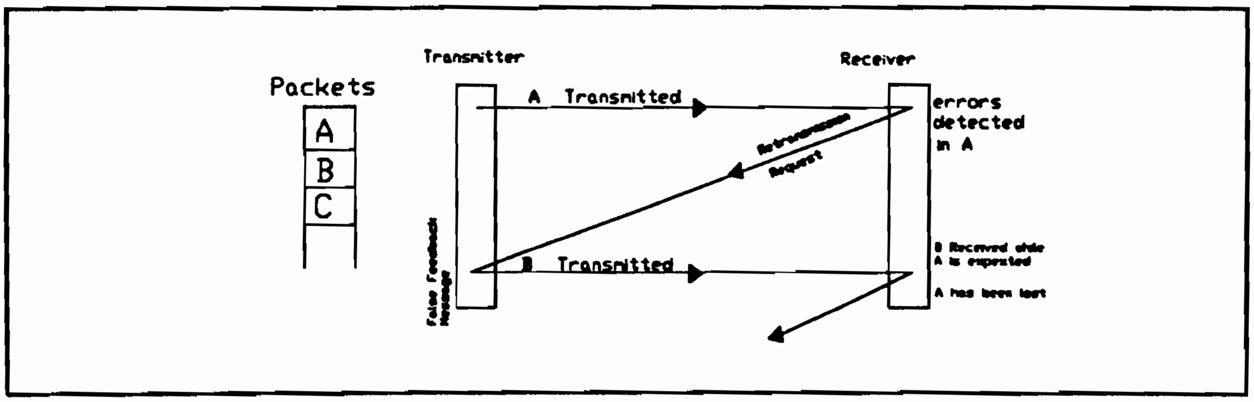

Figure 4.2 The Loss of a Packet

$$P_c(2) \;=\; P(T/C)P(C/T)P(T/D)P(D), \tag{4.5a}$$

$$P_l(2) \;=\; P(F/D)P(D/T)P(T/D)P(D). \tag{4.5b}$$

From this point on, we shall determine only $P_c(k)$, and $P_l(k)$, leaving out $P_e(k)$, since the information provided by $P_e(k)$ can, in fact, be obtained from the other two. Extending the above procedure to three steps results in the diagram of Figure 4.4.

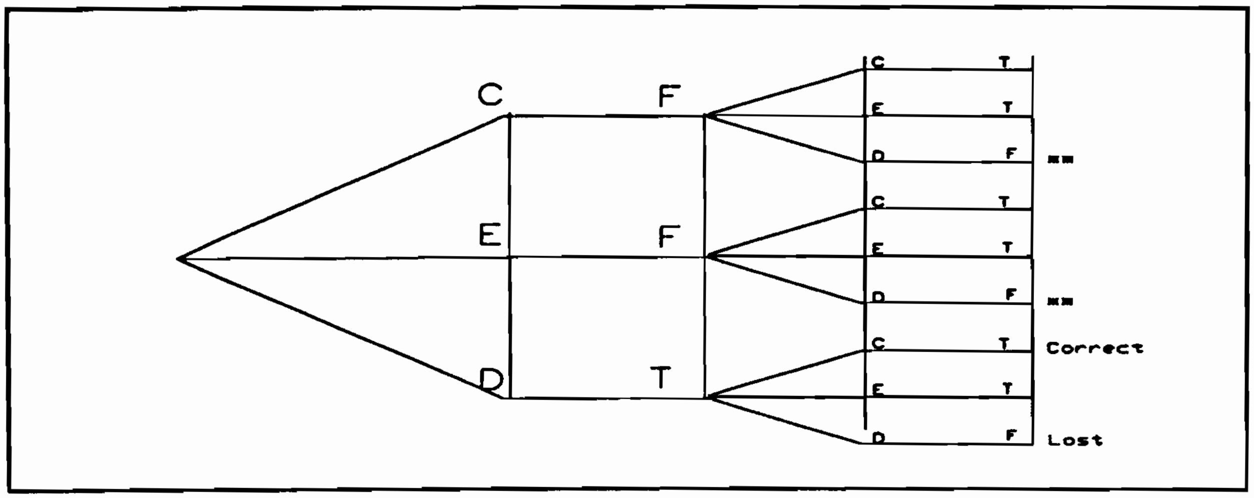

Figure 4.3 Departure of a Packet on the Second Transmission[2]

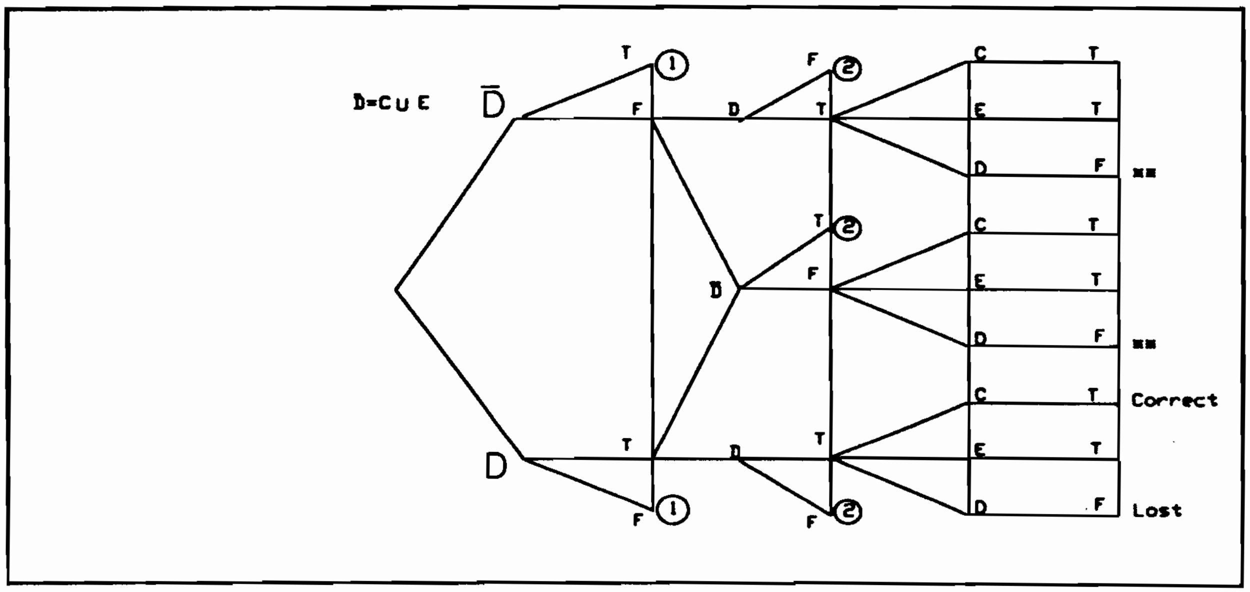

Figure 4.4 Departure of a Packet on the Third Transmission

2 In this Figure, and in subsequent figures, the points labelled ** denote the loss of a packet that has already been delivered. This is not a real loss, since there has already been a delivery. It is really a spurious transmission that has turned into a "loss."

In Figure 4.4, $\overline{D}$ is the complement of D, and is the union of C and E; i.e., $\overline{D} = C \cup E$. The circled numbers show the termination of transmission at the try indicated by the numbers. The expressions for the probabilities of correct delivery and packet loss are found by following the highlighted paths in Figure 4.4. Reading the event sequence, we have

$$P_c(3) = P(T/C)P(C/T)P(T/D)P(D/T)P(T/D)P(D)$$

$$= P(D)(T/D)P(C/T)P(T/C)\cdot[P(D/T)P(T/D)] \quad (4.6a)$$

$$P_l(3) = P(F/D)P(D/T)P(T/D)P(D/T)P(T/D)P(D)$$

$$= P(D)(F/D)\cdot[P(D/T)P(T/D)]^2 \quad (4.6b)$$

Defining $\sigma_1 = P(C/T)P(T/C)$, $\sigma_2 = P(D)P(F/D)$, and $\alpha = P(T/D)P(D/T)$, and continuing as before, we obtain the following expressions

$$P_c(1) = P(C)P(T/C), \quad (4.7a)$$

$$P_c(k) = \sigma_1\alpha^{k-2} \quad k \geq 2 \quad (4.7b)$$

$$P_l(k) = \sigma_2\alpha^{k-1} \quad k \geq 1 \quad (4.7c)$$

and completes the proof of the lemma. We turn next to the delivery of a packet after a certain number of transmission attempts.

4.2 Delivery of a Packet at the *k*th Transmission

The expression for the probability of delivery at the *k*th attempt is given by the Lemma 4.2 below.

Let $\rho_1 = P([DT \cup \overline{D}F]/[DT \cup \overline{D}F])$, and (4.8a)

$$\sigma_3 = P(\overline{D}T/[DT \cup \overline{D}F])P([DT \cup \overline{D}F]). \quad (4.8b)$$

Lemma 4.2: Let $P_{del}(k)$ be the probability of packet delivery at the *k*th transmission, then this probability is given by

$$P_{del}(k) = \begin{cases} P(\overline{D}T)P(T/\overline{D}), & k = 1 \\ \sigma_3\rho_1^{k-2} & k \geq 2 \end{cases} \quad (4.9)$$

Proof: Here, we again have $\overline{D} = C \cup E$. The probabilities of delivery at the first and the second transmissions are, respectively,

$$P_{del}(1) = P(\overline{D}T)P(T/\overline{D}), \quad \text{and} \tag{4.10a}$$

$$P_{del}(2) = P([DT \cup \overline{D}F]\overline{D}T). \tag{4.10b}$$

The events are read from left to right; thus, $[DT \cup \overline{D}F]\overline{D}T$ means that the square brackets precede the $\overline{D}T$. Continuing, we obtain $P_{del}(3)$ as

$$P_{del}(3) = P([DT \cup \overline{D}F]^2\overline{D}T), \tag{4.11}$$

where we have used $[DT \cup \overline{D}F]^2$ to mean the occurrence of the event in square brackets two times. In general, we find that

$$P_{del}(k) = P([DT \cup \overline{D}F]^{k-1}\overline{D}T). \tag{4.12}$$

We must now determine the following two probabilities: $P([DT \cup \overline{D}F]/[DT \cup \overline{D}F])$ and $P([\overline{D}T]/[DT \cup \overline{D}F])$. We begin with the first,

$$P([DT \cup \overline{D}F]/[DT \cup \overline{D}F]) = \frac{P(DTDT \cup DT\overline{D}F \cup \overline{D}FDT \cup \overline{D}F\overline{D}F)}{P(DT \cup \overline{D}F)}. \tag{4.13}$$

Both the numerator and denominator are now the probabilities of unions of disjoint events. The numerator is the sum of the following terms:

$$P(DTDT) = P(D)P(T/D)P(D/T)P(T/D), \tag{4.14a}$$

$$P(DT\overline{D}F) = P(D)P(T/D)P(\overline{D}/T)P(F/\overline{D}), \tag{4.14b}$$

$$P(\overline{D}FDT) = P(\overline{D})P(F/\overline{D})P(D/F)P(T/D), \tag{4.14c}$$

$$P(\overline{D}F\overline{D}F) = P(\overline{D})P(F/\overline{D})P(\overline{D}/F)P(F/\overline{D}), \tag{4.14d}$$

and the denominator is the sum of $P(DT) = P(D)P(T/D)$ and $P(\overline{D}F) = P(\overline{D})P(F/\overline{D})$. Similarly,

$$P(\bar{D}T/[DT \cup \bar{D}F]) = \frac{P(DT\bar{D}T \cup \bar{D}F\bar{D}T)}{P(DT \cup \bar{D}F)} \tag{4.15a}$$

$$= \frac{P(DT\bar{D}T) + P(\bar{D}F\bar{D}T)}{P(DT) + P(\bar{D}F)}. \tag{4.15b}$$

Occurrence of the event $[DT \cup \bar{D}F]$ *k-1* times is equivalent to *k-2* transitions from $[DT \cup \bar{D}F]$ to itself. In the expression for $P_{del}(k)$, we have one final transition to the event $[\bar{D}T]$. Thus, the probability of delivery at the *k*th transmission is

$$P_{del}(k) = P([DT \cup \bar{D}F] \cdot (P([DT \cup \bar{D}F]/[DT \cup \bar{D}F])^{k-2} \cdot P(\bar{D}T/[DT \cup \bar{D}F]). \tag{4.16}$$

Let $\rho_1 = P([DT \cup \bar{D}F]/[DT \cup \bar{D}F])$ and $\sigma_3 = P(\bar{D}T/[DT \cup \bar{D}F]) \cdot P([DT \cup \bar{D}F])$. Then, the expression for $P_{del}(k)$ can now be written as

$$P_{del}(k) = \begin{cases} P(\bar{D}) \cdot P(T/\bar{D}), & k = 1 \\ \sigma_3 \rho_1^{k-2} & k \geq 2 \end{cases}, \tag{4.17}$$

which completes the proof. Thus far, we have used the preceding diagrams to guide our progress. In what follows, we shall apply what has so far been acquired to derive the expressions for performance parameters such as the throughput efficiency, the probabilities of undetected error, packet loss, and correct delivery. We shall begin with the throughput efficiency.

4.3 The Throughput Efficiency of the SR-ARQ Scheme

There are two ways in which a packet may leave the transmitter's queue: it can either be lost or delivered to the user. If delivered, it may be error-free or contain undetected errors. The events leading to a packet leaving the queue at the *k*th transmission are then

E = { *erroneous delivery* }
C = { *error-free delivery* }
L = { *packet loss* }

The previously defined events F, D, T and E still remain as defined. The departure probability at a general transmission is given by Lemma 4.3.

Define $\sigma_2 = P(D)P(F/D),$ (4.18*a*)

$\sigma_3 = P(D)P(T/D)P(\overline{D}/T)P(T/\overline{D}) + P(\overline{D})P(F/\overline{D})P(\overline{D}/F)P(T/\overline{D}),$ (4.18*b*)

$\alpha = P(T/D)P(D/T),$ and (4.18*c*)

$\rho = P([DT \cup \overline{D}F]/[DT \cup \overline{D}F]).$ (4.18*d*)

Lemma 4.3: Let $P_{dep}(k)$ be the probability of departure at a general transmission, then

$$P_{dep}(k) = \begin{cases} \sigma_2 + P(\overline{D}T)P(T/\overline{D}), & k = 1 \\ \sigma_2\alpha^{k-1} + \sigma_3\rho_1^{k-2}, & k \geq 2 \end{cases} . \quad (4.19)$$

Proof: The departure at time-k is the union of the events E, C, and L, as defined above. That is, **Dep** = **C** $\cup$ **E** $\cup$ **L**, and the probability of this event is sum of the individual probabilities, since the events are mutually exclusive. We have

$$\begin{aligned} P_{dep} &= P(E \cup C) + P(L) \\ &= P\{\text{delivery at the kth transmission}\} + P\{\text{packet loss at the kth transmission}\} \\ &= P_{del}(k) + P_l(k). \end{aligned} \quad (4.20)$$

Substituting for $P_{del}(k)$ and $P_l(k)$ from Lemma 4.1 and Lemma 4.2, we obtain

$$P_{dep}(k) = \begin{cases} \sigma_2 + P(\overline{D}T)P(T/\overline{D}), & k = 1 \\ \sigma_2\alpha^{k-1} + \sigma_3\rho_1^{k-2}, & k \geq 2 \end{cases} , \quad (4.21)$$

where $\sigma_2 = P(D)P(F/D),$ (4.22*a*)

$\sigma_3 = P(D)P(T/D)P(\overline{D}/T)P(T/\overline{D}) + P(\overline{D})P(F/\overline{D})P(\overline{D}/F)P(T/\overline{D}),$ (4.22*b*)

$\alpha = P(T/D)P(D/T),$ and (4.22*c*)

$\rho = P([DT \cup \overline{D}F]/[DT \cup \overline{D}F]).$ (4.22*d*)

This completes the proof. The throughput efficiency is then given by the following theorem.

Theorem 4.1: With the definitions given above, the throughput efficiency is given by the following expression

$$\eta = \frac{(1-\rho_1)^2(1-\alpha)^2}{[\sigma_2 + P(\overline{D})\cdot P(T/\overline{D})]\cdot(1-\rho_1)^2(1-\alpha)^2 + \alpha\sigma_2(2-\alpha)(1-\rho_1)^2 + \sigma_3(2-\rho_1)(1-\alpha)^2}. \quad (4.23)$$

Proof: The proof uses the results of Lemma 4.3 by first finding the average number of transmissions per packet and then takes the reciprocal. To proceed, suppose that we have P packets to transmit. The average number of transmissions is

$$\overline{N}(P) = \sum_{i=1}^{P}\sum_{k=1}^{\infty} kPr\{i\text{th packet departs at the } k\text{th transmission}\} \quad (4.24a)$$

$$= \sum_{i=1}^{P}\sum_{k=1}^{\infty} kP_{dep}(k) \quad (4.24b)$$

$$= P\sum_{k=1}^{\infty} kP_{dep}(k) = P\overline{N}, \quad (4.24c)$$

where $\overline{N}$ is the average number of transmission per packet. Substituting for $P_{dep}(k)$ from above, we obtain

$$\overline{N} = [\sigma_2 + P(\overline{D})P(T/\overline{D})] + \sum_{k=2}^{\infty}[\sigma_2\alpha^{k-1} + \sigma_3\rho_1^{k-2}]. \quad (4.25a)$$

Evaluating the sum gives the result as

$$\overline{N} = [\sigma_2 + P(\overline{D})P(T/\overline{D})] + \frac{\alpha\sigma_2(2-\alpha)}{(1-\alpha)^2} + \frac{\sigma_3(2-\rho_1)}{(1-\rho_1)^2}. \quad (4.25b)$$

The throughput efficiency η is then the reciprocal of $\overline{N}$, and is given by

$$\eta = \frac{(1-\rho_1)^2(1-\alpha)^2}{[\sigma_2 + P(\overline{D})\cdot P(T/\overline{D})]\cdot(1-\rho_1)^2(1-\alpha)^2 + \alpha\sigma_2(2-\alpha)(1-\rho_1)^2 + \sigma_3(2-\rho_1)(1-\alpha)^2}. \quad (4.26)$$

This completes the proof of the theorem.

4.4 Correct Delivery

Correct delivery is the delivery of a packet without any errors. Let P_C be the probability that a packet is correctly delivered. Then, the following theorem gives the probability of correct delivery

$$\text{Let} \quad P_C = \lim_{P \to \infty} \left\{ \frac{\text{average number of correct deliveries for P packets}}{\text{average number of total deliveries for P packets}} \right\}.$$

Theorem 4.2: Given $\sigma_1 = P(D)P(T/D)P(C/T)P(T/C)$, ρ_1, σ_3, and α as defined previously, the probability of correct delivery is

$$P_C = \frac{P(C) \cdot P(T/C) + \sigma_1/(1-\alpha)}{P(\overline{D}) \cdot P(T/\overline{D}) + \sigma_3/(1-\rho_1)}. \tag{4.27}$$

Proof: Define the random variable $C_i(k)$ as follows. The average probability of correct delivery is found from its definition given above as

$$P_C = \lim_{P \to \infty} \left\{ \frac{\overline{N}_c(P)}{\overline{N}_{del}(P)} \right\},$$

where the numerator and denominator represent the average numbers, respectively, of correct deliveries and total deliveries, in P packets. The numerator is given by

$$\overline{N}_c(P) = \sum_{i=1}^{P} \sum_{k=1}^{\infty} E\{C_i(k)\} \tag{4.28a}$$

$$= P \sum_{k=1}^{\infty} P_c(k) = P\overline{N}_c, \tag{4.28b}$$

and $P\overline{N}_c$ is the average number of packets correctly delivered, with $\overline{N}_c$ given by

$$\overline{N}_C = \sum_{k=1}^{\infty} P_c(k) = P_c(1) + \sum_{k=2}^{\infty} \sigma_1 \alpha^{k-2}, \tag{4.29}$$

and $\sigma_1 = P(D) \cdot P(T/D) \cdot P(C/T) \cdot P(T/C)$. Evaluating the above sum, we obtain,

$$\overline{N}_c = P(C) \cdot P(T/C) + \left(\frac{\sigma_1}{1-\alpha} \right). \tag{4.30}$$

$$\text{Similarly,} \quad \overline{N}_{del}(P) = P \sum_{k=1}^{\infty} P_{del}(k) = P\overline{N}_{del}, \tag{4.31}$$

where $P\overline{N}_{del}$ is the average number of deliveries made for the P packets. Since multiple deliveries are possible, this number need not be equal to P. The quantity $\overline{N}_{del}$ is found from the sum

$$\overline{N}_{del} = \sum_{k=1}^{\infty} P_{del}(k). \tag{4.32}$$

Using (4.17), we have

$$\overline{N}_{del} = P(\overline{D}) \cdot P(T/\overline{D}) + \sum_{k=1}^{\infty} \sigma_3 \rho_1^{k-2} \tag{4.33}$$

which, when evaluated, results in

$$\overline{N}_{del} = P(\overline{D}) \cdot P(T/\overline{D}) + \left(\frac{\sigma_3}{1-\rho_1}\right) \tag{4.34}$$

The expression for the probability of correct delivery is then found by taking the ratio of these two averages. We have that

$$P_C = \frac{P(C) \cdot P(T/C) + \sigma_1/(1-\alpha)}{P(\overline{D}) \cdot P(T/\overline{D}) + \sigma_3/(1-\rho_1)}. \tag{4.35}$$

This ends the proof.

4.5 The Loss of a Packet and Erroneous Delivery

The loss of a packet for a general transmission is discussed in Section 3.1. The expressions developed there will now be used to derive the average probability of a packet loss. Let P_L be the probability that a packet is lost, with the following definition:

$$P_L = \lim_{P \to \infty} \left\{ \frac{\text{average number of lost packets for P packets}}{\text{average number of packets transmitted for P packets}} \right\}.$$

We have also the previously defined σ_2 and α, as being

$$\sigma_2 = P(D) \cdot P(F/D), \quad \text{and} \tag{4.36a}$$

$$\alpha = P(T/D) \cdot P(D/T) \tag{4.36b}$$

The following theorem gives the probability of packet loss.

Theorem 4.3: With P_L, σ_2, and α as given above, the probability of packet loss is

$$P_L = \left(\frac{\sigma_2}{1-\alpha}\right). \tag{4.36c}$$

Proof: From the definition of P_L given above, we have

$$P_L = \lim_{P\to\infty}\left\{\frac{\overline{N}_l(P)}{P}\right\} = \lim_{P\to\infty}\left\{\left(\frac{1}{P}\right)\left(P\sum_{k=1}^{P}P_l(k)\right)\right\} \tag{4.37a}$$

$$= \lim_{P\to\infty}\left(\sum_{k=1}^{P}P_l(k)\right). \tag{4.37b}$$

Therefore, the probability of packet loss is given by the sum

$$P_L = \sum_{k=1}^{\infty}P_l(k). \tag{4.38}$$

Substituting for $P_l(k)$ from Lemma 4.1, results in

$$P_L = \sum_{k=1}^{\infty}\sigma_2\alpha^{k-1} = \left(\frac{\sigma_2}{1-\alpha}\right), \tag{4.39}$$

and the proof is complete.

The loss of packets occurs because of errors in the feedback channel. If the return channel were error-free, the probability of loss would be zero, however, there is no physical channel that is error-free. In practical systems, quite often there is as much information transmitted in the reverse direction as in the forward direction. Therefore, if error control is employed in both directions, the error performance will be the same in both cases. The argument that the reverse channel has a lower data rate holds only in those cases previously described.

4.6 Erroneous Delivery

The probability of incorrect delivery, otherwise called the probability of undetected error, can be derived in a manner similar to that employed for the probability of correct delivery. The definition of the probability of undetected error is

$$P_{UD} = \lim_{P\to\infty}\left\{\frac{\text{average number of incorrect deliveries for P packets}}{\text{average number of deliveries for P packets}}\right\}.$$

Theorem 4.4: The probability of undetected error is

$$P_{UD} = \frac{P(E)\cdot P(T/E) + \sigma_3/(1-\rho_1) - \sigma_1/(1-\alpha)}{P(\overline{D})\cdot P(T/\overline{D}) + \sigma_3/(1-\rho_1)}. \tag{4.40}$$

Proof: Using the fact that $P_C + P_{UD} = 1$, we can use the result of Theorem 3.4 to write the probability of undetected error as

$$P_{UD} = 1-\left(\frac{P(C)\cdot P(T/C) + \sigma_1/(1-\alpha)}{P(\overline{D})\cdot P(T/\overline{D}) + \sigma_3/(1-\rho_1)}\right) \tag{4.41a}$$

$$= \frac{P(\overline{D})\cdot P(T/\overline{D}) - P(C)\cdot P(T/C) + \sigma_3/(1-\rho_1) - \sigma_1/(1-\alpha)}{P(\overline{D})\cdot P(T/\overline{D}) + \sigma_3/(1-\rho_1)}. \tag{4.41b}$$

Using the fact that $\overline{D} = C \cup E$, and noting that $P(\overline{D})\cdot P(T/\overline{D}) - P(C)\cdot P(T/C) = P(\overline{D}T) - P(CT)$, we have

$$P(\overline{D})\cdot P(T/\overline{D}) - P(C)\cdot P(T/C) = P(\overline{D}T) - P(CT) \tag{4.42a}$$

$$= P([C \cup E]T) - P(CT) \tag{4.42b}$$

$$= P(CT) + P(ET) - P(CT) \tag{4.42c}$$

$$= P(ET) = P(E)\cdot P(T/E). \tag{4.42d}$$

This ends the proof.

4.7 Spurious Transmissions

When the result of a transmission is $\overline{D}$, i.e., when there has been a delivery, a true feedback message should indicate that a delivery has been made, so that the next packet can be transmitted. If the feedback message is false following $\overline{D}$, another transmission of the already delivered packet will follow. This is an unnecessary transmission, which is called a spurious transmission.

Define $P_{SP} = \lim_{P\to\infty}\left\{\frac{\text{average number of spurious deliveries for P packets}}{\text{average number of deliveries for P packets}}\right\}$,

and let $\sigma_4 = P(DT \cup \overline{D}F)\cdot P(\overline{D}F/[DT \cup \overline{D}F])$. (4.43)

Theorem 4.5: The average probability P_{SP} of a spurious transmission is given as

$$P_{SP} = \eta\left(P(\overline{D}F) + \frac{\sigma_4}{1-\rho_1}\right). \tag{4.44}$$

Proof: The probability $P_{sp}(k)$ that the kth transmission of a packet is spurious is found by following the event sequences that end in $\overline{D}F$. By selecting those sequences, we have

$$P_{sp}(1) = 0, \tag{4.45a}$$

$$P_{sp}(2) = P(\overline{D}F), \tag{4.45b}$$

$$P_{sp}(3) = P([DT \cup \overline{D}F]\overline{D}F), \tag{4.45c}$$

$$P_{sp}(k) = P([DT \cup \overline{D}F]^{k-2}\overline{D}F), \quad k \geq 3. \tag{4.45d}$$

As before, we have

$$P(\overline{D}F/[DT \cup \overline{D}F]) = \frac{P(\overline{D}FDT \cup \overline{D}F\overline{D}F)}{P([DT \cup \overline{D}F])} \tag{4.46a}$$

$$= \frac{P(\overline{D}FDT) + P(\overline{D}F\overline{D}F)}{P(DT) + P(\overline{D}F)}. \tag{4.46b}$$

The following quantities are defined as

$$\rho_1 = P([DT \cup \overline{D}F]/[DT \cup \overline{D}F]), \tag{4.47a}$$

$$\sigma_3 = P(\overline{D}T/[DT \cup \overline{D}F]) \cdot P([DT \cup \overline{D}F]), \text{and} \tag{4.47b}$$

$$\sigma_4 = P([DT \cup \overline{D}F]) \cdot P(\overline{D}F/[DT \cup \overline{D}F]), \tag{4.47c}$$

where ρ_1 and σ_3 are as previously defined, and σ_4 is new. With these quantities, and the preceding expressions for $P_{sp}(k)$, we have

$$P_{sp} = \begin{cases} 0, & k = 1 \\ P(\overline{D}F), & k = 2 \\ \sigma_4\rho_1^{k-3}, & k \geq 3. \end{cases} \tag{4.48}$$

$$
\begin{aligned}
P_{SP} &= \lim_{P \to \infty} \left\{ \frac{\text{average number of spurious deliveries for P packets}}{\text{average number of deliveries for P packets}} \right\} \\
&= \lim_{P \to \infty} \left\{ \frac{\overline{N}_{SP}(P)}{\overline{N}(P)} \right\} = \lim_{P \to \infty} \left\{ \frac{P\overline{N}_{SP}}{P\overline{N}} \right\} = \lim_{P \to \infty} \left\{ \frac{\overline{N}_{SP}}{\overline{N}} \right\} \qquad (4.49)
\end{aligned}
$$

Since the term in curly brackets is independent of P, we have

$$
P_{SP} = \frac{\overline{N}_{SP}}{\overline{N}} = \eta \overline{N}_{SP}, \qquad (4.50)
$$

where we have used the fact that $\eta = 1/\overline{N}$ is the throughput, already derived. Continuing to substitute for $P_{sp}(k)$, we have

$$
P_{SP} = \eta \sum_{k=1}^{\infty} P_{sp}(k) \qquad (4.51a)
$$

$$
= \eta \left(P(\overline{D}F) + \frac{\sigma_4}{1-\rho_1} \right) \qquad (4.51b)
$$

This ends the proof.

4.8 Summary

By introducing variables that simplify the notation, this chapter has derived expressions for the performance parameters of the selective repeat ARQ scheme. These are now presented here as a summary.

Table 4.1 Performance Parameters of the Selective Repeat ARQ Scheme	
Throughput Efficiency	$\eta = \dfrac{(1-\rho_1)^2(1-\alpha)^2}{[\sigma_2 + P(\overline{D})\cdot P(T/\overline{D})]\,(1-\rho_1)^2(1-\alpha)^2 + \alpha\sigma_2(2-\alpha)\,(1-\rho_1)^2 + \sigma_3(2-\rho_1)\,(1-\alpha)^2}$
Prob. of Undetected error	$P_{UD} = \dfrac{P(E)\cdot P(T/E) + \sigma_3/(1-\rho_1) - \sigma_1/(1-\alpha)}{P(\overline{D})\cdot P(T/\overline{D}) + \sigma_3/(1-\rho_1)}$
Prob. of Correct Delivery	$P_C = \dfrac{P(C)\cdot P(T/C) + \sigma_1/(1-\alpha)}{P(\overline{D})\cdot P(T/\overline{D}) + \sigma_3/(1-\rho_1)}$
Prob. of Packet Loss	$P_L = \dfrac{\sigma_2}{1-\alpha}$
Prob. of Spurious Transmission	$P_{SP} = \eta\left(P(\overline{D}F)\dfrac{\sigma_4}{1-\rho_1}\right)$

CHAPTER V
NOISELESS FEEDBACK AND NOISY FEEDBACK COMPARED

In this chapter we shall derive the performance parameters under the assumption of noiseless feedback. In the first part, the results are for the selective repeat ARQ scheme, and in the second part, they are for the Generalized type-II ARQ scheme (the GH-II ARQ).

5.1 Selective Repeat ARQ Scheme with Noiseless Feedback

Let the events C, D, and D be defined as before, i.e.,

C = *{ errors are correctable }*
D = *{ errors are detectable but not correctable }*
E = *{ errors are undetectable }*.

In this case of noiseless feedback, there is no loss of packets. Figure 5.1 shows the transitions among the states C, D, and E from one transmission to the next.

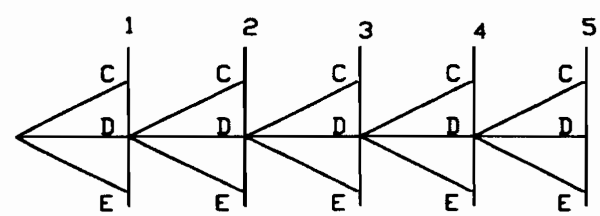

Figure 5.1 Transition Diagram For Error-Free Feedback

Since there are no errors in the feedback channel, once a packet is removed from the queue at the transmitter, it is definitely delivered; there are no lost packets. In fact, there are no spurious transmissions either.

5.1.1 Correct Delivery and Erroneous Delivery

A packet may be error-free or contain errors. The relevant probabilities here are denoted by P_C and P_{UD} for correct delivery and undetected error, respectively. The following theorem gives the corresponding expressions.

Theorem 5.1: Given the events C, D, and E as defined above. The probabilities of correct delivery and undetected error are given, respectively, by

$$P_C = P(C) + \frac{P(C/D)\cdot P(D)}{1-P(D/D)}, \text{ and} \tag{5.1a}$$

$$P_{UD} = P(E) + \frac{P(E/D)\cdot P(D)}{1-P(D/D)}. \tag{5.1b}$$

Proof: Correct delivery of a packet occurs on the first try with probability P(C), i.e.,

$$P_c(1) = P(C). \tag{5.2a}$$

And similarly, the probability of undetected error at the first try is given by

$$P_e(1) = P(E). \tag{5.2b}$$

If the result of the first transmission is the event D, then transmission continues to the second transmission. At this time, the probabilities are given by

$$P_c(2) = P(D)\cdot P(C/D), \text{ and} \tag{5.2c}$$

$$P_e(2) = P(D)\cdot P(E/D), \tag{5.2d}$$

where *P(C/D)*, *P(E/D)*, and *P(D/D)* are D_P-step transition probabilities, and $D_P/2$ is the number of packets transmitted in the time a packet travels from the transmitter to the receiver. Therefore, D_P is the round-trip delay expressed in packet times. In general, we find that

$$P_c(k) = \begin{cases} P(C) & k = 1 \\ P(E/D)\cdot P(D)\cdot (P(D/D))^{k-2} & k \geq 2. \end{cases} \tag{5.3}$$

The probability that a packet is delivered at its *k*th transmission is the sum of $P_c(k)$ and $P_e(k)$. The average number of deliveries per packet is unity. So, if we use the definition of the probability of correct delivery as the ratio of the average number of correct deliveries to the average number of deliveries made, we find that

$$P_C = \sum_{k=1}^{\infty} P_c(k). \tag{5.4}$$

When we substitute for $P_c(k)$ and evaluate the sum, we obtain

$$P_C = P(C) + \frac{P(C/D)\cdot P(D)}{1-P(D/D)}. \tag{5.5}$$

By the same token, the probability of undetected error is

$$P_{UD} = P(E) + \frac{P(E/D)\cdot P(D)}{1-P(D/D)}. \tag{5.6}$$

This ends the proof.

5.1.2 Throughput Efficiency

With a noiseless feedback channel, all the packets that leave the transmitter are delivered; there are no lost packets. Therefore, the delivery of a packet is equivalent to its departure from the queue at the transmitter. The probability that a packet departs at its kth transmission is denoted by $P_{dep}(k)$, and the corresponding probability for a delivery at the kth transmission is denoted by $P_{del}(k)$. By the preceding argument, we have that $P_{del}(k) = P_{dep}(k)$.

Theorem 5.2: The throughput efficiency η is given by the expression

$$\eta = \frac{1-P(D/D)}{1 + P(D) - P(D/D)} \tag{5.7}$$

Proof: The average number of transmissions per packet is

$$\overline{N} = \sum_{k=1}^{\infty} kP_{dep}(k) = \sum_{k=1}^{\infty} k[P_c(k) + P_e(k)]. \tag{5.8a}$$

Substituting from previous equations, the expression $\overline{N}$ becomes

$$\overline{N} = P(C)+P(E) + P(D)\cdot[P(C/D)+P(E/D)]\sum_{k=2}^{\infty} k[P(D/D)]^{k-2}. \tag{5.8b}$$

The geometric series can be summed to obtain

$$\overline{N} = P(C)+P(E) + P(D)\cdot[P(C/D)+P(E/D)]\frac{2-P(D/D)}{[1-P(D/D)]^2}. \tag{5.8c}$$

Noting that $P(E/D)+P(C/D) = 1-P(D/D)$, we get

$$\overline{N} = 1-P(D) + \frac{P(D)\cdot[2-P(D/D)]}{1-P(D/D)}. \tag{5.9}$$

This expression is further simplified to

$$\overline{N} = \frac{1+P(D)-P(D/D)}{1-P(D/D)} \tag{5.10}$$

The throughput efficiency is $\eta = 1/\overline{N}$ which becomes

$$\eta = \frac{1-P(D/D)}{1+P(D)-P(D/D)} \tag{5.11}$$

This ends the proof.

For the random-error channel, the conditional probability $P(D/D)$ is equal to $P(D)$, and the throughput efficiency would then be given by the simple expression $\eta = 1-P(D)$. This is the form usually quoted for the throughput efficiency of the selective repeat ARQ scheme, found, for example, in [6].

5.1.3 Summary

In this section, expressions have been derived for the performance parameters of the selective repeat ARQ scheme under the assumption of a noiseless feedback channel. These are now presented as a summary in Table 5.1.

Table 5.1 Performance Parameters of the Selective Repeat ARQ Scheme With Noiseless Feedback

Throughput Efficiency	$\eta = \dfrac{1-P(D/D)}{1+P(D)-P(D/D)}$
Prob. of Undetected error	$P_{UD} = P(E) + \dfrac{P(E/D)\cdot P(D)}{1-P(D/D)}$.
Prob. of Correct Delivery	$P_C = P(C) + \dfrac{P(C/D)\cdot P(D)}{1-P(D/D)}$.
Prob. of Packet Loss	$P_L = 0$
Prob. of Spurious Transmission	$P_{SP} = 0$

We now proceed to derive the expressions for the GH-II ARQ scheme when feedback errors are ignored.

5.2 GH-II ARQ Scheme with Noiseless Feedback

We rely heavily on the work already done in the preceding pages. Figure 5.1 has been modified to obtain Figure 5.2, in which we see that the depth changes from one transmission to the next. From this figure and earlier procedures, we obtain the probabilities of correct delivery and erroneous delivery at the kth transmission. These are given below.

$$P_c(1) = P(C_1), \qquad P_e(1) = P(E_1), \tag{5.12a}$$

$$P_c(2) = P(D_1)P(C_2/D_1), \qquad P_e(2) = P(D_1)P(E_2/D_1), \tag{5.12b}$$

$$P_c(3) = P(D_1)P(D_2/D_1)P(C_3/D_2). \qquad P_e(3) = P(D_1)P(D_2/D_1)P(E_3/D_2). \tag{5.12c}$$

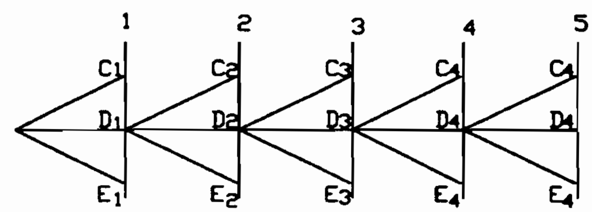

Figure 5.2 GH-II ARQ Scheme with Error-Free Feedback[3]

Continuing for $k \geq 4$, we obtain

$$P_c(4) = P(D_1)P(C_4/D_3)\prod_{j=1}^{2} P(D_{j+1}/D_j), \tag{5.12d}$$

$$P_c(5) = P(D_1)P(C_4/D_4)\prod_{j=1}^{3} P(D_{j+1}/D_j), \tag{5.12e}$$

$$P_e(4) = P(D_1)P(E_4/D_3)\prod_{j=1}^{2} P(D_{j+1}/D_j), \tag{5.12f}$$

$$P_e(5) = P(D_1)P(E_4/D_4)\prod_{j=1}^{3} P(D_{j+1}/D_j). \tag{5.12g}$$

In general, for $k \geq 5$, we obtain

$$P_c(k) = P(D_1)\cdot P(C_4/D_4)\cdot\left(\prod_{j=1}^{3} P(D_{j+1}/D_j)\right)\cdot(P(D_4/D_4))^{k-5}, \qquad k \geq 5, \text{ and} \tag{5.13a}$$

$$P_e(k) = P(D_1)\cdot P(E_4/D_4)\cdot\left(\prod_{j=1}^{3} P(D_{j+1}/D_j)\right)\cdot(P(D_4/D_4))^{k-5}, \qquad k \geq 5. \tag{5.13b}$$

3 Note that after four transmissions the labels in the Figure are the same; the reason is that once the maximum depth (depth-4) has been reached, there is no further depth-change.

5.2.3 Correct Delivery

By retracing the steps for finding P_{UD}, we find the probability of correct delivery as

$$P_C = \sum_{k=1}^{\infty} P_c(k) = \mu_{1C} + \frac{\mu_5}{1-\mu_3}, \text{ with} \tag{5.18}$$

$$\mu_{1C} = \sum_{k=1}^{4} P_c(k), \qquad \mu_5 = P(C_4/D_4) \cdot P(D_1) \cdot \prod_{j=1}^{3} P(D_{j+1}/D_j), \tag{5.19}$$

and μ_3 as before.

5.2.4 Summary

The key equations of this section number only three, since there are no packets lost, and there are no unnecessary transmissions either. The corresponding parameters are zero. The results are summarized in Table 5.2.

Table 5.2 Parameters of the GH-II ARQ Scheme With Noiseless Feedback

throughput	$\eta = \frac{(1-\mu_3)^2}{\mu_1(1-\mu_3)^2 + \mu_2(5-4\mu_3)}$
undetected error	$P_{UD} = \mu_{1E} + \frac{\mu_4}{1-\mu_3}$
correct delivery	$P_C = \mu_{1C} + \frac{\mu_5}{1-\mu_3}$
packet loss	$P_L = 0$
spurious delivery	$P_{SP} = 0$

5.2.5 Comparisons of Results for the GH-II ARQ Scheme

Simulations were run on a Markov channel with and without feedback errors. The results for a bit error rate (BER) of 10^{-4} are given in Table 5.3, from which it is clear that feedback noise lowers the throughput efficiency, increases the probability of undetected error, and creates the loss of packets and unnecessary transmissions of packets already delivered.

Table 5.3 Results For a Markov Channel with BER=10^{-4}, alpha=0.8, D_P=15[4]

		Noiseless Feedback	Noisy Feedback
throughput	η	0.85	0.70
undetected	P_{UD}	2.12×10^{-7}	9.31×10^{-7}
packet loss	P_L	—	2.11×10^{-8}
spurious delivery	P_{SP}	—	8.92×10^{-8}

4 For the meaning of α and D_P see Chapter VII

CHAPTER VI
PERFORMANCE ON THE GILBERT CHANNEL

The expressions for the performance of the GH-II ARQ and the Selective Repeat ARQ Scheme are given in this thesis with the assumption that the channel errors are Markov. Since the generating mechanism of the Gilbert channel is a Markov process, the impression created is that the resulting error process (from one transmitted block to the next) would be Markov. We show below that the number of errors in a packet, observed as the transmission proceeds, does not constitute a Markov chain. This fact means that the expressions derived so far cannot be used on the Gilbert channel. But, since the memory in the Gilbert channel decays rapidly, packets separated by at least one packet can be taken to be practically statistically independent. This is shown later. Therefore, the results of the Markov channel can be modified for use in this channel by disregarding the conditions in the conditional probabilities.

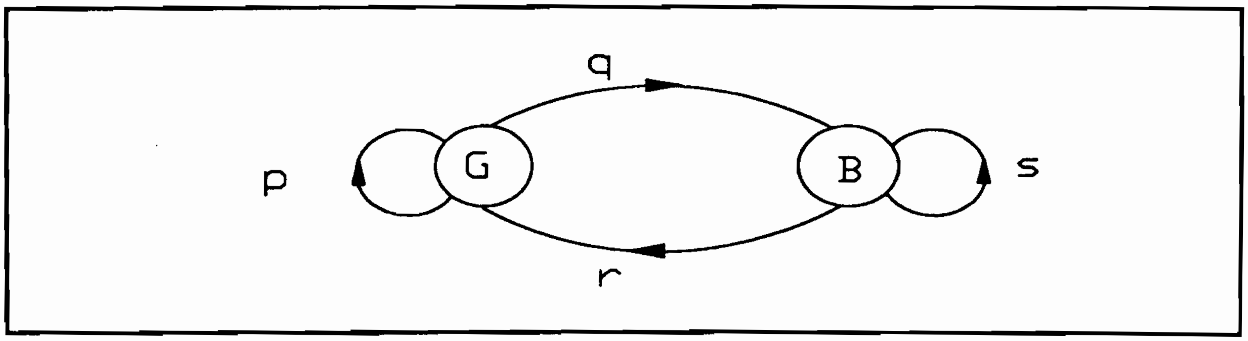

Figure 6.1 Gilbert's Model of a Burst Error Channel
G is the Good State and B is the Bad State

For a three-state model a very detailed analysis has been given by Dravida and others in [85], where the desired distributions are given in terms of generating functions. The corresponding probabilities can be found from these generating functions by an inversion process. For our purposes, we present an analysis of the two-state Gilbert channel. This error model is given by the state transition diagram of Figure 6.1. The state transitions are synchronized with the error digit transitions in the channel. It is taken that in the good state the output is a 0 (no error), and in the bad state the output is a 1 (an error). Therefore, as the chain makes transitions in {G,B}, the error digits make corresponding transitions in {0,1}.

6.1 Performance Evaluation

As shown in the next section, the amount of memory in the channel is practically limited to a block, a fact which means that the blocks can be taken as statistically independent. Within a block, the memory will create burst errors to occur. We assume that many of these errors will be corrected by the codes used. What remains then is the probability distribution of the number of errors in a block. Once this is known, the performance parameters can be determined. These are the topics of the following sections.

6.1.1 Extent of Memory in the Gilbert Channel

From Figure 6.1, the bit transitions are given by the matrix

$$\begin{array}{cc} & \begin{matrix} G & B \end{matrix} \\ \begin{matrix} G \\ B \end{matrix} & \begin{pmatrix} 1-q & q \\ r & 1-r \end{pmatrix}, \end{array} \tag{6.1}$$

where the transitions are taken to be from the state on the left of the matrix to the state above. It can be shown that

$$\begin{pmatrix} 1-q & q \\ r & 1-r \end{pmatrix}^k = \frac{1}{(q+r)}\begin{pmatrix} r & q \\ r & q \end{pmatrix} + \frac{(\gamma-1)^k}{(q+r)}\begin{pmatrix} q & -q \\ -r & r \end{pmatrix}, \tag{6.2}$$

where $\gamma = p+s = 2-(q+r)$, is called the *clustering coefficient* [40], as it measures the tendency of like symbols to follow each other. This parameter is very important in characterizing the Gilbert channel. We digress a little to discuss its importance. Let us consider γ-1. We notice that $\gamma-1 = p-r = s-q$. If $\gamma-1 > 0$, then $(0\rightarrow0)$ and $(1\rightarrow1)$ transitions are more frequent than $(1\rightarrow0)$ and $(0\rightarrow1)$ transitions. This condition is equivalent to $\gamma > 1$. For $\gamma < 1$, the situation is reversed; $(0\rightarrow1)$ and $(1\rightarrow0)$ transitions are more frequent. In this case, bursts are very rare. Therefore, to exhibit the bursty nature of the channel, we should have $\gamma > 1$. The simulations given later are run with $\gamma = 1.5$. The channel can be described by the average probability of bit error P_1 and the clustering coefficient γ. Given these two parameters, the other parameters of the model can be found since $P_1=q/(q+r)$ and $\gamma=2-(q+r)$. We have $q=(2-\gamma)P_1$, $p=1-q$; then, $s=\gamma-p$ and $r=1-s$.

In the matrix equation above, for $\gamma = 1.5$, we see that $0 < (\gamma\text{-}1)^k < 10^{-13}$ for $k > 40$. So beyond 40 bits, the statistical dependence is lost since

$$\begin{pmatrix} 1-q & q \\ r & 1-r \end{pmatrix}^k \approx \frac{1}{(q+r)}\begin{pmatrix} r & q \\ r & q \end{pmatrix} \tag{6.3a}$$

Since block lengths are typically 500 bits or more, it is safe to say that the memory in the channel does not extend beyond the block, except, of course, at the block edges. These end cases are ignored here because the transition probabilities of interest are those for events separated in time by at least half the round-trip delay, which is obviously larger than a packet slot. More precisely, the conditional probabilities of interest are $P(e_{n+D_p} \in S_K \mid e_n \in S_J)$. Since the events referred to are separated by several packet slots, we can write

$$P(e_{n+D_p} \in S_K \mid e_n \in S_J) \cong P(e_{n+D_p} \in S_K), \tag{6.3b}$$

showing that independence can be assumed.

6.1.2 The Number of Errors in a Block

In this section, an expression is derived for the probability distribution of the number of errors in a block of length N. Let us define a cycle as a sequence which begins with a 1 and ends when a 1 next occurs. Let C_n be the length of the nth cycle; then the event $\{C_n = 1\}$ is equivalent to the event $\{1|1\}$. In fact, $\{C_n = k\}$ is equivalent to $\{0^{k-1}1 \mid 1\}$, by which we mean the occurrence of k-1 0's in succession terminated by a 1, conditioned on there having been a 1 at the beginning.

Lemma 6.1: The cycles as defined above are statistically independent, that is

$$P(C_{n+1} = j \mid C_n = i) = P(C_{n+1} = j). \tag{6.4}$$

Proof: We obtain the following probabilities

$$P(C_n = 1) = s, \tag{6.5a}$$

$$P(C_n = 2) = qr, \tag{6.5b}$$

$$P(C_n = 3) = qrp, \tag{6.5c}$$

$$P(C_n = 4) = qrp^2, \tag{6.5d}$$

$$P(C_n=5) \;=\; qrp^3, \tag{6.5e}$$

$$P(C_n=6) \;=\; qrp^4. \tag{6.5f}$$

Continuing this, we find that in general,

$$P(C_n=m) \;=\; \begin{cases} 0 & m < 0 \\ s & m = 1 \\ qrp^{m-2} & m \geq 2. \end{cases} \tag{6.5g}$$

We now prove that the cycles are statistically independent.

$$P(C_{n+1}=j \mid C_n=i) \;=\; \frac{P(C_{n+1}=j, C_n=i)}{P(C_n=i)} \tag{6.6a}$$

$$= \frac{P([1(0)^{j-1}1(0)^{i-1}1])}{P([1(0)^{i-1}1])}, \tag{6.6b}$$

where, as before, we have used the notation $(0)^j$ to denote the occurrence of j uninterrupted successive 0's. Using the expressions just derived, we obtain

$$P(C_{n+1}=j \mid C_n=i) \;=\; \frac{(qs)p^{j-2}(qs)p^{i-2}}{qsp^{i-2}} \tag{6.7a}$$

$$= (qs)p^{j-2} = P(C_{n+1}=j). \tag{6.7b}$$

From the above, we have the result $P(C_{n+1}=j \mid C_n=i) = P(C_{n+1}=j)$, proving that successive cycles are statistically independent. In fact, all the cycles are statistically independent. This ends the proof.

We now proceed to find the probability distribution of a finite sum of cycles. We begin by noting that a sequence of weight k+1 has exactly k cycles. Let Y_k be the sum of these k cycles, i.e.,

$$Y_k \;=\; C_1 + C_2 + C_3 + \ldots + C_k. \tag{6.8}$$

Theorem 6.1: Given the events Y_k as defined above, let $P_k(m)$ be the probability of having k cycles whose length add up to m, and $Q_k(m)$ be the probability of having k zero-cycles whose lengths add up to m. Then

$$P_k(m) = \begin{cases} 0 & m < k , \\ s^k & m = k , \\ s^k \sum_{l=1}^{m-k} \binom{k}{l} \binom{m-k-1}{l-1} \left(\frac{qr}{ps}\right)^l & k < m \leq 2k , \\ s^k \sum_{l=1}^{k} \binom{k}{l} \binom{m-k-1}{l-1} \left(\frac{qr}{ps}\right)^l & m \geq 2k , \end{cases} \tag{6.9}$$

and

$$Q_k(m) = \begin{cases} 0 & m < k , \\ p^k & m = k , \\ p^k \sum_{l=1}^{m-k} \binom{k}{l} \binom{m-k-1}{l-1} \left(\frac{qr}{ps}\right)^l & k < m \leq 2k , \\ p^k \sum_{l=1}^{k} \binom{k}{l} \binom{m-k-1}{l-1} \left(\frac{qr}{ps}\right)^l & m \geq 2k . \end{cases} \tag{6.10}$$

__Proof__: Since the cycles are statistically independent, the moment generating function of Y_k is the product of the individual moment generating functions of the cycles. The cycles are identically distributed. So they have a common moment generating function, C(z), giving,

$$Y_k(z) \underline{\Delta} E\{z^{Y_k}\} \tag{6.11a}$$

$$= [C(z)]^k. \tag{6.11b}$$

so we must first find $C(z)$, which is defined as

$$C(z) \underline{\Delta} E\{z^C\} \tag{6.12}$$

Using the previous expressions, C(z) is found to be

$$C(z) = sz + qrz^2 \sum_{j=2}^{\infty} (pz)^{j-2}. \tag{6.13}$$

Evaluating the sum, we obtain

$$C(z) = sz + \left(\frac{qrz^2}{1-pz}\right), \tag{6.14}$$

which can be written as

$$C(z) = sz\left(1 + \frac{z(qr/s)}{1-pz}\right), \tag{6.15}$$

where the convergence of the sum is guaranteed for $|z| < 1/p$. Taking the kth power of $C(z)$, gives

$$[C(z)]^k = s^k z^k \left(1 + \frac{z(qr/s)}{1-pz}\right)^k = s^k z^k + \sum_{l=1}^{k} \binom{k}{l} \left(\frac{qr}{s}\right)^l \frac{z^l}{(1-pz)^l}. \tag{6.16}$$

Expanding the term $1/(1-pz)^l$, we obtain

$$[C(z)]^k = s^k z^k + s^k z^k \sum_{l=1}^{k} \binom{k}{l} \left(\frac{qr}{s}\right)^l z^l \sum_{u=0}^{\infty} \binom{l+u-1}{l-1} (pz)^u \tag{6.17a}$$

$$= s^k z^k + s^k z^k \sum_{l=1}^{k} \sum_{m=l}^{\infty} \binom{k}{l} \binom{m-1}{l-1} \left(\frac{qr}{ps}\right)^l z^m. \tag{6.17b}$$

The double sum can be broken into two parts to give

$$[C(z)]^k = s^k z^k + s^k z^k \sum_{l=1}^{k} \sum_{m=l}^{k} \binom{k}{l} \binom{m-1}{l-1} \left(\frac{qr}{ps}\right)^l z^m + s^k z^k \sum_{l=1}^{k} \sum_{m=k+1}^{\infty} \binom{k}{l} \binom{m-1}{l-1} \left(\frac{qr}{ps}\right)^l z^m. \tag{6.18}$$

We now change the order of summation in both sums, and then alter the indices to facilitate the recognition of the coefficient of z^m; the result is

$$[C(z)]^k = s^k z^k + s^k \sum_{m=k+1}^{2k} \sum_{l=1}^{m-k} \binom{k}{l} \binom{m-k-1}{l-1} \left(\frac{qr}{ps}\right)^l z^m$$

$$+ s^k \sum_{m=2k+1}^{\infty} \sum_{l=1}^{k} \binom{k}{l} \binom{m-k-1}{l-1} \left(\frac{qr}{ps}\right)^l z^m \tag{6.19a}$$

$$= \sum_{m=k}^{\infty} P_k(m) z^m. \tag{6.19b}$$

The first double sum contains powers of z from k+1 to 2k, and the second contains powers larger than 2k. It is also evident that powers of z less than k are absent. Using these observations, we obtain $P_k(m)$ as

$$P_k(m) = \begin{cases} 0 & m < k, \\ s^k & m = k, \\ s^k \sum_{l=1}^{m-k} \binom{k}{l} \binom{m-k-1}{l-1} \left(\frac{qr}{ps}\right)^l & k < m \le 2k, \\ s^k \sum_{l=1}^{k} \binom{k}{l} \binom{m-k-1}{l-1} \left(\frac{qr}{ps}\right)^l & m \ge 2k, \end{cases} \tag{6.20}$$

and $P_k(m)$ is the coefficient of the mth power of z in $\{ C(z) \}^k$ and is the probability of having k cycles whose lengths add up to m. It is also useful to know the distribution of zero-cycles. The probability that k zero-cycles have a total length equal to m is denoted by $Q_k(m)$. By replacing s and r by p and q, respectively, we can obtain the relevant results by similar reasoning. We have

$$Q_k(m) = \begin{cases} 0 & m < k, \\ p^k & m = k, \\ p^k \sum_{l=1}^{m-k} \binom{k}{l} \binom{m-k-1}{l-1} \left(\frac{qr}{ps}\right)^l & k < m \le 2k, \\ p^k \sum_{l=1}^{k} \binom{k}{l} \binom{m-k-1}{l-1} \left(\frac{qr}{ps}\right)^l & m \ge 2k. \end{cases} \tag{6.21}$$

This completes the proof.

Now we are ready to find an expression for the probability that at time n a sequence of length N has weight k. A 0 cycle is defined as a sequence that begins with a 0 and ends when a 0 next occurs. When a sequence begins with a 0 and ends with a 0, it can be viewed as consisting of 0-cycles. Such sequences that begin and end with a 0 have probability $P_0\, Q_{N-k-1}(N-1)$, by which we mean that after the first 0, the rest of the bits must be N-1 in number and the weight (the number of 1's) must be k. So the number of 0's must be N-k-1. A similar argument shows that sequences of weight k beginning and ending with a 1 have probability $P_1\, P_{k-1}(N-1)$, since after the first 1, the remaining digits must be N-1 in number and the number of 1's they contain must be k-1. Sequences beginning with a 0 and ending with a 1 will initially have 0-cycles (possibly), undergo a (0→1) transition somewhere (certainly), and end (possibly) with a string of 1's. The number of trailing 1's will vary from one sequence to another. If the last 0 occurs m positions from the very first 0, then the 0-cycles must have a total length of m. The remaining N-m-1 slots must be filled with 1's, which means there are N-m-1 (0→1) transitions. These will contribute a factor of s^{N-m-2} to the expression for the probability of these sequences. There is an initial (0→1) transition from the last 0 to the first 1. This transition contributes the factor q. Noting that the number of 0's, excluding the very first one, must be N-k-1, we must have $m=N-k-1$. Summing over all possible values of m, we get the probability of sequences beginning with a 0 and ending with a 1 as

$$P_0 \sum_{m=N-k-1}^{N-2} Q_{N-k-1}(m) q s^{N-m-2}. \tag{6.22}$$

The sum ends at N-2 because the first position has been taken by the beginning 0 and the last position must be a 1, leaving a maximum of N-2 for m. For sequences beginning with a 1 and ending with 0, we follow parallel paths in the reasoning and obtain their probability as

$$P_1 \sum_{m=k-1}^{N-2} P_{k-1}(m) r p^{N-m-2}. \tag{6.23}$$

Putting our results together, we obtain the following probability that a length-N sequence has k 1's:

$$\begin{aligned} P(e_n=k) &= P_1\left(\sum_{m=k-1}^{N-2} P_{k-1}(m) r p^{N-m-2} + P_{k-1}(N-1)\right) \\ &+ P_0\left(\sum_{m=N-k-1}^{N-2} Q_{N-k-1}(m) q s^{N-m-2} + Q_{N-k-1}(N-1)\right). \end{aligned} \tag{6.24}$$

If we know the error detecting power of the code, we can determine the probability of undetected error as

$$P_{UD} = \sum_{k=1+\varepsilon_{ED}}^{N} P(e_n=k), \tag{6.25}$$

where ε_{ED} is the maximum number of errors the code can detect.

Theorem 6.2: The sequence, $e_1, e_2, e_3, \ldots$, as defined above, is not first-order Markov.

Proof: We note that the process $\{X_n\}_{n=1}^{\infty}$ is Markov, with $X_n \in \{0,1\}$ and X_n is the nth error bit in the Gilbert channel. The transition probabilities are those of the Gilbert channel model. The number of errors in a packet of length N is a new process $\{e_n\}$ defined as follows:

$$e_1 = X_1 + X_2 + X_3 + \dots + X_N, \tag{6.26a}$$

$$e_2 = X_{N+1} + X_{N+2} + X_{N+3} + \dots + X_{2N}, \tag{6.26b}$$

.

$$e_k = X_{(k-1)N+1} + X_{(k-1)N+2} + X_{(k-1)N+3} + \dots + X_{kN}. \tag{6.26c}$$

Let $P(m,l,k) = P(e_{n+2}=m, e_{n+1}=l, e_n=k),$ (6.27a)

$$P(l,k) = P(e_{n+1}=l, e_n=k), \tag{6.27b}$$

$$P(k) = P(e_n=k), \tag{6.27c}$$

$$P(m,l \mid X_N=i) = P(e_{n+2}=m, e_{n+1}=l \mid X_N=i), \tag{6.27d}$$

$$P(l \mid X_N=i) = P(e_{n+1}=l, \mid X_N=i). \tag{6.27e}$$

Then

$$P(m,l,k)=P(m,l \mid X_N=0)\cdot P(e_1=k,X_N=0)+P(m,l \mid X_N=1)\cdot P(e_1=k,X_N=1) \tag{6.28a}$$

$$P(l,k) = P(l \mid X_N=0)\cdot P(e_1=k,X_N=0)+P(l \mid X_N=1)\cdot P(e_1=k,X_N=1) \tag{6.28b}$$

$$P(m \mid l,k) = \frac{P(m,l,k)}{P(l,k)} \tag{6.28c}$$

Substituting from the preceding expressions, we obtain

$$P(m \mid l,k)=\frac{P(m,l \mid X_N=0)P(e_1=k,X_N=0)+P(m,l \mid X_N=1)P(e_1=k,X_N=1)}{P(l \mid X_N=0)P(e_1=k,X_N=0)+P(l \mid X_N=1)P(e_1=k,X_N=1)} \tag{6.29a}$$

$$=\frac{P(m,l \mid X_N=0)+P(m,l \mid X_N=1)\left(\frac{P(e_1=k,X_N=1)}{P(e_1=k,X_N=0)}\right)}{P(l \mid X_N=0)+P(l \mid X_N=1)\left(\frac{P(e_1=k,X_N=1)}{P(e_1=k,X_N=0)}\right)} \tag{6.29b}$$

$$=\frac{P(m,l \mid X_N=0)+P(m,l \mid X_N=1)\rho_k(N)}{P(l \mid X_N=0)+P(l \mid X_N=1)\rho_k(N)}, \tag{6.29c}$$

where $\rho_k(N)$ is given by the ratio

$$\rho_k(N)=\frac{P(e_1=k,X_N=1)}{P(e_1=k,X_N=0)}. \tag{6.30}$$

If $\rho_k(N) = P_1/P_0$, then

$$P(m \mid l,k) \;=\; \frac{P(m,l \mid X_N=0)P_0 + P(m,l \mid X_N=1)P_1}{P(l \mid X_N=0)P_0 + P(l \mid X_N=1)P_1} = \frac{P(m,l)}{P(l)} = P(m \mid l). \tag{6.31}$$

The dependence on k would be lost. This condition is the minimum requirement for the sequence to be Markov. If we show that $\rho_k(N) \neq P_1/P_0$, then we will have shown that the process $\{ e_k \}$ cannot be Markov. For N=2, we consider k=1 and k=2.

$$k=1: \qquad \rho_k(N) \;=\; \frac{P(e_1=k,\;\; X_N=1)}{P(e_1=k,\;\; X_N=0)} \;=\; \frac{P(01)}{P(10)} \;=\; \frac{P_0q}{P_1r} \;=\; 1, \tag{6.32}$$

where we have used the fact that $P_0 = r/(q+r)$ and $P_1 = q/(q+r)$, giving $P_0 = (r/q)P_1$.

$$k=2: \qquad \rho_k(N) \;=\; \frac{P(11)}{P(\phi)}, \tag{6.33}$$

where (ϕ) denotes the null set. In this case, $\rho_k(N)$ is infinite, since the denominator is zero. These calculations, and others, are summarized in the following table.

$\rho_k(N) = \dfrac{P(e_1=k, X_1=1)}{P(e_1=k, X_N=0)}$		
k	*N=2*	*N=3*
0	$\rho_0(2) = \dfrac{P(\phi)}{P(00)} = 0$	$\rho_0(3) = \dfrac{P(\phi)}{P(000)} = 0$
1	$\rho_1(2) = \dfrac{P(01)}{P(10)} = \dfrac{P_0q}{P_1r} = 1$	$\rho_1(3) = \dfrac{P(001)}{P(010 \cup 011)} = \dfrac{P_0pq}{P_0qr + P_1rp}$
2	$\rho_2(2) = \dfrac{P(11)}{P(\phi)}$, infinite	$\rho_2(3) = \dfrac{P(101 \cup 011)}{P(110)} = \dfrac{P_1rq + P_0qs}{P_1sr}$
3	——	$\rho_3(3) = \dfrac{P(111)}{P(\phi)}$, infinite

For N=4, it is easily verified that $\rho_k(N) = 0$ for k=0, and $\rho_k(N)$ is infinite for k=4. Since $\rho_k(N)$ is seen to vary with the value of k, it cannot equal the ratio P_1/P_0; thus, the process $\{ e_k \}$ cannot be first-order Markov. This ends the proof.

The consequence of this is that the expressions derived earlier for performance parameters cannot be used directly to establish performance on the Gilbert channel. From the remark made earlier about the extent of memory in the Gilbert channel, however, the conditional probabilities in the Markov channel can be replaced with unconditional probabilities. Once this is done, a parallel set of expressions can be found for the present channel. Realistic conclusions can then be drawn from the simulation results provided here.

CHAPTER VII
SIMULATIONS

The simulations described here are presented first for the Gilbert channel and then presented for the Markov channel. Some of the results for the Gilbert channel are given here; the rest are given later. For the Markov channel, a preliminary discussion is provided to show how the channel is modelled. The model is then used to show how the round-trip delay D_P enters into the simulations.

7.1 Simulations on The Gilbert Channel

The parameters of the channel were varied to yield different bit error rates. For each set of parameter values, the throughput efficiency of the GH-II ARQ scheme was determined by simulation. Also found were the corresponding values for the probabilities of loss and undetected error. The following figures are a record of the results. For the Gilbert Channel, the parameter γ was set to 1.5, and the probability of bit error was varied.

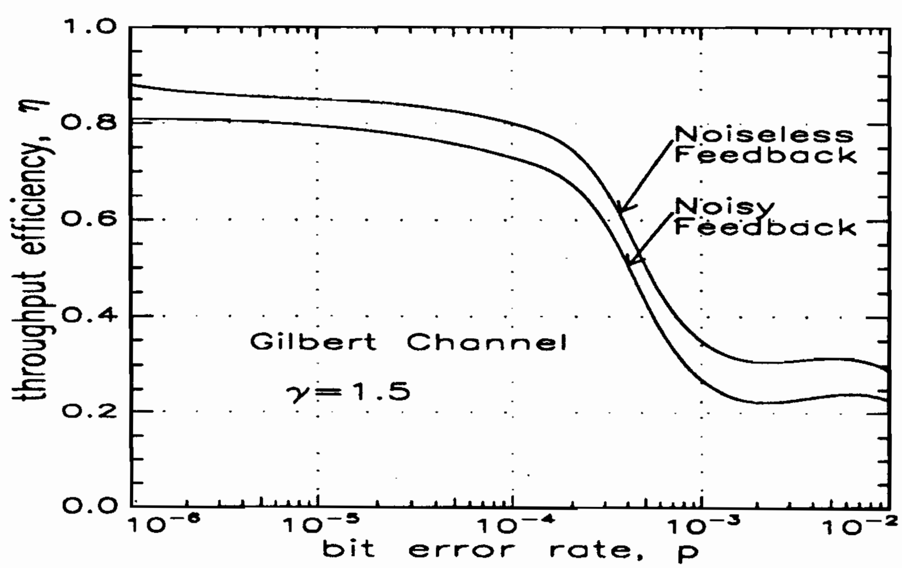

Figure 7.1 Throughput of the GH-II ARQ on the Gilbert Channel

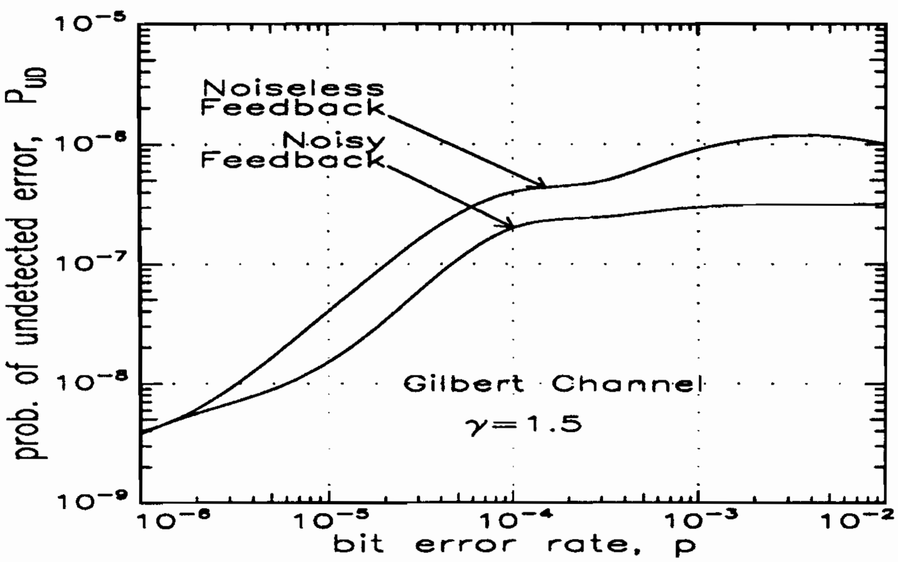

Figure 7.2 Probability of Undetected Error on the Gilbert Channel

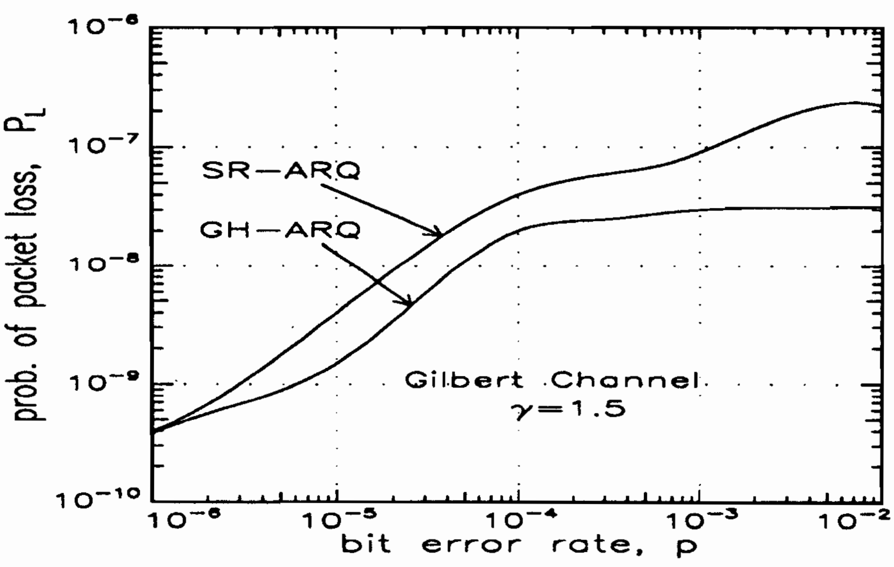

Figure 7.3 Comparison of Packet Loss Probabilities on the Gilbert Channel

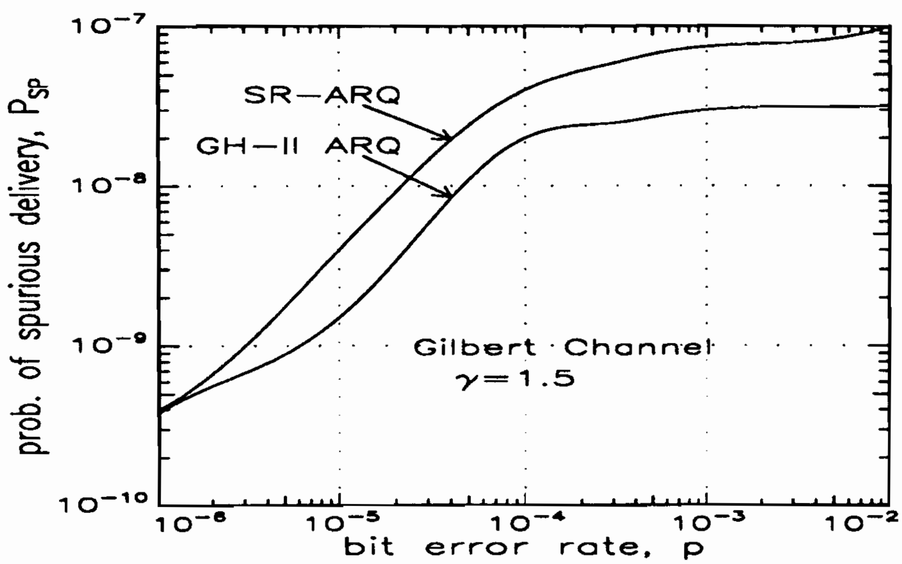

Figure 7.4 Probability of Spurious Delivery on the Gilbert Channel

It is evident from these results that the throughput efficiency is lower when feedback errors are included. Also more undetected errors occur. Further, the loss of packets cannot be ignored. The superiority of the GH-II ARQ scheme over the SR-ARQ scheme is seen in its lower probabilities of packet loss and spurious delivery.

7.2 Simulations on The Markov Channel

To describe the method used for simulating the Markov Channel, we will first construct an appropriate error process. After the error process has been described, a method of generating an error pattern of a given length and weight is given. The process and the method are combined to generate the results given later.

7.2.1 Construction of a First-Order Markov Error Process

Let $e_n = \alpha e_{n-1} + (1-\alpha)u_n$, where $\{ u_n \}$ are independent identically distributed random variables, with mean m and variance σ_u^2. The interval of definition for u_n must be set such that for each n, the interval of e_n is not changed. That is, if N is the maximum value of e_n and the minimum is zero, then we must have

$$\min\{u_n\} = 0, \quad \text{and} \tag{7.1a}$$

$$\max\{u_n\} = N. \tag{7.1b}$$

From this, e_n can expressed as

$$e_n = \alpha^n e_0 + (1-\alpha)\sum_{i=1}^{n} \alpha^{n-i} u_i, \tag{7.2}$$

where each u_i is independent of e_0. For $k < l$, e_l can be written as

$$e_l = \alpha^{l-k} e_k + (1-\alpha)\sum_{i=1}^{l-k} \alpha^{l-k-i} u_{i+k}. \tag{7.3}$$

From (7.2), it follows that

$$\overline{e_n} = \alpha^n \overline{e_0} + (1-\alpha)\sum_{i=1}^{n} \alpha^{n-i} \overline{u_i}, \tag{7.4}$$

It is assumed that the error process e_n has a constant mean. Letting μ be the mean, the above equation yields

$$\mu = \alpha^n \mu + (1-\alpha)\frac{m\alpha^{n-1}}{1-\alpha^{-1}}(1-\alpha^{-n}). \tag{7.5}$$

Solving this for μ, gives $\mu = m$. The variance σ^2 of e_n is

$$\overline{e_n^2} - \mu^2 = \alpha^{2n}\ \overline{e_0^2} + 2(1-\alpha)\alpha^{2n} m\mu \sum_{i=1}^{n} \alpha^{-i}$$

$$+ (1-\alpha)^2\alpha^{2n} \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha^{-i-j}\ \overline{u_i u_j} - \alpha^{2n}\left(\overline{e_0} + (1-\alpha)m \sum_{i=1}^{n} \alpha^{-i}\right)^2. \tag{7.6}$$

The double sum is evaluated by first isolating the n terms for which $i=j$ and taking the remaining $n(n-1)$ terms together. The following intermediate result is obtained after cancelling out terms:

$$\overline{e_n^2} - \mu^2 = \alpha^{2n}(\overline{e_0^2} - \mu^2) + (1-\alpha)^2\alpha^{2n} \sum_{i=1}^{n} \alpha^{-2i}(\overline{u^2} - m^2). \tag{7.7}$$

Assuming that the variance is independent of time for the error process, the expression for the variance becomes

$$\sigma^2(1-\alpha^{2n}) \;=\; \frac{\sigma_u^2(1-\alpha)^2}{\alpha^2-1}(\alpha^{2n}-1). \tag{7.8}$$

which finally gives

$$\sigma^2 \;=\; \sigma_u^2\left(\frac{1-\alpha}{1+\alpha}\right). \tag{7.9}$$

<u>Autocorrelation Function:</u> The autocorrelation function $r_{k,l}$ of the process is

$$r_{k,l} \;=\; E\{e_l e_k\} \;=\; E\left\{\alpha^{l-k}e_k^2 \;+\; e_k(1-\alpha)\sum_{i=1}^{l-k}\alpha^{l-k-i}u_{i+k}\right\}. \tag{7.10}$$

Using the independence of the u_i's results in

$$r_{k,l} \;=\; \alpha^{l-k}\;\overline{e_k^2} \;+\; \mu m(1-\alpha)\sum_{i=1}^{l-k}\alpha^{l-k-i} \tag{7.11}$$

Evaluating the sum and rearranging gives,

$$r_{l,k} \;=\; \alpha^{l-k}\;\overline{e_k^2} \;+\; \mu m(1-\alpha)\left(\frac{1-\alpha^{l-k}}{1-\alpha}\right). \tag{7.12}$$

Using the relationship between μ and m found earlier in (7.5), the autocorrelation is

$$r_{l,k} \;=\; \alpha^{l-k}\;\sigma^2 \;+\; \mu^2. \;=\; \left(\frac{1-\alpha}{1+\alpha}\right)\alpha^{l-k}\sigma_u^2 \;+\; \mu^2. \tag{7.13}$$

This result has been obtained under the assumption that $k < l$, from which it is seen that if $k > l$, the roles will be reversed; therefore,

$$r_{l,k} \;=\; r_{|l-k|}. \tag{7.14}$$

The autocovariance function $c_{k,l}$ of the process is seen to be

$$c_{k,l} = \sigma^2\alpha^{|k-l|}. \tag{7.15}$$

The random variables $\{u_n\}$ are chosen to be binomial with parameters N and p, where p is the probability of error. The mean of u_n is therefore Np, and the variance is $Np(1-p)$. The mean and variance of the constructed process are, therefore, given by

$$\mu = m = Np \tag{7.16}$$

$$\sigma^2 = \sigma_u^2\left(\frac{1-\alpha}{1+\alpha}\right) = Np(1-p)\left(\frac{1-\alpha}{1+\alpha}\right). \tag{7.17}$$

Thus, with the indicated assumptions (constant mean and constant variance) we have constructed a weakly stationary process $\{e_n\}$. The errors that occur in many channels can be adequately represented by a decaying autocovariance function of the type found here [54]. This error model is used in the simulations with the parameters $\alpha = 0.8$ and $N = 504$. The quantity p is varied with each u_n being a binomial random variable with parameters 504 and p.

7.2.2 Calculating the Conditional Probabilities

Every event referred to here occurs when the number of errors lies in a particular interval. Let $S_Q = \{\text{error interval corresponding to event Q}\}$. With the error model given by $e_n = \alpha e_{n-1} + (1-\alpha)u_n$, where the u_n's are independent and identically distributed random variables, the conditional probability P(T|D$_1$) used in the GH-II ARQ scheme can be found as follows. First, we define the random variable $W_{DP}(\alpha)$ as

$$W_{DP}(\alpha) = (1-\alpha)\sum_{m=1}^{D_P}\alpha^{D_P-m}u_{n+m}. \tag{7.18}$$

From (7.3) we can write

$$e_{n+D_P} = \alpha^{D_P}e_n + (1-\alpha)\sum_{i=1}^{D_P}\alpha^{D_P-i}u_{n+i} \tag{7.19a}$$

$$= \alpha^{D_P}e_n + W_{DP}(\alpha). \tag{7.19b}$$

Then the conditional probability P(T|D$_1$) is given by

$$P(T\,|\,D_1) = P(e_{n+D_P}\in S_T\,|\,e_n\in S_{D_1}) = \sum_{j\in S_{D_1}}\ \sum_{x\in S_T-j\alpha^{D_P}} P(W_{DP}(\alpha)=x), \tag{7.20}$$

where $S_T - j\alpha^{D_P}$ is a translation of the interval S_T. Similar expressions can be found for the remaining conditional probabilities. The random variable $W_{DP}(\alpha)$ is not necessarily an integer; it is, however, discrete.

7.2.3 Generating an Error Pattern of a Given Weight and Length

Using the process constructed above, we first note that successive transmission times of blocks that belong to the same packet are separated by D_P packet times. So, we are interested in generating e_{n+D_p} once we have e_n. From (7.3), we have

$$e_{n+D_p} = \alpha^{D_p} e_n + (1-\alpha) \sum_{i=1}^{D_p} \alpha^{D_p - i} u_{n+i}. \tag{7.21}$$

Let WT be the integer closest to e_{n+D_p}. Then we wish to generate an error pattern of weight WT and of length N. The algorithm for accomplishing this is given in the pseudocode in Figure 7.5. From this procedure, the vector E, of dimension N, is the error pattern of the required weight. This vector is added modulo-2 to the transmitted block in the simulations to represent the effect of the noise in the channel. The bit error rate is given by the parameter p in the binomial random variable u_n. By setting the round-trip delay to 15 (D_P=15), and varying the bit error rate p, several results were found. The situation for $p=10^{-4}$ and $\alpha=0.8$ is summarized in Table 7.1, which is a reproduction of Table 5.3. The rest of the results are given later in Figures 8.9-8.15.

```
1 N1=N
  P(i)=i, i=1,2,...,N1
  If WT=0 goto step 5
  If WT=N1 then
    {
    E(i)=1,  i=1,2,...,N1
    exit
    }
   count=0
2 If count=WT goto 4
3 generate a random number in the closed interval [1,N1]
  let j = the integer nearest to this number.
  E(P(j)) = 1
  count=count+1
  If j < N1  then  P(i)=P(i+1),  i=j,j+1,j+2,...,N1-1
  N1=N1-1
  goto step 2
4 If N1=0 exit
5 E(P(i))=0  i=1,2,3,...,N1
  exit
```

Figure 7.5 Generating an Error Pattern of a Given Weight and Length

Table 7.1 Simulation Results For a Markov Channel with BER=10^{-4}, D_P=15, α=0.8

	Noiseless Feedback	Noisy Feedback
throughput η	0.85	0.70
undetected P_{UD}	2.12×10^{-7}	9.31×10^{-7}
packet loss P_L	—	2.11×10^{-8}
spurious delivery P_{SP}	—	8.92×10^{-8}

CHAPTER VIII
IMPROVEMENTS

In this chapter, we consider some ways of improving the performance of ARQ schemes which take into account the errors in the return channel. The methods discussed are later used in simulations to determine how much improvement they provide.

8.1 ARQ Scheme with a Predictor

Since the channel noise level varies with time, it is desirable to have an idea of what it is going to be in the next transmission slot; this would facilitate choosing the appropriate coding parameters. Suppose that for each decoding, the decoder produces the weight of the corresponding error pattern. These weights could then be fed into a predictor to produce an estimate of the noise level in the channel for the transmission slot. This estimate can be sent to the transmitter, where it can be used to determine the starting depth for the next packet to be transmitted. This idea is illustrated in Figure 8.1. The output marked E is the number of errors in the received block.

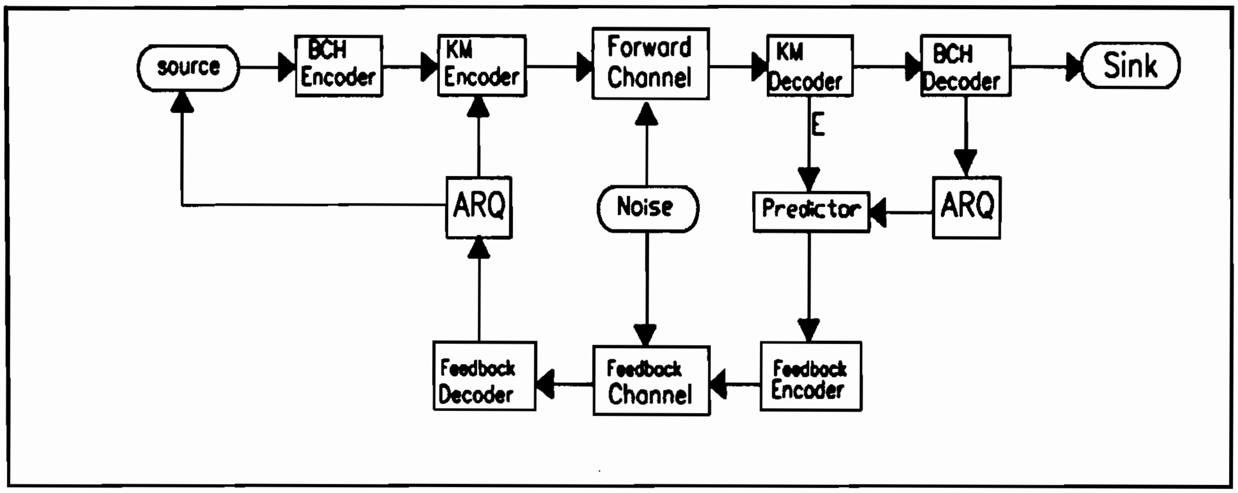

Figure 8.1: ARQ with a Predictor at the Receiver

Suppose the transmission time for a packet is T_t seconds, and the propagation delay is T_P seconds, with $T_P = D_P T_t$. The GH-II ARQ scheme always starts the packet transmission at depth-1. If the channel is in a state suitable for depth-l, then the scheme will require $T_l = l(T_P + T_t) = l(D_P + 1)T_t$ seconds to transmit a packet. With a predictor, all the l blocks will be transmitted together, giving a total time of $T_2 = T_P + T_t l = (D_P + l)T_t$ seconds. The ratio of the throughputs is $T_1/T_2 = l(D_P + 1)/(D_P + l)$. For $D_P \gg 1$, $T_1/T_2 \approx l$. This shows that for large round-trip delay, the throughput improvement due to the predictor is approximately equal to the actual *depth* that is appropriate for the channel. Table 8.1 gives the ratios for some values of D_P and l.

Table 8.1 Throughput Ratio $\left[\frac{\eta_{prd}}{\eta} = \frac{l(D_P+1)}{D_P+l}\right]$ for Various Round-trip Delays[1]

	l=1	l=2	l=3	l=4
D_P=15	1	1.88	2.67	3.37
D_P=20	1	1.91	2.74	3.50
D_P=30	1	1.94	2.82	3.65
D_P=40	1	1.95	2.86	3.74

Another advantage of using a predictor is that while with the conventional method time is spent decoding each depth (starting with the lowest) until the final depth, with the predictor only one decoding is done. In this way, all the intermediate decodings are eliminated.

8.2 Predictor at the Receiver

One problem with the arrangement of Figure 8.1 is obvious: there may be a long travel time delay between the transmitter and the receiver, so that by the time the noise estimate arrives at the transmitter it is obsolete. Besides, the noise in the return channel can also alter the value of the estimate so as to render it misleading. So, although at a first glance, the approach appears to have some merit, the arrangement has some obvious shortcomings. We now consider a method that circumvents some of these difficulties.

8.3 Predictor at the Transmitter

When the main traffic is from the transmitter to the receiver, the return traffic, being much lighter, is used for indicating the status of the channel and the status of previous transmissions. We propose to place the predictor at the transmitter. In this way, the predictions of the channel noise level do

1 The quantity η_{prd} is the throughput efficiency with prediction, and η is the throughput efficiency without prediction.

not suffer from the long travel time delay mentioned above. Besides, the predictions do not incur any corruption by the noise in the channel, as was the case in the above arrangement. Figure 8.2 illustrates the procedure.

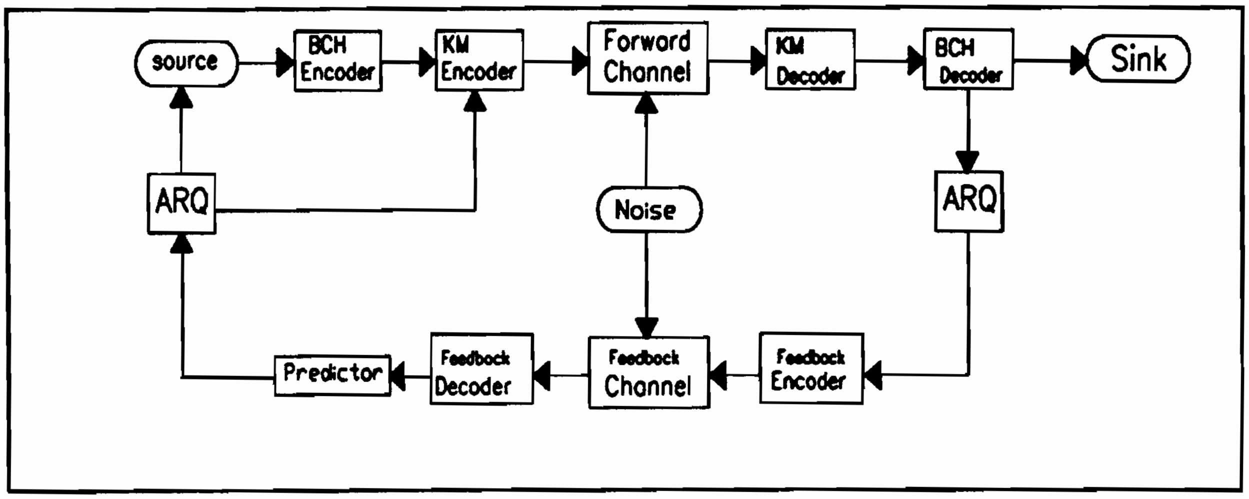

Figure 8.2: ARQ with a Predictor at the Transmitter

Now the ACK/NACK message can be encoded into a relatively short codeword (28 bits long, say). The predictor then takes the number of errors (available from the decoder) and combines this with the previous values to produce an estimate $\hat{e}_n$ for the number of errors in the next block. This expanded ACK/NACK message tells the source whether or not to supply a new packet. If necessary, the length of the feedback code can be increased so as to detect more errors.

8.4 The Predictor

The prediction of a parameter from the observations of a random process is a topic that has received considerable attention in engineering and science; it has even been given attention in such areas as economics and the social sciences. It is not the intention of the present work to duplicate any results that have been obtained on this topic. The aim here is to demonstrate that prediction does provide some performance improvement in the GH-II ARQ scheme.

We propose to use the predictor with the GH-II ARQ scheme, discussed in Chapter II, as follows. When a request is made for a retransmission, GH-II ARQ is used. When a delivery is made, the predictions are used to determine the starting depth. Once the depth is set, the original GH-II ARQ scheme takes over. The idea behind this is that sometimes the number of errors in the channel may suit a depth other than depth-1. The predictor will hit the proper starting depth (hopefully) most of the time. This will certainly increase the reliability of the scheme. It must be noted that it necessarily increases the complexity of the scheme. If the error-detecting code C_{ED} is powerful enough, then adding a predictor will also increase the throughput efficiency.

The time difference between the receiver and the transmitter is D_P packet times. First estimate the mean $\hat{\mu}$ of the error process, as

$$\hat{\mu} = \frac{1}{M}\sum_{i=1}^{M} e_{n-i} \tag{8.1}$$

Suppose the predictor is of order P and is linear. Then, the prediction $\hat{e}_{n-1+D_p}$ of e_{n-1+D_p}, given e_{n-1} and all the preceding P-1 values, is given by

$$\hat{e}_{n-1+D_p} = \hat{\mu} + \sum_{i=1}^{P} a_i(e_{n-i} - \hat{\mu}). \tag{8.2}$$

The predictor coefficients a_i, $i=1,2,\ldots,P$, are then determined by minimizing the mean square error $\overline{\varepsilon^2}$ of the predictor. The mean square error is given by

$$\overline{\varepsilon^2} = E(e_{n-1+D_p} - \hat{e}_{n-1+D_p})^2 = E\left(e_{n-1+D_p} - \hat{\mu} - \sum_{i=1}^{P} a_i(e_{n-i} - \hat{\mu})\right)^2 \tag{8.3a}$$

$$= E\{(e_{n-1+D_p} - \hat{\mu})^2\} + \sum_{i=1}^{P}\sum_{j=1}^{P} a_i E\{(e_{n-i} - \hat{\mu})(e_{n-j} - \hat{\mu})\} a_j$$

$$- 2\sum_{i=1}^{P} a_i E\{(e_{n-i} - \hat{\mu})(e_{n-1+D_p} - \hat{\mu})\} \tag{8.3b}$$

$$= c_0 + \sum_{i=1}^{P}\sum_{j=1}^{P} a_i c_{i-j} a_j - 2\sum_{i=1}^{P} a_i c_{D_p-i-1} \tag{8.3c}$$

where $c_{i-j} = \sigma^2\alpha^{|i-j|}$ as found earlier. Optimizing with respect to each a_l, gives

$$\sum_{i=1}^{P} a_i c_{i-l} = c_{D_p-1-l} \qquad l = 1,2,\ldots,P, \tag{8.4}$$

which can be written as

$$\sum_{i=1}^{P} a_i \alpha^{|i-l|} = \alpha^{D_p+l-1} \qquad l = 1,2,\ldots,P. \tag{8.5}$$

This set of equations can be solved to find the predictor coefficients a_i, $i=1,2,\ldots,P$. In matrix-vector form, the equations become

$$\begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{P-1} \\ \alpha & 1 & \alpha & \cdots & \alpha^{P-2} \\ \alpha^2 & \alpha & 1 & \cdots & \alpha^{P-3} \\ \vdots & \vdots & \vdots & & \vdots \\ \alpha^{P-1} & \alpha^{P-2} & \alpha^{P-3} & \cdots & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_P \end{pmatrix} = \begin{pmatrix} 1 \\ \alpha \\ \alpha^2 \\ \vdots \\ \alpha^{P-1} \end{pmatrix} \alpha^{D_P} \tag{8.6}$$

Since the matrix is Toeplitz, the system of equations can be solved using the Levinson-Durbin algorithm [89-92] to obtain the predictor coefficients. The prediction $\hat{e}_{n-1+D_P}$ of e_{n-1+D_P} is then given by

$$\hat{e}_{n-1+D_P} = \hat{\mu} + \sum_{i=1}^{P} a_i(e_{n-i} - \hat{\mu}), \tag{8.7}$$

With the example process given above, the solution is found to be quite simple. We have that

$$a_1 = \alpha^{D_P} \tag{8.8}$$

$$a_i = 0 \qquad \text{for} \quad i \geq 2, \tag{8.9}$$

resulting in the prediction,

$$\hat{e}_{n-1+D_P} = \hat{\mu} + \alpha^{D_P} e_{n-1}, \tag{8.10}$$

with mean square error

$$\overline{\varepsilon^2} = E\left\{\left[(e_{n-1+D_P} - \hat{\mu}) - \alpha^{D_P}(e_{n-1} - \hat{\mu})\right]^2\right\} \tag{8.11a}$$

$$= E\left[(e_{n-1+D_P} - \hat{\mu})^2\right] + \alpha^{2D_P} E[(e_{n-1} - \hat{\mu})^2]$$

$$- 2\alpha^{D_P} E[(e_{n-1+D_P} - \hat{\mu})(e_{n-1} - \hat{\mu})] \tag{8.11b}$$

$$= c_0 + \alpha^{2D_P} c_0 - 2\alpha^{D_P} c_{D_P}. \tag{8.11c}$$

The mean square error simplifies further to

$$\overline{\varepsilon^2} = \sigma^2\left(1-\alpha^{2D_p}\right). \tag{8.12}$$

In applications, the values of the autocorrelation function are not known and have to be estimated from past observations. One way [89,90] to estimate these is

$$\hat{r}_k(n) = \left(\frac{1}{M-k}\right)\sum_{j=1}^{M-k} e_{n-j}e_{n-j-k}, \qquad k = 0,1,2,\ldots,P. \tag{8.13}$$

Then $\hat{c}_k = \hat{r}_k - (\hat{\mu})^2$.

It is evident that, for $0<\alpha<1$, the mean squared error increases as the delay (D_P) increases. The effect of this delay on the performance of the predictor was studied and was found to exhibit the trend shown in Figure 8.3 for M=40, BER=10^{-4}, α=0.8, and a packet size of 504 bits. These results were also found to depend on the choice of the parameter M used in estimating the mean, as shown in Table 8.2.

Table 8.2 Effect of M on Prediction Error

M	10	20	30	40	50
$\overline{\varepsilon^2}/r_0$	0.092	0.051	0.032	0.011	0.006

A value of 40 was used for M in the simulations that are reported here. Two cases were considered—one with a bit error rate (BER) of 10^{-4} and varying the round trip delay (D_P), and the other with D_P=15 and varying BER. The results are reported in Figures 8.10, 8.12, 8.14, and 8.15. Table 8.3 gives the results when both parameters are fixed (D_P=15 and BER=10^{-4}).

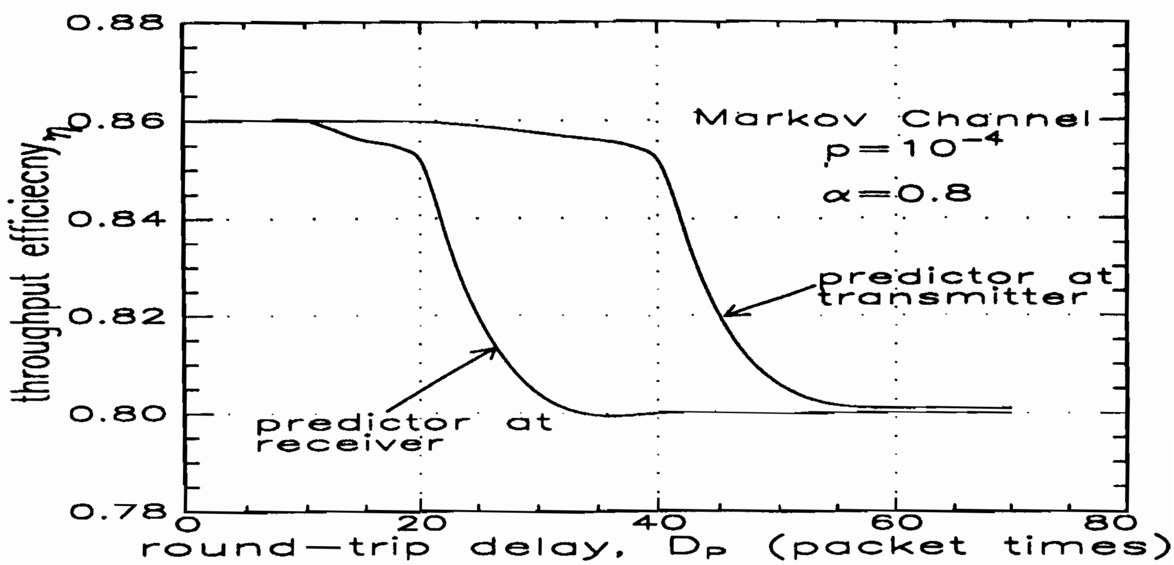

Figure 8.3 Effect of Round-Trip Delay on Throughput

Table 8.3 Effect of Prediction on GH-II ARQ Performance For BER=10^{-4}, D_P=15, α=0.8			
	Noisy Feedback Without Prediction	Noisy Feedback With Prediction	Noiseless Feedback Without Prediction
throughput efficiency η	0.70	0.80	0.85
undetected error prob. P_{UD}	9.31×10^{-7}	3.22×10^{-7}	2.12×10^{-7}
packet loss prob. P_L	2.11×10^{-8}	5.10×10^{-9}	—
spurious delivery prob. P_{SP}	8.92×10^{-8}	2.01×10^{-8}	—

We see from Table 8.3 that the predictor has increased the throughput efficiency, lowered the number of lost packets, and reduced the probability of undetected error.

8.5 The Feedback Encoder with a Threshold

At the receiver, an all-1 sequence is used for a NACK, and an all-0 sequence is used to indicate an ACK. At the transmitter, the number of 0's is found, and denoted by W_n. The decision rule regarding the status of the previously sent packet is

send a new packet if $W_n > \theta$

retransmit the packet if $W_n \leq \theta$,

where θ is some preselected threshold. It is interesting to observe (as is demonstrated later) that some performance parameters, such as the throughput efficiency and the probability of loss, are sensitive to this threshold. Incidentally, this type of decoding can be traced to Schwartz, who in [90] mentions this idea just briefly. Let us consider the situation where we interpret the return message to mean "ACCEPTED" only when the message is all 0's, i.e., $\theta = 0$. Let us also define a false start to be the transmission of a new packet when, in fact, a retransmission is required. Then in this case, the probability of false start has been greatly reduced. The problem is that the throughput has also been reduced, and unnecessary transmissions are quite frequent. By allowing some margin (increasing the threshold), we obtain a higher throughput. It is then seen that there is a trade-off between reducing the probability of a false start (which leads to the loss of packets) and having a reasonably high throughput. The effect on the probability of loss is illustrated in Figure 8.4 and the effect on the throughput efficiency is shown in Figure 8.5

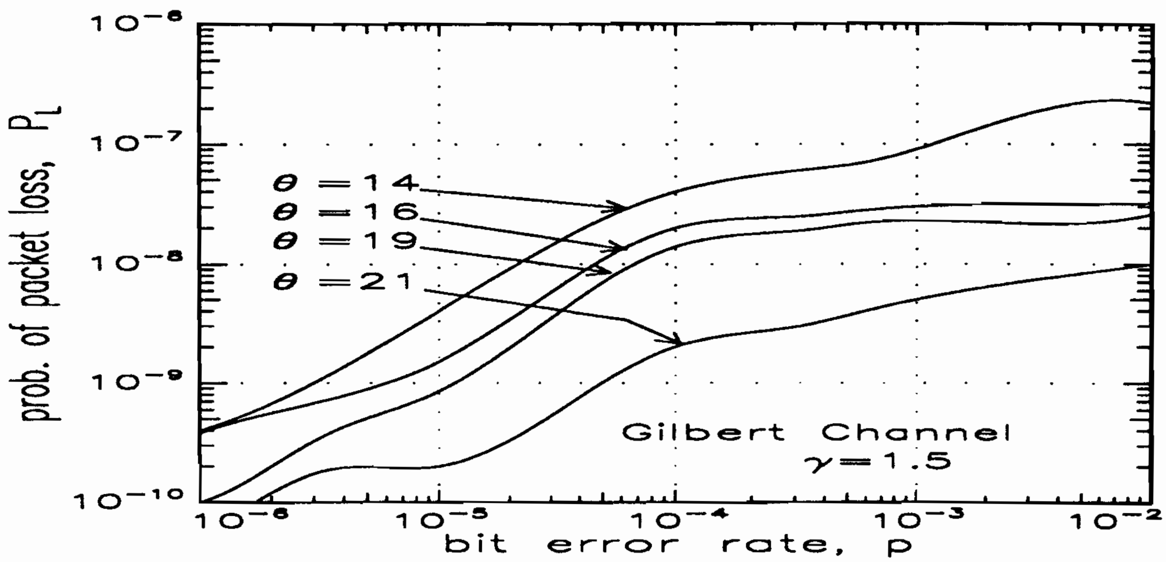

Figure 8.4 Effect of Threshold on the Probability of Loss

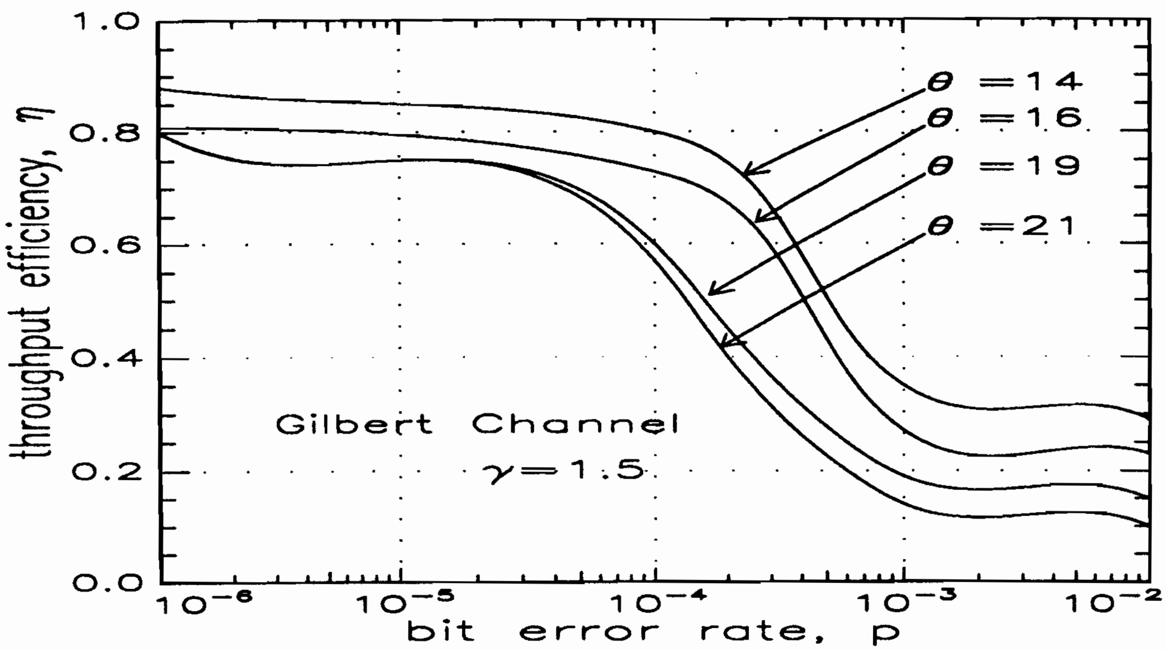

Figure 8.5 Effect of Threshold on the throughput Efficiency

8.6 Using an Invertible Code to Avoid Packet Loss

In variable redundancy transmission, it is possible to virtually eliminate the loss of packets. At the price of some added complexity, the following procedure could be adopted. Suppose the block received is divided into sub-blocks, so that

$$\mathbf{B}^{(k)} = [\mathbf{b}_1^{(k)}, \ldots, \mathbf{b}_L^{(k)}], \tag{8.14}$$

where each $b_j^{(k)}$ is exactly k_l bits long, with k_l = N/L, and N is the length of the block $\mathbf{B}^{(k)}$. In the simulations N = 504 and L=72, giving k_l = 7. If the block $\mathbf{B}^{(1)}$ is the first block of a packet in an ARQ scheme that uses a code with invertible sub-blocks, then each $b_j^{(1)}$ can be inverted to form the vector

$$\mathbf{C} \;=\; [\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_L]. \tag{8.15}$$

The vector **C** is then checked for errors. If errors are found, **C** can be used at the receiver to determine the next block to be received. Let us call this $\hat{\mathbf{B}}^{(2)}$. The received block is compared with this, and, if found to differ beyond some acceptable threshold, it is considered to belong to a different packet, and is rejected; otherwise, it is taken to belong to the same packet and is combined with the previous one for decoding. If it is rejected, then a request is made for the retransmission of the previous packet. The procedure is illustrated in the flowchart of Figure 8.6.

It is crucial to note that the procedure is possible only for a code with a generator matrix whose blocks are invertible. The invertibility of the first block is assumed by the box labelled "*decode* $\mathbf{B}^{(k)}$". If the transmission is other than the first, it can be combined with the previous ones for decoding. The result of this decoding is then used in determining an estimate of the expected block; thus, only the first block needs to be invertible. Invertibility is a property of many KM codes, a fact which creates room for additional strength in the GH-II ARQ scheme that uses these codes. The results of this method are shown in Figure 8.8. As was to be expected, the throughput efficiency is reduced by using this scheme. The effect on the loss of packets is shown in Figure 8.9, from which we see that the loss of packets has been tremendously reduced. Sometimes it is important that no message be lost. In such a case, a lower throughput may be tolerated in order to have a lower probability of packet loss.

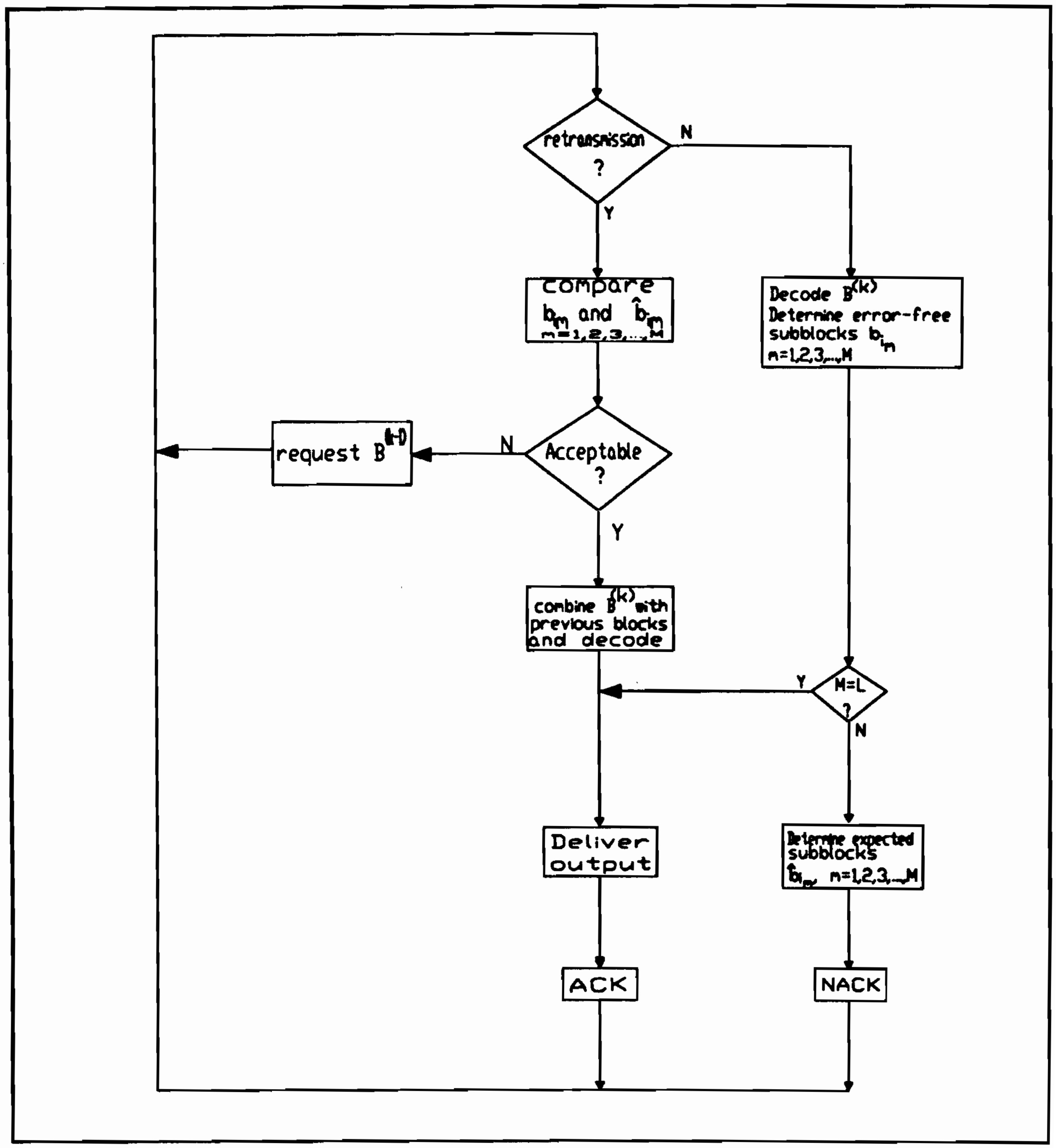

Figure 8.6 Avoiding the Loss of Packets

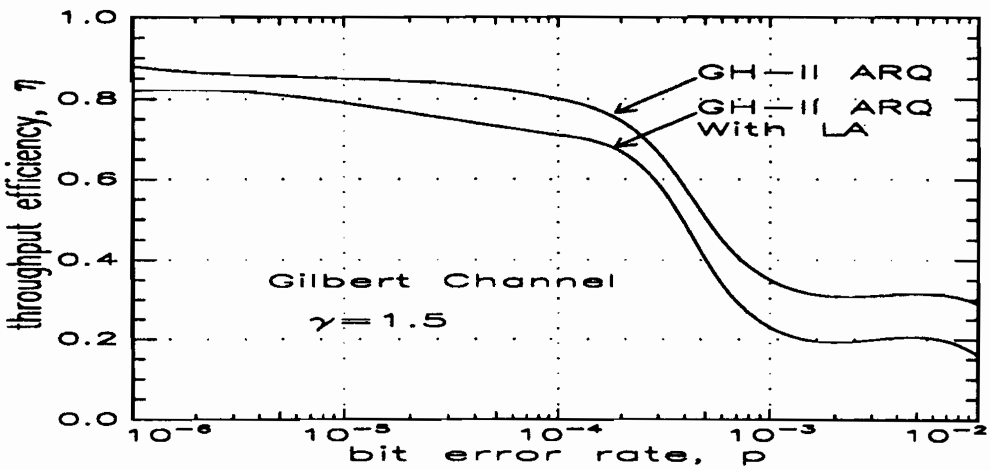

Figure 8.7 Effect of Loss Avoidance on Throughput Efficiency

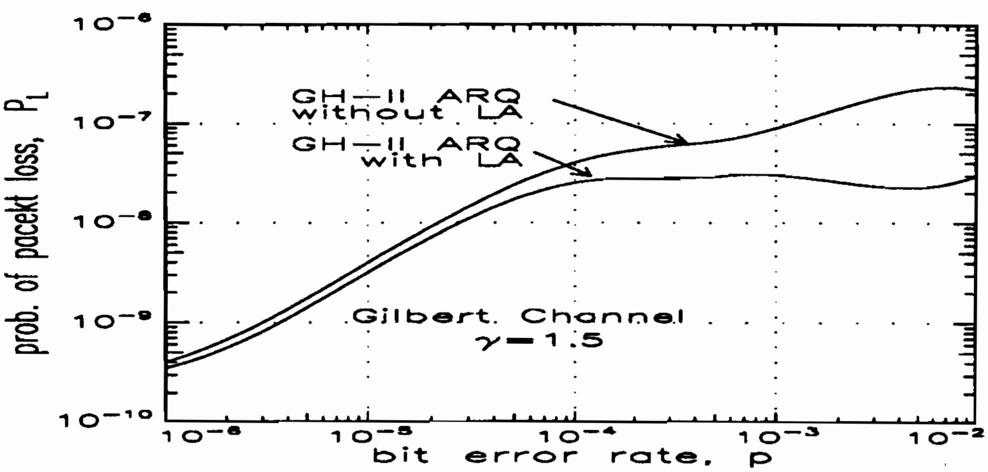

Figure 8.8 Effect of Loss Avoidance on Packet Loss

8.7 Results For The Markov Channel

Using the process constructed in Section 7.2.1, simulations were performed. The results presented here show that noise in the feedback channel results in a higher probability of undetected error and a lower throughput efficiency. A comparison is also made between the performance of the GH-II ARQ scheme and the SR-ARQ scheme. It is found that GH-II ARQ has better performance in terms of the probabilities of packet loss and spurious delivery. Figure 8.10 and Figure 8.15 show the results when a predictor is used with GH-II ARQ. These figures indicate that significant improvement is possible when prediction is used.

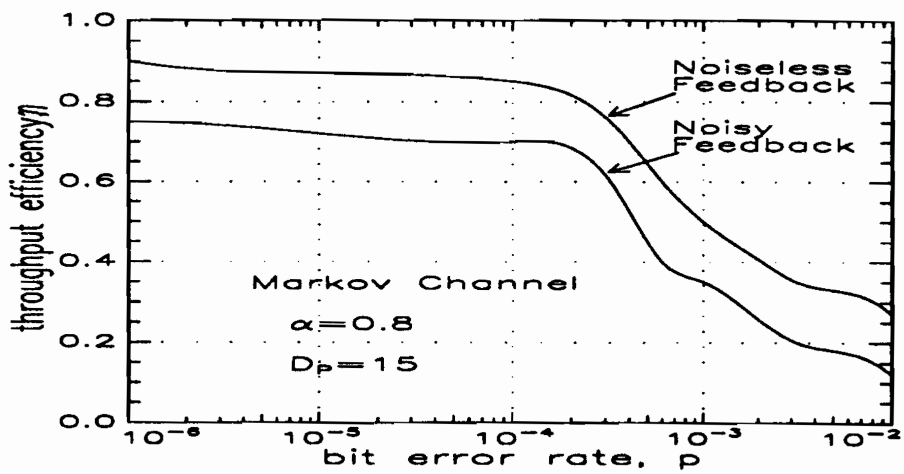

Figure 8.9 Throughput Efficiency on The Markov Channel

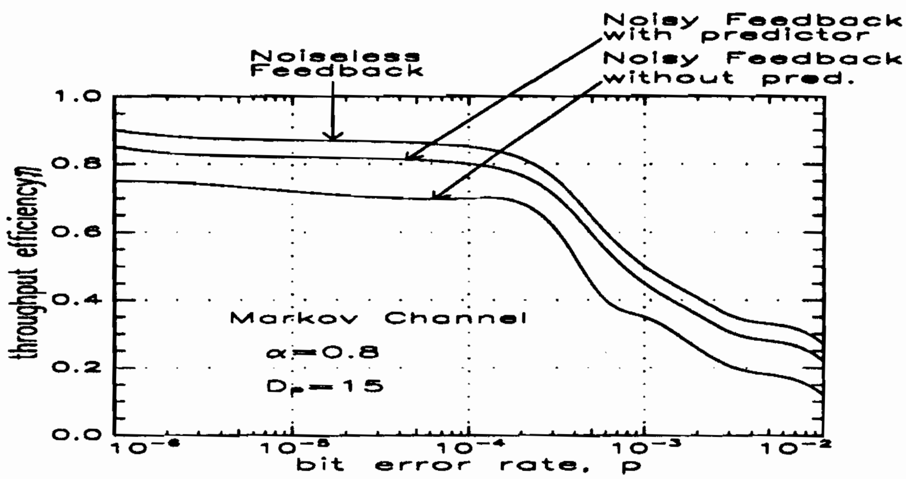

Figure 8.10 Throughput Comparisons on the Markov Channel

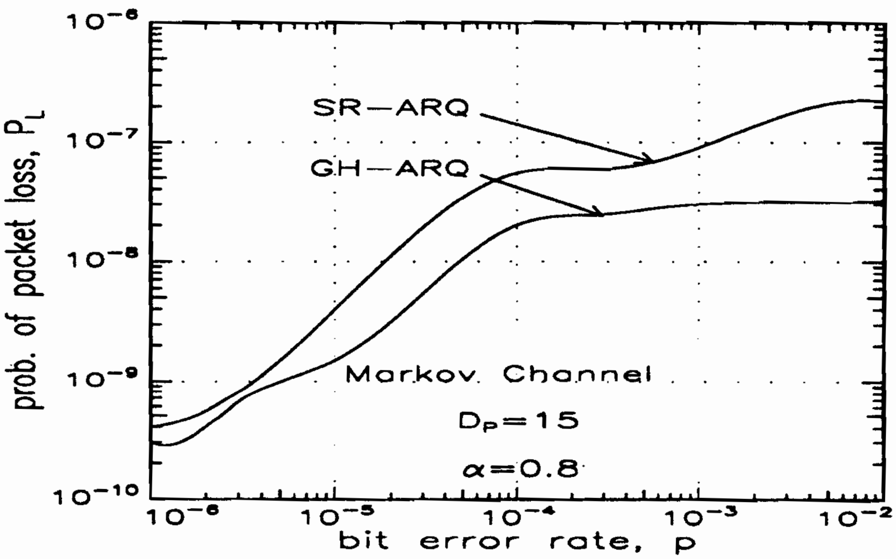

Figure 8.11 Probability of Packet Loss on The Markov Channel

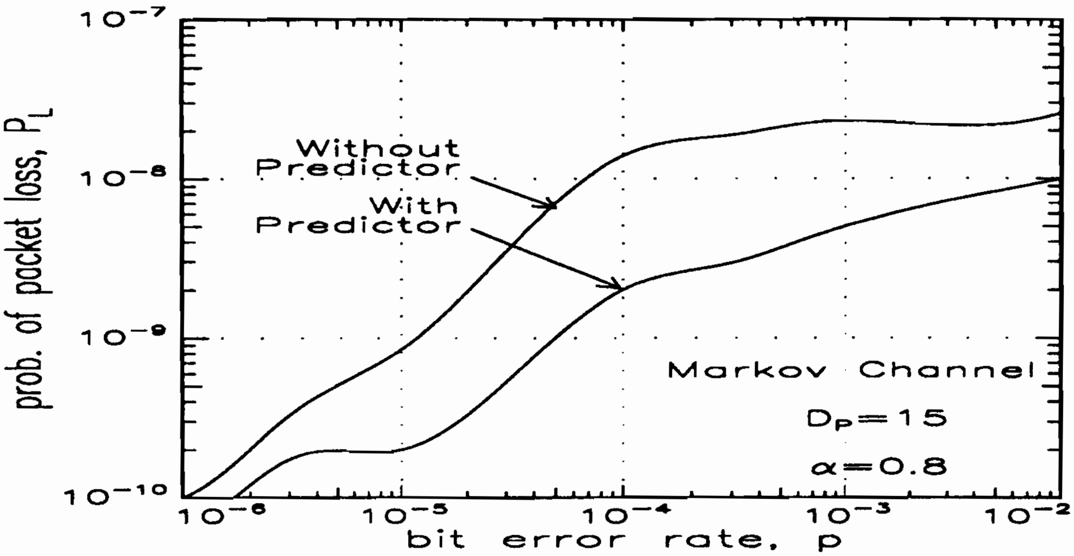

Figure 8.12 Effect of Prediction on Packet Loss on The Markov Channel

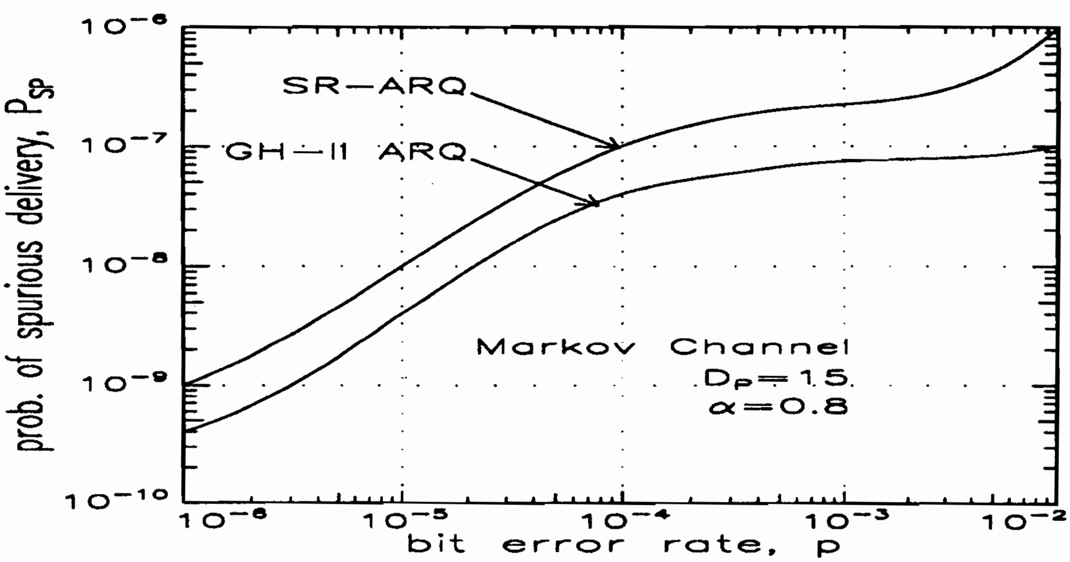

Figure 8.13 Spurious Delivery on the Markov Channel

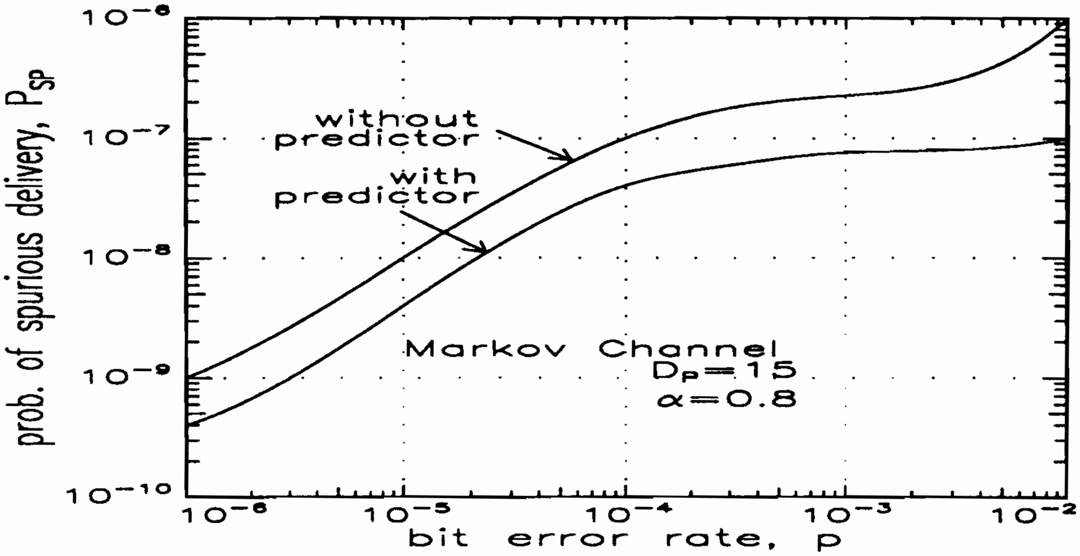

Figure 8.14 Effect of Prediction on Spurious Delivery

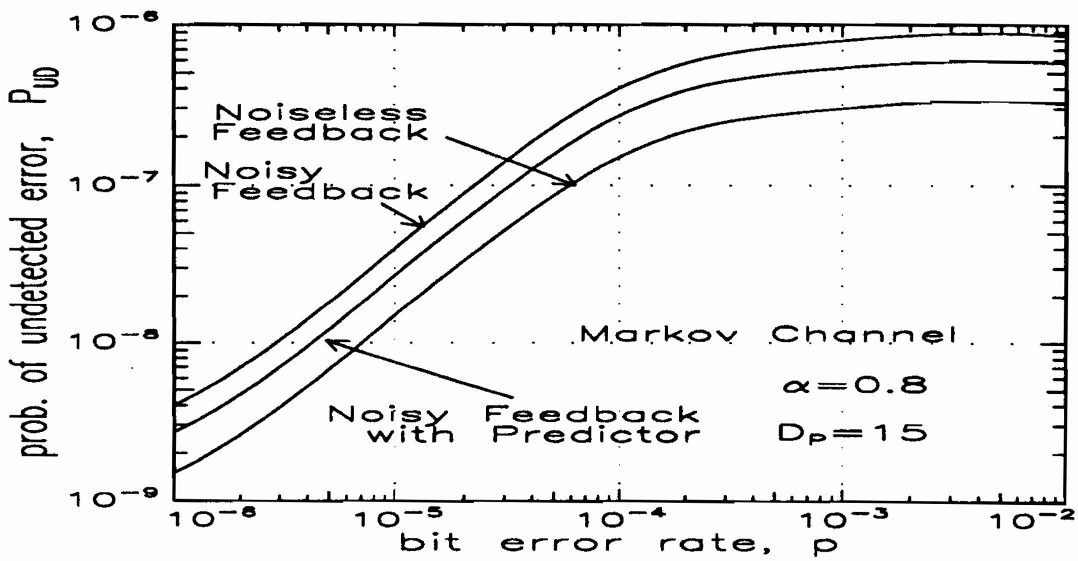

Figure 8.15 Effect of Prediction on Undetected Error Probability

CHAPTER IX
CONCLUSIONS

Overview

The applicability of the methods described in this work was discussed in Section 1.1, with the conclusion that significant performance improvement can be realized over a wide range of application areas. The effect of feedback errors on ARQ schemes has been examined by looking at the selective repeat ARQ scheme and the GH-II ARQ scheme using a Markov channel and a Gilbert channel. In the analysis of these systems for the Markov channel, appropriate variables were introduced that greatly facilitated the derivation of expressions for the relevant performance parameters. To provide a means of comparison, the noiseless assumption was also considered, and a parallel set of expressions were derived.

Since KM codes play an important role in the GH-II ARQ scheme treated here, their discussion was also included in Appendix A, with a detailed look at the (28,7,10) KM code. The encoder and decoder for this code were discussed, and the hardware diagrams provided for the decoder.

Channel models were also briefly considered; two models were singled out and used in the simulations to produce the results presented here. These were the Gilbert model and the Markov error model. An analysis of the Gilbert model was also given, a result of which was that the number of errors in successive blocks did not constitute a Markov chain. Despite this, the results of the Markov channel can be adopted in the Gilbert channel after some necessary modification. This is made possible by the fact that the events associated with certain conditional probabilities are separated by time intervals such that the performance evaluation can proceed as if the blocks were statistically independent. Using the above two models, performance results were compared under the noisy feedback and the noiseless feedback assumptions.

Finally, improvements were considered that cope with the effects of feedback errors. Two methods are proposed. One uses a predictor at the transmitter and the other uses the invertibility of KM codes in avoiding packet loss. For each improvement, simulation results are provided. In general, the following conclusions can be drawn from the analysis and simulation studies:

Effects of Feedback Noise on ARQ Performance

Noise in the feedback channel degrades the performance of ARQ schemes. In this work, the degradation has been studied by looking at individual events and seeing how their corresponding parameters are affected. The events of concern are, packet loss, spurious delivery, and undetected error. A brief summary of each of these is given in the following paragraphs, including their effects on the throughput efficiency and packet delay.

Packet Loss and Spurious Delivery

The loss of a packet is one undesirable consequence of errors in the return channel, and is more frequent the more noisy the return channel becomes. It would not occur in a system with an error-free feedback channel.

Throughput Efficiency and Packet Delay

Noise in the feedback channel introduces two undesirable events. One is the unnecessary retransmission of a packet that has already been delivered (either in error or error-free). The other is the loss of packets. Both of these tend to lower the throughput efficiency. As discussed in Section 3.6, a lower throughput efficiency results in a greater packet delay.

Undetected Error

From the results given here, the conclusion is that errors in the return channel increase the chance of making decoding errors. For the methods employing variable redundancy transmission, feedback errors cause the decoder to combine transmissions belonging to different packets and decode them as though they belonged together, thus creating obvious errors. For other methods, such as selective repeat ARQ, the unnecessary transmissions of a packet may still be delivered to the destination. These contribute to undetected errors. Both forms of ARQ will have increased undetected errors due to feedback errors.

The Predictor

The use of a predictor with the GH-II ARQ scheme was found to yield a significant performance improvement. For example, for a bit error rate of 10^{-4}, a round trip delay of 15 packet times, the improvement in throughput was found to be from 70% to 80%. A similar improvement was observed in the reliability. The predictor improvement was found to decrease as the link gets longer.

A conclusion is usually where one decides to take a break; it is rarely the end of all inquiry. Quite often, a lot more questions are brought into view than are answered. In the case at hand, we end by giving some issues that future work could address.

Suggestions for Further Work

Some problems that could be investigated are given below. This list is not exhaustive; other problems can easily be added.

1. Other forms of predictors could be studied. An example assuming a Markov error model is to estimate the state transition probability matrix, and use the estimate to decide on the starting depth for the next packet after a packet has been delivered. A simple method would be to choose the most likely next-state from the current state. The state space may be partitioned into classes to make the problem more manageable.

2. A two-way communication system could be studied. The KM scheme could be applied in both directions. An attempt can be made to see if expressions can be derived for the performance parameters of such a system.

3. The role of modulation on ARQ could be studied as well. This could be done with any of the above suggestions. Using modulation allows one to consider soft-decision decoding, and it is known that soft-decision decoding has an advantage over hard-decision decoding. For KM codes soft-decision decoders are easy to implement.

4. If data can be collected from real channels, then non renewal models could be considered. The parameters of the models can be assigned values as dictated by the data.

APPENDIX A: KM CODES

The reason for discussing KM codes here is that these codes were used in the simulations found in this work. In the course of the work, the encoder and decoder for one particular KM code were developed. Binary KM codes are obtained from algorithms for the computation of the aperiodic convolution of two sequences of lengths k and d over GF(2) [7]. The length n of the code equals the multiplicative complexity of the algorithm used. In the brief discussion of these codes given here, we skip several details, and concentrate on presenting what is necessary for the construction and decoding of these codes. A sequence $\{y_r; r = 0, 1, \ldots, \alpha - 1\}$ is represented as the polynomial $Y(u)$, where

$$Y(u) = \sum_{r=0}^{\alpha-1} y_r u^r \qquad (A.1)$$

The aperiodic convolution of the two sequences $\{ z_j\ j = 0,1,2, \ldots, k-1\}$ and $\{ y_l, l = 0,1,2, \ldots, d-1 \}$ over GF(2), is another sequence $\{\Phi_i; i = 1,2,\ldots,N-1\}$ where N=k+d-1. The parameter d is called the design minimum distance of the code. It is shown in [7] and [8] how the generator matrix for an (n, k,$\hat{d}$) code results from the aperiodic convolution computations, where $\hat{d}$ is the actual minimum distance of the resulting code. It is found that $\hat{d} \geq d$. The resulting convolution sequence can be represented as a polynomial $\Phi(u) = Z(u)Y(u)$. Since $\Phi(u)$ is of degree N-1, it is unchanged if it is defined modulo any polynomial $P(u)$ whose degree is at least N. Now, if $P(u)$ is a product of mutually prime polynomials $P_i(u)$, $i = 1,2,\ldots,t$, that is,

$$P(u) = \prod_{i=1}^{t} P_i(u), \qquad (A.2)$$

where $P_i(u)$ and $P_j(u)$ for $j \neq i$ have no common factors, then the polynomial product can be computed by first reducing each of the polynomials $Y(u)$ and $Z(u)$ modulo $P_i(u)$, to obtain $Z_i(u) \equiv Z(u)$ modulo $P_i(u)$ and $Y_i(u) \equiv Y(u)$ modulo $P_i(u)$. Next, the partial products $\Phi_i(u) \equiv Z_i(u)Y_i(u)$ modulo $P_i(u)$, $i = 1,2,3,\ldots,t$, are then computed. From these products, $\Phi(u)$ can be uniquely reconstructed using the Chinese Remainder Theorem. Let $M(P(u),N)$ be the multiplicative complexity of the polynomial product $Z(u)Y(u)$ modulo $P(u)$, ($P(u)$ has degree N). If in the computation, it is found that a polynomial $P'(u)$ of degree N-1 exists such that $M(P'(u),N-1) < M(P(u),N) - 1$, then it is more efficient to compute $Z(u)Y(u)$ using $P'(u)$ with one extra multiplication. In this case, it is said that one wraparound has been introduced. More wraparounds can be examined and the number yielding the lowest complexity chosen. The important fact for our purpose here is that the generator matrix of the code is block structured, having t+1 blocks. The first t blocks can be arranged so as to correspond to the factor polynomials $P_i(u)$ with the last block corresponding to the wraparound computation.

Example 1: Suppose k=4, d=5 $P(u) = u^2(u^2+1)(u^2+u+1)$, and $s = 2$. This choice results in the (12,4,5) code,[1] with the generator matrix[2]

$$\mathbf{M} = \begin{pmatrix} 101|101|111|000 \\ 011|011|011|000 \\ 000|101|101|011 \\ 000|011|110|101 \end{pmatrix} \qquad \begin{aligned} P_1(u) &= u^2 \\ P_2(u) &= u^2 + 1 \\ P_3(u) &= u^2 + u + 1 \\ \\ s &= 2 \end{aligned}$$

$$\begin{matrix} For & For & For & wrap \\ P_1(u) & P_2(u) & P_3(u) & s=2 \end{matrix} \qquad (A.3)$$

The codewords of this code are given in Table A.1.

Table A.1 shows that the actual minimum distance of the code is 6, whereas the design minimum distance is 5. The construction of KM codes is well-documented in [7]; here we present the construction of these codes by considering the (28,7,10) KM code. This code was found in an earlier work [51] to have better performance than other codes of the same depth. For this reason, it is used in the simulations presented. For this code, the polynomial *P(u)* is taken to be

1 This code is referred to here as (12,4,5) to reflect that the design distance d=5; it is shown next that the actual minimum distance is 6.

2 We will show shortly how the columns of the generator matrix are obtained from the factor polynomials.

Table A.1 Codewords of the (12,4,5) KM Code [94]	
CODEWORD	WEIGHT
1 1 0 1 1 1 0 0 0 1 0 0	6
0 0 1 0 1 1 1 0 1 1 0 0	6
1 1 1 1 0 0 1 0 1 0 0 0	6
0 1 0 1 1 0 1 0 0 0 1 1	6
1 0 0 0 0 1 1 0 0 1 1 1	6
0 1 1 1 0 1 0 0 1 1 1 1	8
1 0 1 0 1 0 0 0 1 0 1 1	6
0 0 1 1 1 1 0 1 0 0 0 1	6
1 1 1 0 0 0 0 1 0 1 0 1	6
0 0 0 1 0 0 1 1 1 1 0 1	6
1 1 0 0 1 1 1 1 1 0 0 1	8
0 1 1 0 0 1 1 1 0 0 1 0	6
1 0 1 1 1 0 1 1 0 1 1 0	8
0 1 0 0 1 0 0 1 1 1 1 0	6
1 0 0 1 0 1 0 1 1 0 1 0	6
0 0 0 0 0 0 0 0 0 0 0 0	0

WEIGHT DISTRIBUTION													
WEIGHT	0	1	2	3	4	5	6	7	8	9	10	11	12
#CODEWORDS	1	0	0	0	0	0	12	0	3	0	0	0	0

$$P(u) = u^2(u^2+1)(u^2+u+1)(u^3+u^2+1)(u^3+u+1) \tag{A.4}$$

and three wraparounds *(s = 3)* are selected. In the following section, we will include the factors u and $(u+1)$ and also two wraparounds *(s = 2)*, as these are used in the other codes that are subcodes of the (28,7,10) code. This point will become clearer as we proceed.

A.1 The Encoding Units

The information polynomial is the degree k-1 polynomial $Z(u)$, $(k = 7)$. The polynomial $Y(u)$ has degree d-1. The number of multiplications used in computing their product is the length of the code. $Z(u)$ is written as

$$Z(u) \quad = \quad z_0 + z_1 u + z_2 u^2 + \ldots + z_6 u^6. \tag{A.5}$$

The computation of the polynomial products necessitates reducing the polynomials $Y(u)$ and $Z(u)$ modulo each of the factor polynomials, $P_i(u)$. Note that

$$\begin{aligned} Z(u) &= z_0 \quad \text{modulo} \quad u \\ &= c_0, \end{aligned} \tag{A.6}$$

where c_0 corresponds to the first column of the generator matrix of the code. Also,

$$Z(u) = (z_0+z_1+z_2+\ldots+z_6) \quad \text{modulo} \quad u+1$$
$$= c_1. \qquad (A.7)$$

The contribution of the degree d-1 polynomial Y(u) is not discussed here, but the combined effect results in the first two columns of the generator matrix being equal to

$$\overset{c_0}{\begin{pmatrix}1\\0\\0\\0\\0\\0\\0\end{pmatrix}} \quad \text{and} \quad \overset{c_1}{\begin{pmatrix}1\\1\\1\\1\\1\\1\\1\end{pmatrix}}$$

From this point on, the column labels are going to be out of order; the reason for this is that we would like larger codes to contain smaller ones as subcodes. We now have

$$Z(u) = (z_0+z_2+z_4+z_6) + (z_1+z_3+z_5)u \quad \text{modulo} \quad u^2+1$$
$$= c_{16}+c_{17}u. \qquad (A.8)$$

By adding c_{16} and c_{17} modulo-2, we obtain c_1.

$$Z(u) = (z_0+z_2+z_3+z_5+z_6) + (z_1+z_2+z_4+z_5)u \quad \text{modulo} \quad u^2+u+1$$
$$= c_2 + c_7u \qquad (A.9)$$
$$c_8 = c_2+c_7. \qquad (A.10)$$

The columns corresponding to the last two factor polynomials are given below.

$c_{16}\, c_1\, c_{17}$ $\qquad\qquad$ $c_2\, c_8\, c_7$

$$\begin{pmatrix}110\\011\\110\\011\\110\\011\\110\end{pmatrix} \quad \begin{pmatrix}110\\011\\101\\110\\011\\101\\110\end{pmatrix}$$

$$Z(u) = (z_0+z_3+z_4+z_5) + (z_1+zz_4+z_5+z_6)u + (z_2+z_3+z_4+z_6)u^2 \text{ modulo } u^3+u^2+1$$

$$= c_3+c_4u+c_9u^2, \qquad (A.11)$$

$$c_{10}=c_3+c_4, \qquad (A.12a)$$

$$c_{11}=c_3+c_9, \qquad (A.12b)$$

$$c_5=c_4+c_9. \qquad (A.12c)$$

$$Z(u) = (z_0+z_3+z_4+z_5+z_6) + (z_1+z_3+z_4+z_5)u + (z_2+z_4+z_5+z_6)u^2 \text{ modulo } u^3+u+1$$

$$= c_{21}+c_{22}u+c_{23}u^2, \qquad (A.13)$$

$$c_{24}=c_{21}+c_{22}, \qquad (A.14a)$$

$$c_{26}=c_{21}+c_{23}, \qquad (A.14b)$$

$$c_{25}=c_{22}+c_{23}. \qquad (A.14c)$$

The columns arising from the preceding two factors are

$c_{10}c_3c_{11}c_4c_5c_9$ $\qquad$ $c_{24}c_{21}c_{26}c_{22}c_{25}c_{23}$

$$\begin{pmatrix}111100\\100010\\001111\\110011\\010001\\011110\\100101\end{pmatrix} \quad \begin{pmatrix}111000\\100110\\001011\\110011\\010101\\011110\\100101\end{pmatrix}$$

For the wraparound $s = 2$ we use the polynomial

$$Z(u) = z_5 + z_6 u$$

$$= c_{12} + c_6 u. \tag{A.15}$$

We add $c_1$2 and c_6 modulo-2 to obtain c_{13}. For the wraparound $s = 3$ we use the polynomial,

$$Z(u) = z_4 + z_5 u + z_6 u^2$$

$$= c_{18} + c_{12} u + c_6 u^2. \tag{A.16}$$

Then,

$$c_{20} = c_{12} + c_{18} \tag{A.17a}$$

$$c_{19} = c_{18} + c_6 \tag{A.17b}$$

$$c_{13} = c_6 + c_{12} \tag{A.17c}$$

These factors contribute to the first two blocks below; the third block (which is a single column) is for c_{27} and is obtained by adding the first 27 columns modulo-2.

$$\begin{array}{ccc} c_{12}c_{13}c_6 & c_{10}c_{18}c_{19}c_{12}c_{13}c_6 & & c_{27} \\ \begin{pmatrix} 000 \\ 000 \\ 000 \\ 000 \\ 000 \\ 110 \\ 011 \end{pmatrix} & \begin{pmatrix} 000000 \\ 000000 \\ 000000 \\ 000000 \\ 110000 \\ 100110 \\ 001001 \end{pmatrix} & \text{and} & \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \end{array}$$

We observe from the encoding units that the number of columns produced by each factor polynomial of degree α_i and the wraparound-(s) are, respectively,

$$\alpha_i + \binom{\alpha_i}{2} \quad \text{and} \quad s + \binom{s}{2}$$

With polynomial factors of degree at most three, higher-degree polynomials can be obtained by selecting these lower-degree polynomials as factors. So, to obtain an *(n, k, d)* KM code, we need to choose the polynomial factors and the wraparound (s) so that the following conditions hold:

1. $\text{GCD}\{P_i(u),\ P_j(u)\} = 1$ for $j \neq i$; and $i,j \in \{1,2,\ldots,t\}$.

2. $\sum_{i=1}^{t}\left[\alpha_i + \binom{\alpha_i}{2}\right] + s + \binom{s}{2} = n$ with $\alpha_i = \deg\{P_i(u)\}$.

3. $\deg\left[\prod_{j=1}^{t} P_i(u)\right] - s = k + d - 1.$

Each factor polynomial is then used to reduce an information polynomial of degree k-1. From these reductions, the columns of the generator matrix for the code are obtained as was done above for the (28,7,10) KM code. In [7], tables of polynomials of increasing degree are given and can be used in the steps above. We proceed to consider an example.

Example 2: Consider k=7, d=3 and $N = k + d\text{-}1 = 9$, that is, *P(u)* is a degree-9 polynomial. We can, however, use a degree-7 polynomial, $P(u) = u(u+1)(u^2+u+1)(u^3+u^2+1)$, and two intentional wraparounds (s=2). Then, applying the second requirement above, we see that $n = (1)+(1)+(2+1)+(3+3)+(2+1) = 14$. This approach leads to the (14,7,3) KM Code. Using the columns from the preceding discussion, we find the generator matrix of the code as

$$\mathbf{M} = \begin{pmatrix} 1|1|110|111000|000 \\ 0|1|011|100110|000 \\ 0|1|101|001011|000 \\ 0|1|110|110011|000 \\ 0|1|011|010101|000 \\ 0|1|101|011110|110 \\ 0|1|110|101101|011 \end{pmatrix}$$

$$u \quad u+1 \quad u^2+u+1 \quad u^3+u^2+1 \quad \begin{matrix}\text{wrap} \\ s=2\end{matrix}$$

At this point, the construction of KM codes should be clear; we now present the reverse process.

A.2 The Decoding of KM Codes

The decoder is obtained in three steps. First, we use the redundancy introduced by the encoding units to detect any uncorrectable errors. At this stage, we form parity equations for each decoding unit. A decoding unit corresponds to one of the polynomial factors. Those decoding units whose parity equations are not satisfied, are ignored in the next stage. Second, we form combinations of decoding units that will enable us to recover the information vector. If too many parity failures have occurred, leaving us with no candidates for decoding, we declare a decoding failure. Third, we take the decoding from each candidate and obtain its corresponding code vector. The difference between this vector and the received vector is called the error pattern corresponding to the candidate. The candidate whose error pattern has the smallest weight is selected. More precisely, let the received vector be $\mathbf{r} = (r_1 r_2 \ldots r_n)$ and let $\hat{\mathbf{Z}}_j = (z_{j,1} z_{j,2} \ldots z_{j,k})$, $k<n$, be the output from the jth candidate. This can be fed into a KM encoder to produce a code vector $\hat{\mathbf{C}}_j = (c_{j,1}\, c_{j,2} \ldots\, c_{j,n})$. The number of places where $\hat{\mathbf{C}}_j$ and $\mathbf{r}$ differ is the weight of the error pattern for the jth candidate. We denote this difference by $w_e(j)$. The decoder then accepts $\hat{\mathbf{Z}}_l$ as the information vector whenever

$$w_e(l) = \min_{j_1 \le j \le j_2} \{w_e(j)\}, \qquad (A.18)$$

where j_1 and j_2 are determined by the candidates examined. From the encoding units, we see that the factor polynomials of degree 2 and 3 introduce redundancy, since extra columns are produced by modulo-2 addition. Suppose $Z(u) \equiv v1 + (v2)u \bmod P_2(u)$ and $Z(u) \equiv w1 + (w2)u + (w3)u^2 \bmod P_3(u)$, where $\deg\{P_2(u)\} = 2$ and $\deg\{P_3(u)\} = 3$. Then, Figure A.1 shows how the other columns are obtained from the coefficients $v1$, $v2$, $w1$, $w2$, and $w3$.

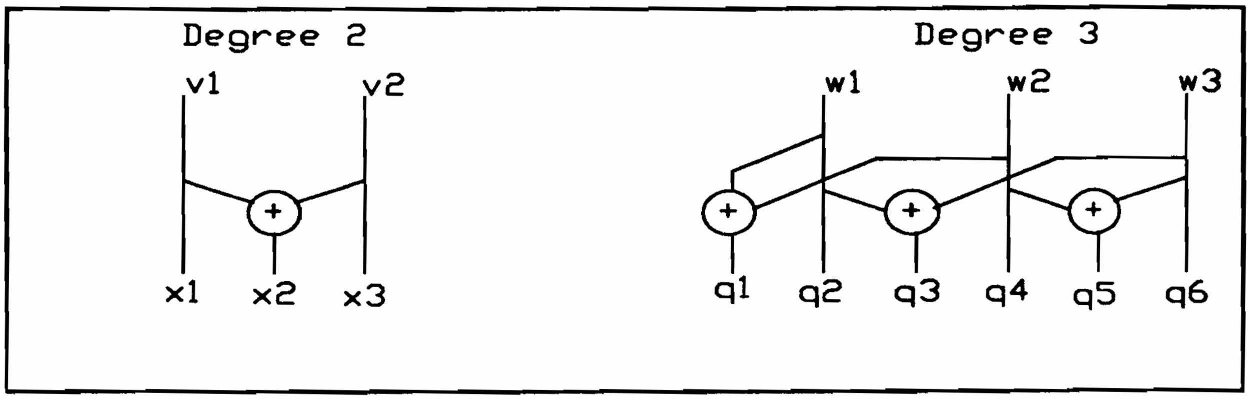

Figure A.1 Encoding Units

Before we can recover the original information bits, we must first remove the redundancy. This redundancy provides us with an over determined system of equations —we have more equations than unknowns. This can be exploited to correct/detect some of the transmission (or storage) errors. For the degree-2 polynomial, there is one extra equation which can be used to detect a single error in the three bits. Let $P = x1 + x2 + x3$ (modulo2). If $P = 0$, we accept $v1 = x1$ and $v2 = x3$, otherwise we reject the combination of $x1$, $x2$, and $x3$. Now $v1$ and $v2$ are expressed in terms of the z_i's. So from *v1* and *v2*, we get a pair of equations which we can use with other equations (other units) in the reconstruction of the information vector. The simple parity check decoder is a (3,2,2) decoder. This is shown in Figure A.2. This decoder is referred to later by the function

$$P(x1,x2,x3) \quad = \quad (v1,v2) \qquad (A.19)$$

From the degree-3 polynomial we need *w1*, *w2*, and *w3*. The equations for the qi's are given in Table A.2.

Table A.2 Determining the q's from the w's

$q1=w1+w2$	$q3=w1+w3$	$q5=w2+w3$,
$q2=w1$	$q4=w2$	$q6=w3$.

P(x1,x2,x3) = (v1,v2)

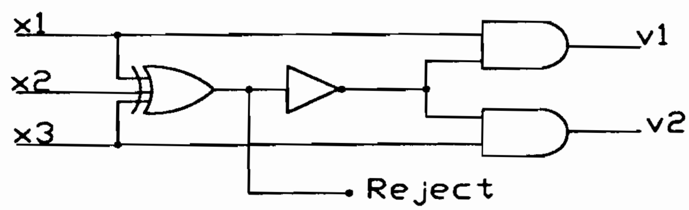

Figure A.2 The (3,2,2) Decoder

From the *q*'s we obtain the equations for the w's shown in Table A.3.

Table A.3 Determining the w's from the q's

$w1=q2$	$w2=q1+q2$,	$w3=q4+q5$,
$w1=q1+q4$,	$w2=q4$,	$w3=q2+q3$,
$w1=q3+q6$,	$w2=q5+q6$,	$w3=q6$.

It is observed that each *wi* is found in three possible ways. If there are errors, some of these ways will be different. The decoding procedure is to decide in favour of the majority: if two or three ways agree, then *wi* is the common value of these. A one-step majority logic decoder would suffice. This is shown in Figure A.3. This decoder is represented henceforth by the function

$$\mathrm{M}(q1,q2,q3,q4,q5,q6) \;=\; (w1,w2,w3). \tag{A.20}$$

This procedure corrects any single error in the six bits, *q1*, *q2*, *q3*, *q4*, *q5*, and *q6*. To detect two errors, some further equations are needed. The three parity equations

$$P_1 = q1 + q2 + q4, \tag{A.21a}$$

$$P_2 = q2 + q3 + q6, \tag{A.21b}$$

$$P_3 = q4 + q5 + q6, \tag{A.21c}$$

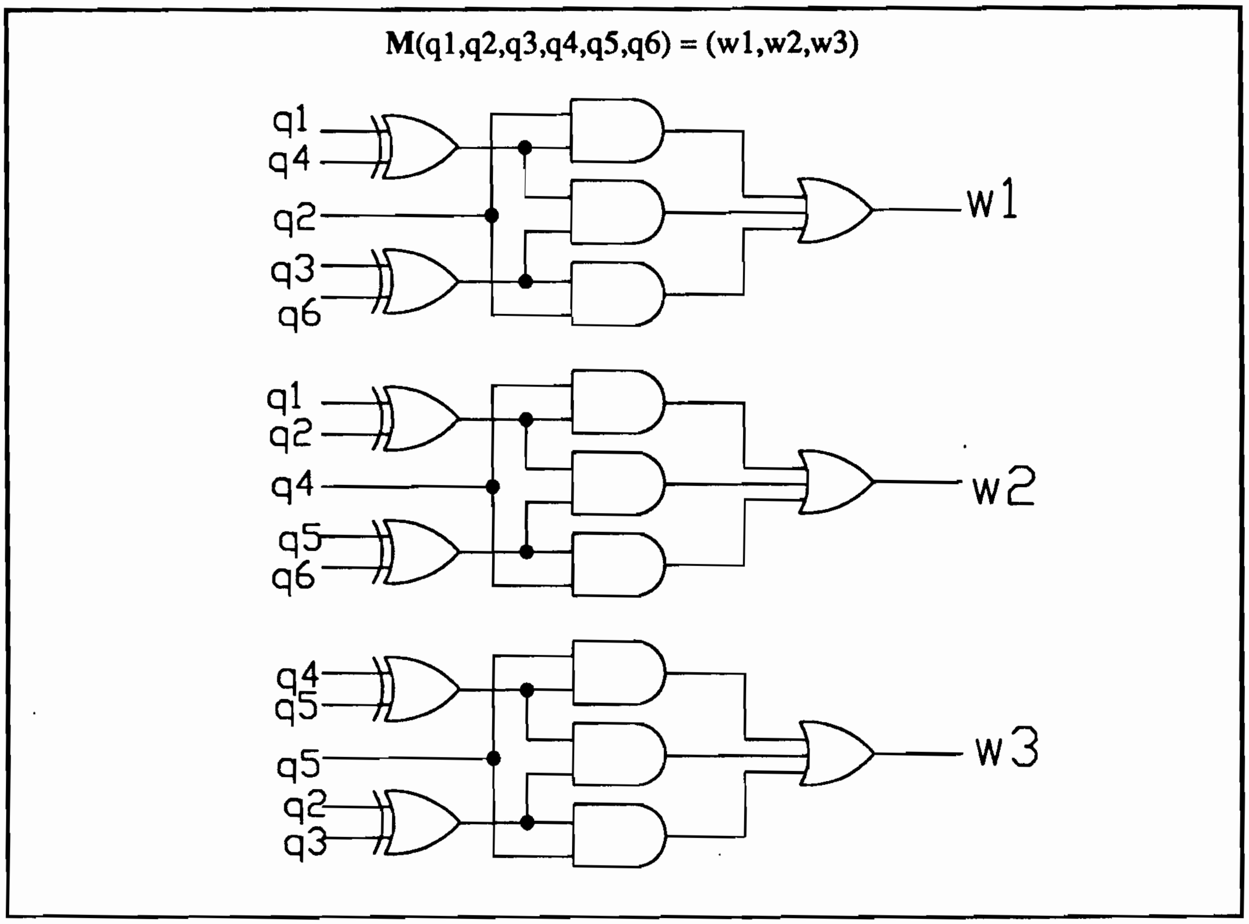

Figure A.3 The (6,3,3) Majority Logic Decoder

can be used to determine when two or more errors have occurred. This is illustrated by the logic diagram of Figure A.4.

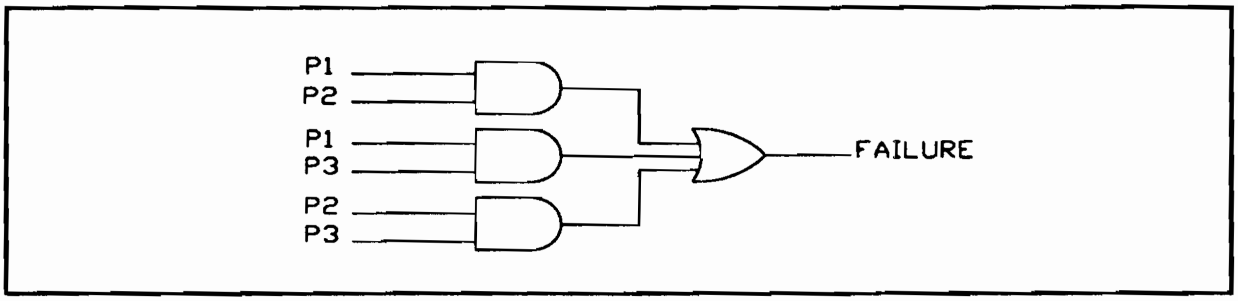

Figure A.4 Double Error-detector for a Degree-3 unit

In this way, the unit can correct one error and detect some patterns with two errors; some two-error patterns are not detectable. For example, if *q1* and *q2* are both in error, then the decoder will not detect this.

Again, to make the discussion concrete, we use the (28,7,10) code. This code contains the (14,7,3) code, and the (21,7,6) code. These are referred to, respectively, as depth-2 and depth-3 codes. The (28,7,10) code is a depth-4 code. The polynomials used are

$(14,7,3)\text{Code}: u(u+1)(u^2+u+1)(u^3+u^2+1), \quad s=2$

$(21,7,6)\text{Code}: u^2(u^2+1)(u^2+u+1)(u^3+u^2+1), s=3$

$(28,7,10)\text{Code}: u^2(u^2+1)(u^2+u+1)(u^3+u^2+1)(u^3+u+1), s=3$

plus an overall parity bit (c_{27}).

We note here that in going from depth-4 to depth-3, we ignore the factor (u^3+u+1). This is useful in adaptive ARQ schemes, in which it is desirable that, in going from one code to a larger one, the decoder configuration should stay more or less the same. We see that depth-3 and depth-4 differ only through the above factor polynomial. In going from depth-3 to depth-2, we disable the units corresponding to (u^2), (u^2+1), and *(s=2)* and activate the decoding units for *(u)* and *(u+1)*. Using the columns derived earlier on, the generator matrix is found to be

$$M = \begin{pmatrix} 1111000|0101100|0110000|1001010 \\ 0100110|1101000|1101000|0101100 \\ 0110010|1010100|0010000|0010110 \\ 0111010|0111000|0001000|1100110 \\ 0101100|1110000|0010111|0111010 \\ 0111110|1000111|0001001|1110000 \\ 0110101|0111101|0010010|1011101 \end{pmatrix}. \tag{A.22}$$

From this matrix, one obtains the encoding equations in Table A.4.

Table A.4 Encoding Equations

$c_0 = z_0$	$c_{14} = z_1$
$c_1 = z_0 + z_1 + z_2 + z_3 + z_4 + z_5 + z_6$	$c_{15} = z_0 + z_1$
$c_2 = z_0 + z_2 + z_3 + z_5 + z_6$	$c_{16} = z_0 + z_2 + z_4 + z_6$
$c_3 = z_0 + z_2 + z_4 + z_5$	$c_{17} = z_1 + z_3 + z_5$
$c_4 = z_1 + z_4 + z_5 + z_6$	$c_{18} = z_4$
$c_5 = z_1 + z_2 + z_3 + z_5$	$c_{19} = z_4 + z_6$
$c_6 = z_6$	$c_{20} = z_4 + z_5$
$c_7 = z_1 + z_2 + z_4 + z_5$	$c_{21} = z_0 + z_3 + z_5 + z_6$
$c_8 = z_0 + z_1 + z_3 + z_4 + z_6$	$c_{22} = z_1 + z_3 + z_4 + z_5$
$c_9 = z_2 + z_3 + z_4 + z_6$	$c_{23} = z_2 + z_4 + z_5 + z_6$
$c_{10} = z_0 + z_1 + z_3 + z_6$	$c_{24} = z_0 + z_1 + z_4 + z_6$
$c_{11} = z_0 + z_2 + z_5 + z_6$	$c_{25} = z_1 + z_2 + z_3 + z_6$
$c_{12} = z_5$	$c_{26} = z_0 + z_2 + z_3 + z_4$
$c_{13} = z_5 + z_6$	$c_{27} = z_6$

From these equations, we find that the first three blocks are invertible, and have the inversion equations given in Table A.5.

Table A.5 Inversions

Block-1	Block-3
$z_0 = c_0$	$z_0 = c_{14} + c_{15}$
$z_1 = c_0 + c_2 + c_5 + c_6$	$z_1 = c_{14}$
$z_2 = c_0 + c_1 + c_2 + c_3 + c_5$	$z_2 = c_{14} + c_{16} + c_{19}$
$z_3 = c_2 + c_3 + c_4 + c_5$	$z_3 = c_{14} + c_{15} + c_{17} + c_{18} + c_{20}$
$z_4 = c_0 + c_1 + c_5 + c_6$	$z_4 = c_{18}$
$z_5 = c_1 + c_2 + c_4 + c_6$	$z_5 = c_{18} + c_{20}$
$z_6 = c_6$	$z_6 = c_{18} + c_{19}$
Block-2	**Block-4 is not invertible**
$z_0 = c_7 + c_9 + c_{10} + c_{12}$	
$z_1 = c_8 + c_9 + c_{11} + c_{13}$	
$z_2 = c_7 + c_9 + c_{10} + c_{11} + c_{12} + c_{13}$	
$z_3 = c_7 + c_8 + c_{11}$	
$z_4 = c_8 + c_{10}$	
$z_5 = c_{12}$	
$z_6 = c_{12} + c_{13}$	

A.3 Decoding Candidates

The candidates are combinations of polynomial factors whose product has degree at least 7. We potentially have 19 candidates, but at any one time, only a few of these are used. We shall use the numbers 1 to 9 to represent the polynomials, as shown in Table A.6.

Table A.6 Unit Indexing

1	2	3	4	5	6	7	8	9
u^2	u^2+1	u^2+u+1	u^3+u^2+1	u^3+u+1	$s=3$	$s=2$	u	$u+1$

The selection S_1 = { 1,2,4 } corresponds to the polynomial product $u^2(u^2+1)(u^3+u^2+1)$. This is a degree-7 polynomial, and can be used to recover the information polynomial, provided the appropriate parity equations are satisfied. Many other combinations are selected to satisfy the equation

$$s \; + \; \text{degree}\left[\prod_{i=1}^{t} P_i(u)\right] \; \geq \; k \qquad (A.23)$$

The P_i are the factor polynomials mentioned above, and s refers to the wraparound. The selections are denoted S_i, $i = 1,2,...,19$ and are given in Table A.7.

Table A.7 Decoding Selections

	S_7 = {1,4,6}	
S_1 = {1,2,3}	S_8 = {2,4,6}	S_{14} = {1,5,6}
S_2 = {2,3,6}	S_9 = {3,4,6}	S_{15} = {2,5,6}
S_3 = {1,2,6}	S_{10} = {1,4,5}	S_{16} = {2,3,5}
S_4 = {1,3,6}	S_{11} = {1,2,5}	S_{17} = {3,4,5}
S_5 = {1,3,4}	S_{12} = {1,3,5}	S_{18} = {3,5,6}
S_6 = {2,3,4}	S_{13} = {2,4,5}	S_{19} = {3,4,7}

Each selection gives a set of equations which can be solved to find the coefficients of the information polynomial *Z(u)*. The steps are omitted. The first two selections are given below with the respective decoding equations. A complete listing of the selections and their equations is given in Appendix D.

$S_1 = \{1,2,4\}$	$z_0 = c_0$
	$z_1 = c_{14}$
$(c_0, c_{14}) = \mathbf{P}(r_0, r_{15}, r_{14})$	$z_2 = c_0 + c_4 + c_9 + c_{14} + c_{17}$
$(c_{16}, c_{17}) = \mathbf{P}(r_{16}, r_1, r_{17})$	$z_3 = c_0 + c_9 + c_{16}$
	$z_4 = c_0 + c_3 + c_{14} + c_{17}$
$(c_3, c_4, c_9) = \mathbf{M}(r_{10}, r_3, r_{11}, r_4, r_5, r_9)$	$z_5 = c_0 + c_9 + c_{14} + c_{16} + c_{17}$
	$z_6 = c_0 + c_3 + c_4 + c_9 + c_{16}$

$S_2 = \{2,3,6\}$	$z_0 = c_2 + c_6 + c_7 + c_{17} + c_{18}$
	$z_1 = c_2 + c_{16} + c_{17} + c_{18}$
$(c_{16}, c_{17}) = \mathbf{P}(r_{16}, r_1, r_{17})$	$z_2 = c_2 + c_7 + c_{12} + c_{16} + c_{17}$
$(c_2, c_7) = \mathbf{P}(r_2, r_8, r_7)$	$z_3 = c_2 + c_{12} + c_{16} + c_{18}$
	$z_4 = c_{18}$
	$z_5 = c_{12}$
$(c_{18}, c_{12}, c_6) = \mathbf{M}(r_{20}, r_{18}, r_{19}, r_{12}, r_{13}, r_6)$	$z_6 = c_6$

In the above tables, the vector $\mathbf{r} = (r_0, r_1, \ldots, r_{26}, r_{27})$ is the received vector.

A.4 Determining the Decoding Candidates

Parity equations are used here to determine the eligible decoding selections. The computation of these parity equations begins with the logic diagrams in the following figures. The additions are modulo-2.

A.4.1 Depth 2

$P1=r_2+r_7+r_8$ $P_2=r_6+r_{12}+r_{13}$ $Q_2=r_3+r_4+r_{10}$ $Q_4=r_3+r_9+r_{11}$ $Q_5=r_4+r_5+r_9$

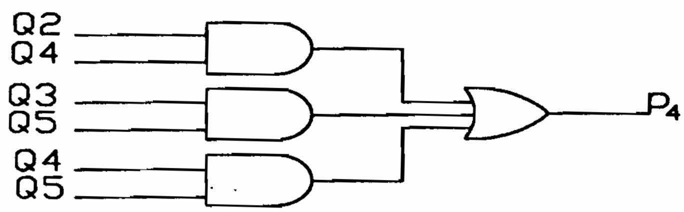

Figure A.5 Depth-2 Candidate Selection

A.4.2 Depth 3

Here we use P_1, P_4, and P_3, P_6, and P_7 (given below)

$P3=r_1+r_{16}+r_{17}$ $P_7=r_0+r_{14}+r_{15}$ $Q_8=r_6+r_{12}+r_{13}$ $Q_9=r_6+r_{18}+r_{19}$ $Q_{10}=r_{12}+r_{18}+r_{20}$

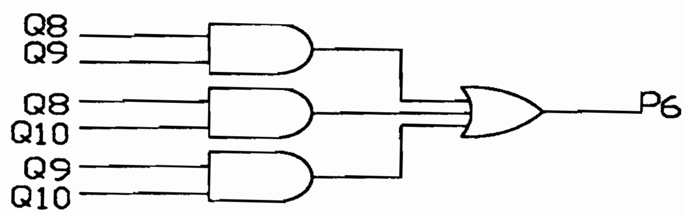

Figure A.6 Depth-3 Candidate Selection

A.4.3 Depth 4

This part uses P_1, P_3, P_4, P_6, P_7, and P_5 (below).

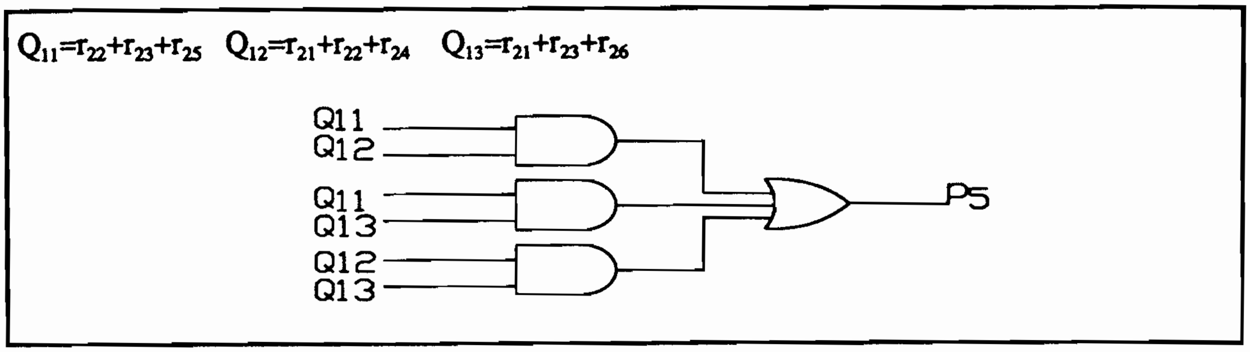

Figure A.7 Depth-4 Candidate Selection

A.4.4 Depth-2 Decoding

This is the simplest of them all; only one parity equation ($A_0 = \overline{P}_1\overline{P}_2\overline{P}_4$) is needed, and only one candidate *(S_{19})* is examined. If the parity fails ($\overline{P}_1\overline{P}_2\overline{P}_4 = 0$), we declare a decoding failure.

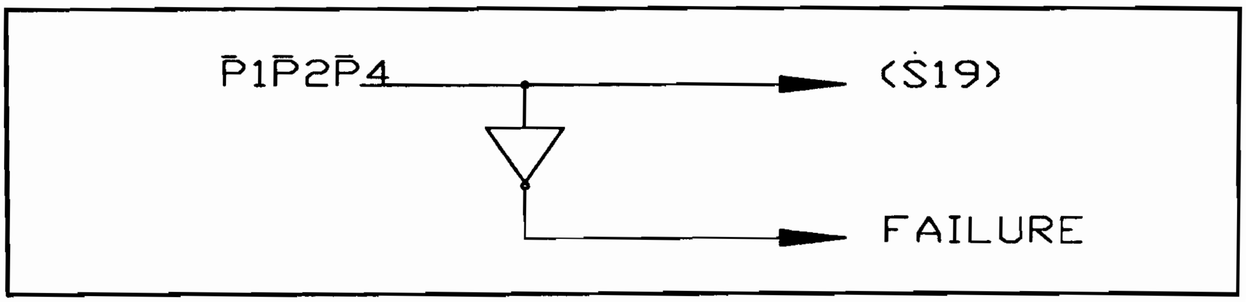

Figure A.8 Depth-2 Decoding Failure Detection

A.4.5 Depth-3 Decoding

There are sixteen possible sets: $A_0, A_1, A_2, \ldots, A_{14}\,A_{15}$,each A_i selecting a subset of decoding selections. Appendix D lists the corresponding decoding selections. The decoder has to indicate whether or not a decoding failure has occurred. This is done by examining FAILURE = NOR($A_1, A_2, A_3, \ldots, A_{14}, A_{15}$).

A.4.6 Depth-4 Decoding

Here we have 37 options: FAILURE = NOR($A16, A17, A18, A19, \ldots, A51, A52$), and their details are also found in Appendix-D. Finally, the weight distribution of the (28,7,10) KM code is given in Table A.8. The list has omitted those weights for which there are no codewords. The decoder for a general KM code is available in [7]. For the specific (28,7,10) KM code, the descriptions given above are summarized in the diagrams of Figures A.9 - A.12. This is the decoder used in the simulations presented in this thesis.

Table A.8 The Weight Distribution of the (28,7,10) KM Code

WEIGHT	0	10	12	14	16	18	20	22
#CODEWORDS	1	10	29	26	38	27	9	1

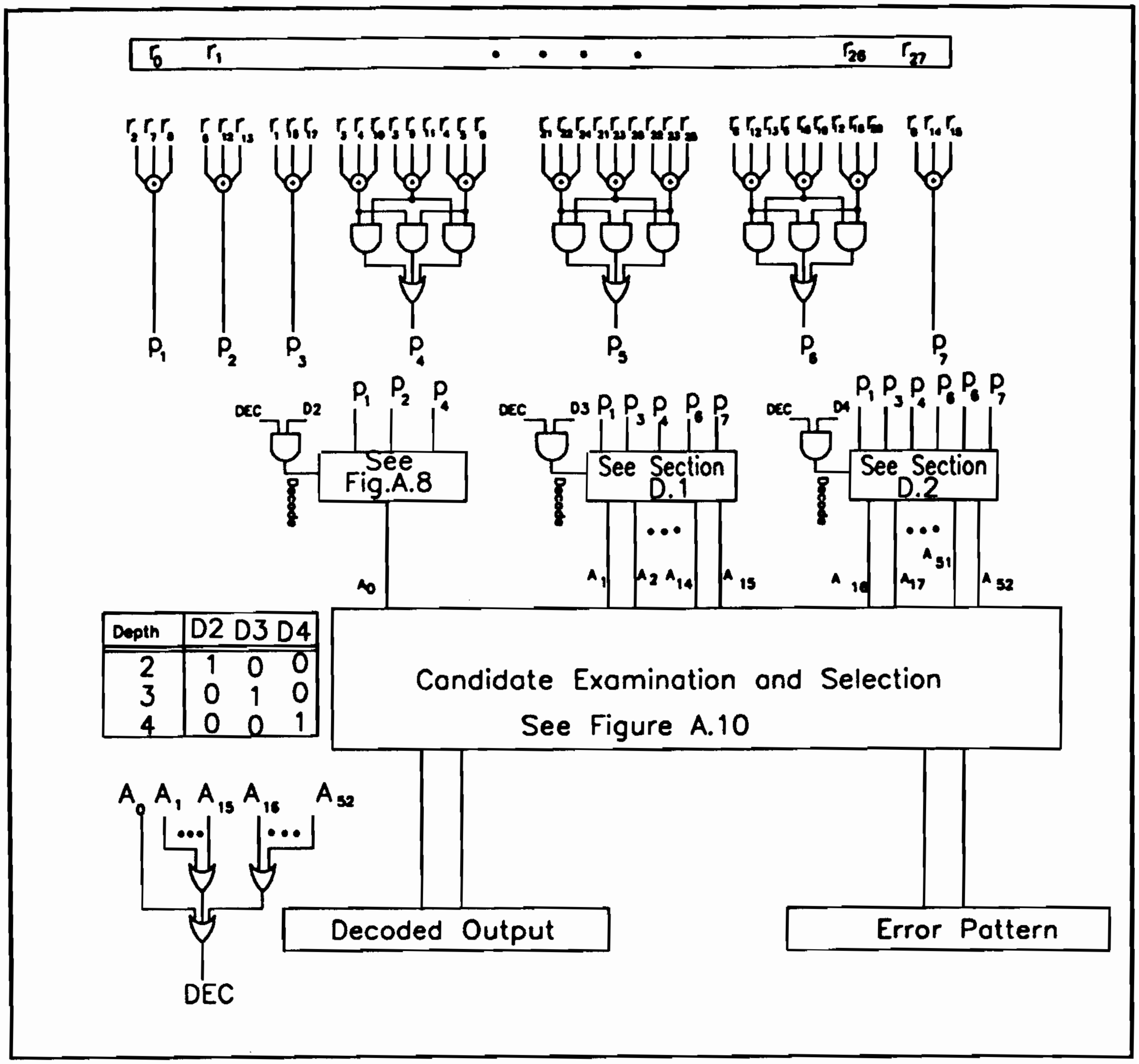

Figure A.9 The (28,7,10) KM Decoder.

Of the input variables, $A_0,A_1,A_2,\ldots,A_{52}$, in Figure A.10, only one can be active at a time. That is, $A_i=1$ and $A_j=0$ for $j\neq i$. The active A_i determines the decoding candidates $S_{l_1}, S_{l_2}, S_{l_3}$, etc., as described in Sections D.1 and D.2 of Appendix D.

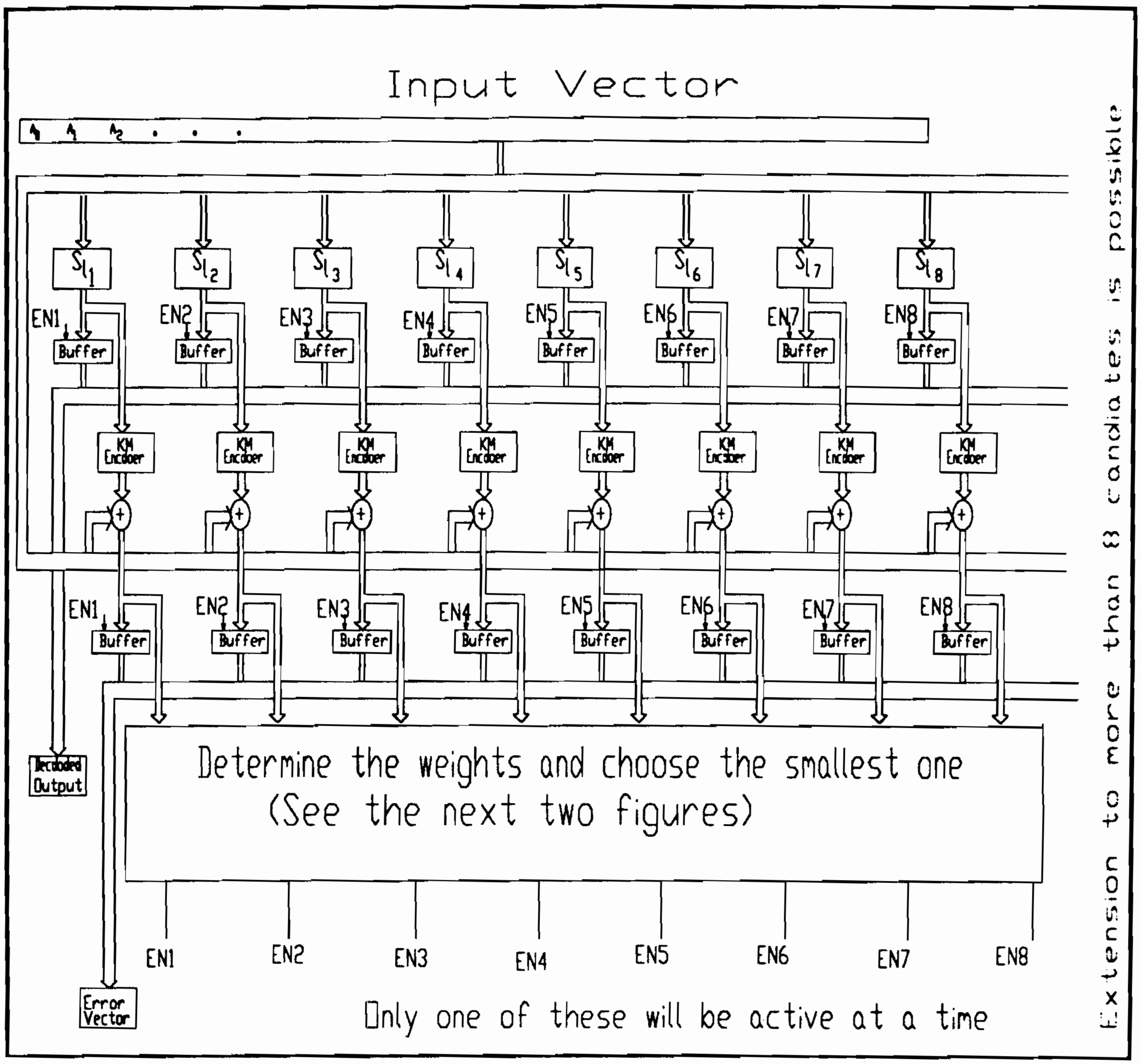

Figure A.10 Examination of Decoding Candidates.

Depth	D2	D3	D4	#Shifts	Counter
2	1	0	0	14	0 → 13
3	0	1	0	21	0 → 20
4	0	0	1	27	0 → 27

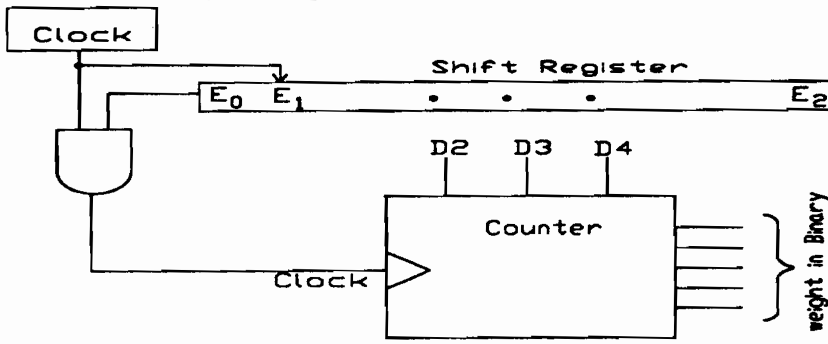

Figure A.11 Determining Error Weight.

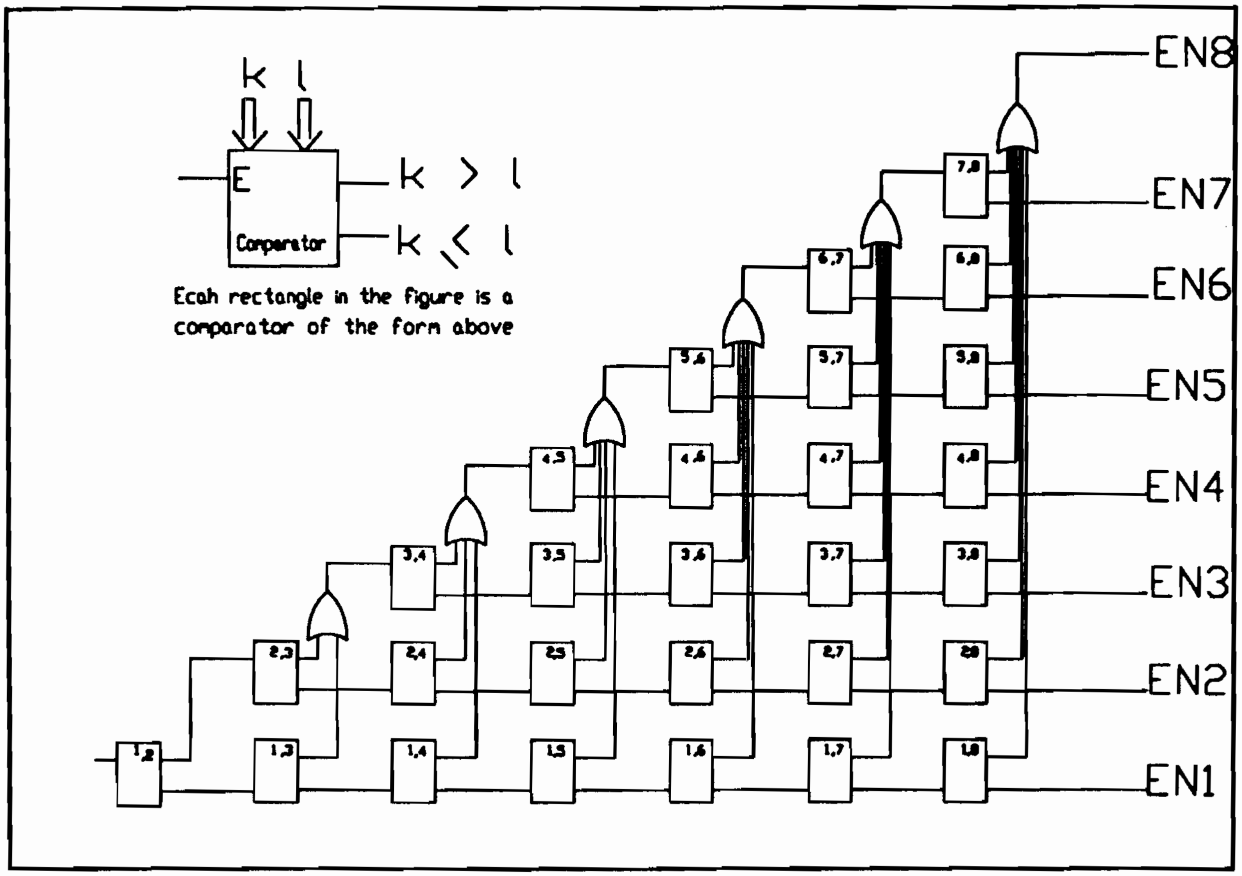

Figure A.12 Selecting the Smallest Weight.

APPENDIX B: DECODER TRANSITIONS

The KM decoder takes the received block and partitions it into sub-blocks for decoding as shown earlier in Figures 3.3 and 3.4 of Chapter III. For each sub-block, the decoding can end in one of three ways: errors may be correctable, detectable, or undetectable. Once errors are detected in one sub-block, the whole block is rejected. If errors are undetected, then the whole block is considered to contain undetected errors. These considerations give rise to the transition diagram of Figure B.1 below for the KM decoder.

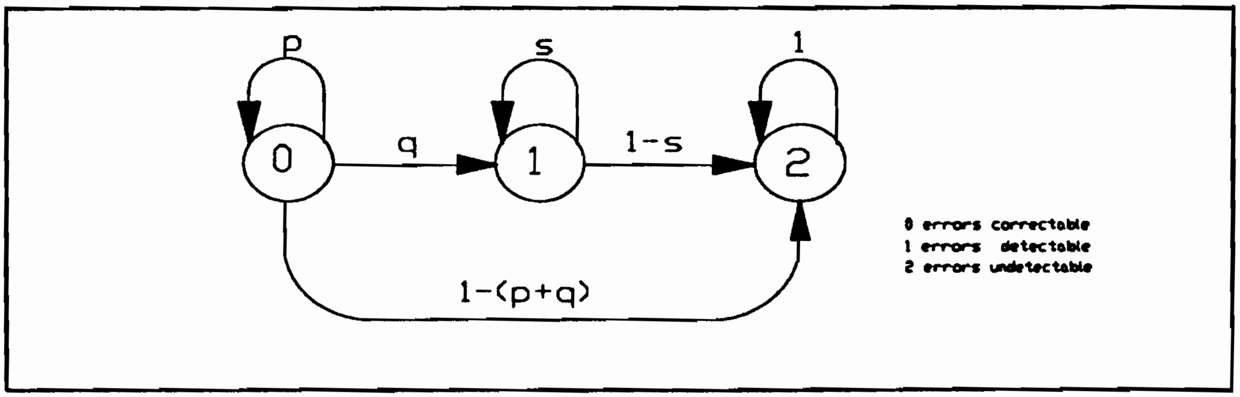

Figure B.1: State Transition Diagram of the KM Decoder

In order to decode the block, the decoder undergoes $(n_1/k_2)-1$ transitions, where n_l is the length of the outer code C_{ED} and k_2 is the number of information bits in the KM code. The (28,7,10) KM code is used here, since it was found in an earlier work [91] to be better than other codes of the same depth. The packet length used is $n_l = 504$ and $k_2 = 7$. This gives the decoder a total of 71 transitions. The transition probabilities are given by the matrix

$$\mathbf{P}_{KM} = \begin{pmatrix} p & q & 1-(p+q) \\ 0 & s & (1-s) \\ 0 & 0 & 1 \end{pmatrix}. \qquad (B.1)$$

It can be shown [40] that

$$\mathbf{P}_{KM}^{k} = \begin{pmatrix} p^{k} & \dfrac{q(p^{k}-s^{k})}{(p-s)} & 1-\left(\dfrac{(p-s)p^{k}+q(p^{k}-s^{k})}{(p-s)}\right) \\ 0 & s^{k} & 1-s^{k} \\ 0 & 0 & 1 \end{pmatrix} \qquad (B.2)$$

For each decoding, the decoder always starts in state 0. The performance of the decoder can be evaluated once p, q and s are known. These can be determined from the error statistics.

APPENDIX C: CHANNEL TRANSITIONS

We have determined all the expressions required for the performance evaluation parameters. We now face the task of determining the state transition probabilities, as these are needed in the expressions. To begin, we note that there is a time lag between the reception of a packet and the arrival of its acknowledgement at the transmitter. This is depicted in Figure C.1.

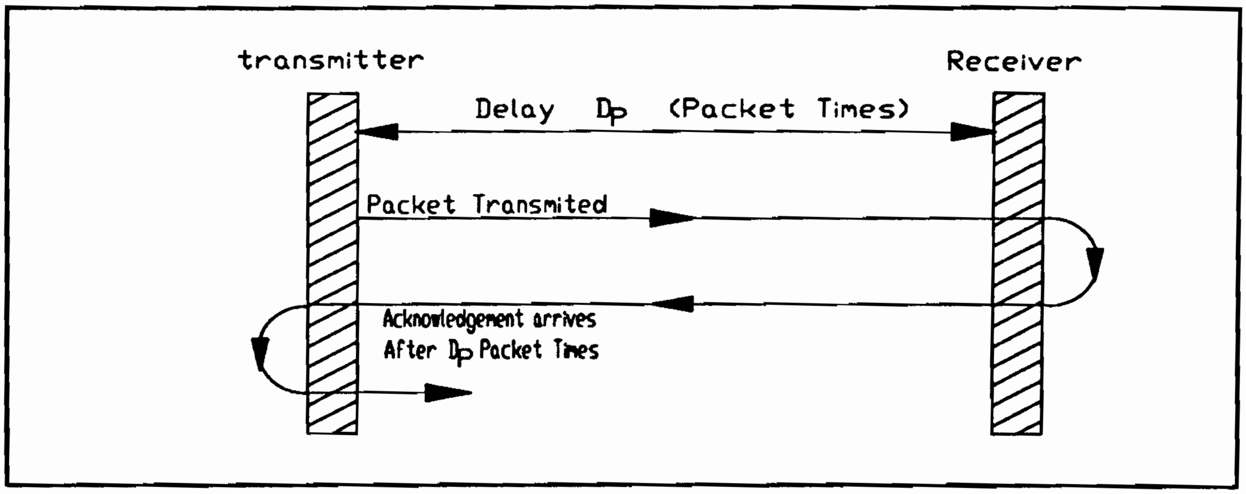

Figure C.1: Delay Between Packet Arrival and Its Acknowledgement

In order to take this delay into account, we introduce the matrices given below. Let **P** be the matrix of one-step transition probabilities.[3]

$$\mathbf{P} = \begin{pmatrix} P(C/C) & P(D/C) & P(E/C) \\ P(C/D) & P(D/D) & P(E/D) \\ P(C/E) & P(D/E) & P(E/E) \end{pmatrix} \qquad (C.1)$$

Let **Q** be the matrix of one-step transition probabilities from the set {C,D,E} to the set {T,F}. The D_P transitions required are broken into D_{P-1} transitions plus a final transition. By appropriate manipulations, it can be shown that

3 It is assumed here that the states of the chain have been partitioned to accord with the events C, D, and E as used here.

$$\mathbf{P}^{D_p-1}\mathbf{Q} = \begin{pmatrix} P(T/C) & P(F/C) \\ P(T/D) & P(F/D) \\ P(T/E) & P(F/E) \end{pmatrix} \qquad (C.2)$$

To keep the expressions simple, we assume that the same code is used on the return channel as in the forward channel. In this case a *false* feedback message is equivalent to E, and *a true* feedback message is equivalent to $C \cup D$. This implicitly says that the feedback message will be false only when the errors are undetectable. With this in mind, we define the matrices **Q, R,** and **S** as given below.

$$\mathbf{Q} = \begin{pmatrix} P(C \cup D/C) & P(E/C) \\ P(C \cup D/D) & P(E/D) \\ P(C \cup D/E) & P(E/E) \end{pmatrix} = \begin{pmatrix} P(C/C)+P(D/C) & P(E/C) \\ P(C/D)+P(D/D) & P(E/D) \\ P(C/E)+P(D/E) & P(E/E) \end{pmatrix} \qquad (C.3)$$

$$\mathbf{R} = \begin{pmatrix} P(C \cup D/C \cup D) & P(E/C \cup D) \\ P(C \cup D/E) & P(E/E) \end{pmatrix} \qquad (C.4)$$

and

$$\mathbf{S} = \begin{pmatrix} P(C/C \cup D) & P(D/C \cup D) & P(E/C \cup D) \\ P(C/E) & P(D/E) & P(E/E) \end{pmatrix} \qquad (C.5)$$

With these matrices, we have

$$\mathbf{P}^{D_p-1}\mathbf{Q} = \mathbf{Q}\mathbf{R}^{D_p-1} = \begin{pmatrix} P(T/C) & P(F/C) \\ P(T/D) & P(F/D) \\ P(T/E) & P(F/E) \end{pmatrix} \qquad (C.6)$$

$$\mathbf{R}^{D_p-1}\mathbf{S} = \mathbf{S}\mathbf{P}^{D_p-1} = \begin{pmatrix} P(C/T) & P(D/T) & P(E/T) \\ P(C/F) & P(D/F) & P(E/F) \end{pmatrix} \qquad (C.7)$$

The number of packet times from one feedback message to the next (for the same packet) is $2D_P$. Using this fact, we have

$$\mathbf{R}^{2D} = \begin{pmatrix} P(T/T) & P(F/T) \\ P(T/F) & P(F/F) \end{pmatrix} \tag{C.8}$$

Finally, we need the matrix $\mathbf{P}^{2D}$ given as

$$\mathbf{P}^{2D} = \mathbf{Q}\mathbf{R}^{2D_P-2}\mathbf{S} = \begin{pmatrix} P(C/C) & P(D/C) & P(E/C) \\ P(C/D) & P(D/D) & P(E/D) \\ P(C/E) & P(D/E) & P(E/E) \end{pmatrix}, \tag{C.9}$$

where the entries are now $2D_P$-step transition probabilities. All the quantities appearing in the previous expressions are elements of the matrices above. The matrix **P** can be determined experimentally, and from it, the other three matrices can be determined. Since **R** is lower in dimension than **P**, it is easier to calculate $\mathbf{R}^{D_P-1}$. In fact, if we let

$$\gamma = P(C \cup D/C \cup D) + P(E/E), \tag{C.10}$$

then

$$\mathbf{R}^m = \frac{1}{(2-\gamma)} \begin{pmatrix} P(C \cup D)/E & P(E/C \cup D) \\ P(C \cup D)/E & P(E/C \cup D) \end{pmatrix} + \frac{(\gamma-1)^m}{(2-\gamma)} \begin{pmatrix} P(E/C \cup D) & -P(E/C \cup D) \\ -P(C \cup D)/E & P(C \cup D/E) \end{pmatrix} \tag{C.11}$$

Therefore, all the quantities needed in determining the performance parameters can be found once **P** is known. The parameter γ is called the clustering coefficient [37], as it measures the tendency of transmissions of the same type to follow each other.

APPENDIX D: DECODING SELECTIONS AND THEIR EQUATIONS

This appendix gives the equations of the decoding selections from Table A.7 of Appendix A.

$S_1 = \{1,2,4\}$	$z_0 = c_0$
	$z_1 = c_{14}$
$(c_0, c_{14}) = \mathbf{P}(r_0, r_{15}, r_{14})$	$z_2 = c_0 + c_4 + c_9 + c_{14} + c_{17}$
$(c_{16}, c_{17}) = \mathbf{P}(r_{16}, r_1, r_{17})$	$z_3 = c_0 + c_9 + c_{16}$
	$z_4 = c_0 + c_3 + c_{14} + c_{17}$
$(c_3, c_4, c_9) = \mathbf{M}(r_{10}, r_3, r_{11}, r_4, r_5, r_9)$	$z_5 = c_0 + c_9 + c_{14} + c_{16} + c_{17}$
	$z_6 = c_0 + c_3 + c_4 + c_9 + c_{16}$

$S_2 = \{2,3,6\}$	$z_0 = c_2 + c_6 + c_7 + c_{17} + c_{18}$
	$z_1 = c_2 + c_{16} + c_{17} + c_{18}$
$(c_{16}, c_{17}) = \mathbf{P}(r_{16}, r_1, r_{17})$	$z_2 = c_2 + c_7 + c_{12} + c_{16} + c_{17}$
$(c_2, c_7) = \mathbf{P}(r_2, r_8, r_7)$	$z_3 = c_2 + c_{12} + c_{16} + c_{18}$
	$z_4 = c_{18}$
$(c_{18}, c_{12}, c_6) = \mathbf{M}(r_{20}, r_{18}, r_{19}, r_{12}, r_{13}, r_6)$	$z_5 = c_{12}$
	$z_6 = c_6$

$S_3 = \{1,2,6\}$	$z_0 = c_0$
	$z_1 = c_{14}$
$(c_0, c_{14}) = \mathrm{P}(r_0, r_{15}, r_{14})$	$z_2 = c_0 + c_6 + c_{16} + c_{18}$
$(c_{16}, c_{17}) = \mathrm{P}(r_{16}, r_1, r_{17})$	$z_3 = c_{12} + c_{14} + c_{18}$
	$z_4 = c_{18}$
$(c_{18}, c_{12}, c_6) = \mathrm{M}(r_{20}, r_{18}, r_{19}, r_{12}, r_{13}, r_6)$	$z_5 = c_{12}$
	$z_6 = c_6$

$S_4 = \{1,3,6\}$	$z_0 = c_0$
	$z_1 = c_{14}$
$(c_0, c_{14}) = \mathrm{P}(r_0, r_{15}, r_{14})$	$z_2 = c_7 + c_{12} + c_{14} + c_{18}$
$(c_2, c_7) = \mathrm{P}(r_2, r_8, r_7)$	$z_3 = c_0 + c_2 + c_6 + c_7 + c_{14} + c_{18}$
	$z_4 = c_{18}$
$(c_{18}, c_{12}, c_6) = \mathrm{M}(r_{20}, r_{18}, r_{19}, r_{12}, r_{13}, r_6)$	$z_5 = c_{12}$
	$z_6 = c_6$

$S_5 = \{1,3,4\}$	$z_0 = c_0$
	$z_1 = c_{14}$
$(c_0, c_{14}) = \mathrm{P}(r_0, r_{15}, r_{14})$	$z_2 = c_0 + c_2 + c_7 + c_9 + c_{14}$
$(c_2, c_7) = \mathrm{P}(r_2, r_8, r_7)$	$z_3 = c_2 + c_3 + c_9$
	$z_4 = c_2 + c_3 + c_4 + c_7$
$(c_3, c_4, c_9) = \mathrm{M}(r_{10}, r_3, r_{11}, r_4, r_5, r_9)$	$z_5 = c_0 + c_4 + c_7 + c_9$
	$z_6 = c_0 + c_2 + c_4 + c_9 + c_{14}$

$S_6 = \{2,3,4\}$	$z_0 = c_2 + c_3 + c_6$
	$z_1 = c_3 + c_4 + c_7 + c_{16} + c_{17}$
$(c_2, c_7) = \mathbf{P}(r_2, r_8, r_7)$	$z_2 = c_4 + c_9 + c_{17}$
$(c_{16}, c_{17}) = \mathbf{P}(r_{16}, r_1, r_{17})$	$z_3 = c_2 + c_3 + c_9$
	$z_4 = c_2 + c_4 + c_7$
$(c_3, c_4, c_9) = \mathbf{M}(r_{10}, r_3, r_{11}, r_4, r_5, r_9)$	$z_5 = c_2 + c_4 + c_7 + c_9 + c_{16}$
	$z_6 = c_2 + c_4 + c_9$

$S_7 = \{1,4,6\}$	$z_0 = c_0$
	$z_1 = c_{14}$
$(c_0, c_{14}) = \mathbf{P}(r_0, r_{15}, r_{14})$	$z_2 = c_0 + c_3 + c_6 + c_9 + c_{12}$
	$z_3 = c_0 + c_3 + c_{12} + c_{18}$
$(c_3, c_4, c_9) = \mathbf{M}(r_{10}, r_3, r_{11}, r_4, r_5, r_9)$	$z_4 = c_{18}$
$(c_{18}, c_{12}, c_6) = \mathbf{M}(r_{20}, r_{18}, r_{19}, r_{12}, r_{13}, r_6)$	$z_5 = c_{12}$
	$z_6 = c_6$

$S_8 = \{2,4,6\}$	$z_0 = c_2 + c_4 + c_6 + c_{12} + c_{17}$
	$z_1 = c_4 + c_6 + c_{12} + c_{18}$
$(c_{16}, c_{17}) = \mathbf{P}(r_{16}, r_1, r_{17})$	$z_2 = c_3 + c_4 + c_{12} + c_{17} + c_{18}$
	$z_3 = c_4 + c_6 + c_{17} + c_{18}$
$(c_3, c_4, c_9) = \mathbf{M}(r_{10}, r_3, r_{11}, r_4, r_5, r_9)$	$z_4 = c_{18}$
$(c_{18}, c_{12}, c_6) = \mathbf{M}(r_{20}, r_{18}, r_{19}, r_{12}, r_{13}, r_6)$	$z_5 = c_{12}$
	$z_6 = c_6$

$S_9 = \{3,4,6\}$	$z_0 = c_2 + c_9 + c_{12} + c_{18}$
	$z_1 = c_2 + c_3 + c_6 + c_7 + c_{12}$
$(c_2, c_7) = \mathbf{P}(r_2, r_7, r_8)$	$z_2 = c_2 + c_3 + c_6 + c_{18}$
	$z_3 = c_2 + c_3 + c_9$
$(c_3, c_4, c_9) = \mathbf{M}(r_{10}, r_3, r_{11}, r_4, r_5, r_9)$	$z_4 = c_{18}$
$(c_{18}, c_{12}, c_6) = \mathbf{M}(r_{20}, r_{18}, r_{19}, r_{12}, r_{13}, r_6)$	$z_5 = c_{12}$
	$z_6 = c_6$

$S_{10} = \{1,4,5\}$	$z_0 = c_0$
	$z_1 = c_{14}$
$(c_0, c_{14}) = \mathbf{P}(r_0, r_{15}, r_{14})$	$z_2 = c_4 + c_{14} + c_{23}$
	$z_3 = c_3 + c_4 + c_9 + c_{14} + c_{23}$
$(c_3, c_4, c_9) = \mathbf{M}(r_{10}, r_3, r_{11}, r_4, r_5, r_9)$	$z_4 = c_0 + c_3 + c_9 + c_{23}$
$(c_{21}, c_{22}, c_{23}) = \mathbf{M}(r_{24}, r_{21}, r_{26}, r_{22}, r_{25}, r_{23})$	$z_5 = c_3 + c_4 + c_{14} + c_{21}$
	$z_6 = c_0 + c_9 + c_{21} + c_{23}$

$S_{11} = \{1,2,5\}$	$z_0 = c_0$
	$z_1 = c_{14}$
$(c_0, c_{14}) = \mathbf{P}(r_0, r_{15}, r_{14})$	$z_2 = c_{14} + c_{16} + c_{21} + c_{22}$
$(c_{16}, c_{17}) = \mathbf{P}(r_{16}, r_1, r_{17})$	$z_3 = c_0 + c_{14} + c_{16} + c_{17} + c_{23}$
	$z_4 = c_{17} + c_{22}$
$(c_{21}, c_{22}, c_{23}) = \mathbf{M}(r_{24}, r_{21}, r_{26}, r_{22}, r_{25}, r_{23})$	$z_5 = c_0 + c_{16} + c_{23}$
	$z_6 = c_0 + c_{14} + c_{17} + c_{21}$

$S_{12}=\{1,3,5\}$	$z_0=c_0$
	$z_1=c_{14}$
$(c_0,c_{14})=\mathbf{P}(r_0,r_{15},r_{14})$	$z_2=c_2+c_{21}$
$(c_2,c_7)=\mathbf{P}(r_2,r_8,r_{17})$	$z_3=c_2+c_7+c_{21}+c_{22}$
	$z_4=c_0+c_7+c_{21}+c_{22}+c_{23}$
$(c_{21},c_{22},c_{23})=\mathbf{M}(r_{24},r_{21},r_{26},r_{22},r_{25},r_{23})$	$z_5=c_0+c_{14}+c_{22}+c_{23}$
	$z_6=c_7+c_{14}+c_{23}$

$S_{13}=\{2,4,5\}$	$z_0=c_3+c_4+c_9+c_{16}+c_{17}+c_{21}$
	$z_1=c_4+c_9+c_{16}+c_{17}+c_{21}$
$(c_{16},c_{17})=\mathbf{P}(r_{16},r_1,r_{17})$	$z_2=c_4+c_9+c_{17}$
	$z_3=c_3+c_4+c_{17}+c_{21}$
$(c_3,c_4,c_9)=\mathbf{M}(r_{10},r_3,r_{11},r_4,r_5,r_9)$	$z_4=c_{17}+c_{22}$
$(c_{21},c_{22},c_{23})=\mathbf{M}(r_{24},r_{21},r_{26},r_{22},r_{25},r_{23})$	$z_5=c_3+c_9+c_{16}+c_{17}+c_{22}$
	$z_6=c_3+c_{17}+c_{21}+c_{22}$

$S_{14}=\{1,5,6\}$	$z_0=c_0$
	$z_1=c_{14}$
$(c_0,c_{14})=\mathbf{P}(r_0,r_{15},r_{14})$	$z_2=c_{12}+c_{18}+c_{23}$
	$z_3=c_0+c_6+c_{12}+c_{21}$
$(c_{18},c_{12},c_6)=\mathbf{M}(r_{20},r_{18},r_{19},r_{12},r_{13},r_6)$	$z_4=c_{18}$
$(c_{21},c_{22},c_{23})=\mathbf{M}(r_{24},r_{21},r_{26},r_{22},r_{25},r_{23})$	$z_5=c_{12}$
	$z_6=c_6$

$S_{15} = \{2,5,6\}$	$z_0 = c_{12} + c_{16} + c_{23}$
	$z_1 = c_6 + c_{12} + c_{16} + c_{17} + c_{21} + c_{23}$
$(c_{16}, c_{17}) = \mathbf{P}(r_{16}, r_1, r_{17})$	$z_2 = c_6 + c_{12} + c_{18} + c_{23}$
	$z_3 = c_6 + c_{16} + c_{21} + c_{22}$
$(c_{18}, c_{12}, c_6) = \mathbf{M}(r_{20}, r_{18}, r_{19}, r_{12}, r_{13}, r_6)$	$z_4 = c_{18}$
$(c_{21}, c_{22}, c_{23}) = \mathbf{M}(r_{24}, r_{21}, r_{26}, r_{22}, r_{25}, r_{23})$	$z_5 = c_{12}$
	$z_6 = c_6$

$S_{16} = \{2,3,5\}$	$z_0 = c_7 + c_{17} + c_{21} + c_{23}$
	$z_1 = c_2 + c_{16} + c_{22}$
$(c_{16}, c_{17}) = \mathbf{P}(r_{16}, r_1, r_{17})$	$z_2 = c_2 + c_{21}$
$(c_2, c_7) = \mathbf{P}(r_2, r_8, r_{17})$	$z_3 = c_2 + c_7 + c_{22}$
	$z_4 = c_{17} + c_{22}$
$(c_{21}, c_{22}, c_{23}) = \mathbf{M}(r_{24}, r_{21}, r_{26}, r_{22}, r_{25}, r_{23})$	$z_5 = c_7 + c_{16} + c_{17} + c_{21}$
	$z_6 = c_2 + c_7 + c_{16} + c_{22} + c_{23}$

$S_{17} = \{3,4,5\}$	$z_0 = c_2 + c_4 + c_7 + c_9 + c_{23}$
	$z_1 = c_2 + c_4 + c_{21} + c_{23}$
$(c_2, c_7) = \mathbf{P}(r_2, r_8, r_{17})$	$z_2 = c_2 + c_{21}$
	$z_3 = c_2 + c_7 + c_{21} + c_{22}$
$(c_3, c_4, c_9) = \mathbf{M}(r_{10}, r_3, r_{11}, r_4, r_5, r_9)$	$z_4 = c_2 + c_4 + c_{21} + c_{22}$
$(c_{21}, c_{22}, c_{23}) = \mathbf{M}(r_{24}, r_{21}, r_{26}, r_{22}, r_{25}, r_{23})$	$z_5 = c_7 + c_{16} + c_{17} + c_{21}$
	$z_6 = c_2 + c_7 + c_{16} + c_{22} + c_{23}$

$S_{18} = \{3,5,6\}$	$z_0 = c_2 + c_6 + c_7 + c_{12} + c_{22}$
	$z_1 = c_6 + c_7 + c_{23}$
$(c_2, c_7) = \mathrm{P}(r_2, r_8, r_7)$	$z_2 = c_6 + c_{12} + c_{18} + c_{23}$
	$z_3 = c_6 + c_7 + c_{12} + c_{18} + c_{22} + c_{23}$
$(c_{18}, c_{12}, c_6) = \mathrm{M}(r_{20}, r_{18}, r_{19}, r_{12}, r_{13}, r_6)$	$z_4 = c_{18}$
$(c_{21}, c_{22}, c_{23}) = \mathrm{M}(r_{24}, r_{21}, r_{26}, r_{22}, r_{25}, r_{23})$	$z_5 = c_{12}$
	$z_6 = c_6$

$S_{19} = \{3,4,7\}$	$z_0 = c_3 + c_4 + c_7 + c_9 + c_{12}$
	$z_1 = c_2 + c_3 + c_6 + c_7 + c_{12}$
$(c_2, c_7) = \mathrm{P}(r_2, r_8, r_7)$	$z_2 = c_4 + c_6 + c_7$
	$z_3 = c_2 + c_3 + c_6 + c_9$
$(c_3, c_4, c_9) = \mathrm{M}(r_{10}, r_3, r_{11}, r_5, r_5, r_9)$	$z_4 = c_2 + c_3 + c_4 + c_6 + c_7$
	$z_5 = C_{12}$
	$z_6 = c_6$

D.1 Depth-3 Decoding

We introduce A_0, A_1, A_2, ..., A_{13} and A_{14} as follows:

$A_1 = \overline{P}_1\overline{P}_3\overline{P}_4\overline{P}_6\overline{P}_7$	$(S_1 \quad \ldots \quad S_9)$	$A_9 = P_1\overline{P}_3\overline{P}_4P_6\overline{P}_7$	(S_1)
$A_2 = P_1\overline{P}_3\overline{P}_4\overline{P}_6\overline{P}_7$	(S_1, S_3, S_7, S_8)	$A_{10} = P_1\overline{P}_3\overline{P}_4\overline{P}_6P_7$	(S_8)
$A_3 = \overline{P}_1P_3\overline{P}_4\overline{P}_6\overline{P}_7$	(S_4, S_5, S_7, S_9)	$A_{11} = \overline{P}_1P_3P_4\overline{P}_6\overline{P}_7$	(S_4)
$A_4 = \overline{P}_1\overline{P}_3P_4\overline{P}_6\overline{P}_7$	(S_2, S_3, S_4)	$A_{12} = \overline{P}_1P_3\overline{P}_4\overline{P}_6P_7$	(S_9)
$A_5 = \overline{P}_1\overline{P}_3\overline{P}_4P_6\overline{P}_7$	(S_1, S_5, S_6)	$A_{13} = \overline{P}_1P_3\overline{P}_4P_6\overline{P}_7$	(S_5)
$A_6 = \overline{P}_1\overline{P}_3\overline{P}_4\overline{P}_6P_7$	(S_2, S_6, S_8, S_9)	$A_{14} = \overline{P}_1\overline{P}_3P_4\overline{P}_6P_7$	(S_2)
$A_7 = P_1P_3\overline{P}_4\overline{P}_6\overline{P}_7$	$(S_7, \quad S_9)$	$A_{15} = \overline{P}_1\overline{P}_3\overline{P}_4P_6P_7$	(S_6)
$A_8 = P_1\overline{P}_3P_4\overline{P}_6\overline{P}_7$	(S_3)		

FAILURE = NOR(A_1, A_2, $A_3, \ldots, A_{14}$, A_{15})

D.2 Depth-4 Decoding

Here we have 37 options:

$A16=\overline{P}_1\overline{P}_3\overline{P}_4\overline{P}_5\overline{P}_6\overline{P}_7$

$A17=P_1\overline{P}_3\overline{P}_4\overline{P}_5\overline{P}_6\overline{P}_7$

$A18=\overline{P}_1P_3\overline{P}_4\overline{P}_5\overline{P}_6\overline{P}_7$

$A19=\overline{P}_1\overline{P}_3P_4\overline{P}_5\overline{P}_6\overline{P}_7$

$A20=\overline{P}_1\overline{P}_3\overline{P}_4P_5\overline{P}_6\overline{P}_7$

$A21=\overline{P}_1\overline{P}_3\overline{P}_4\overline{P}_5P_6\overline{P}_7$

$A22=\overline{P}_1\overline{P}_3\overline{P}_4\overline{P}_5\overline{P}_6P_7$

$A23=P_1P_3\overline{P}_4\overline{P}_5\overline{P}_6\overline{P}_7$

$A24=P_1\overline{P}_3P_4\overline{P}_5\overline{P}_6\overline{P}_7$

$A25=P_1\overline{P}_3\overline{P}_4P_5\overline{P}_6\overline{P}_7$

$A26=P_1\overline{P}_3\overline{P}_4\overline{P}_5P_6\overline{P}_7$

$A27=P_1\overline{P}_3\overline{P}_4\overline{P}_5\overline{P}_6P_7$

$A27=P_1\overline{P}_3\overline{P}_4\overline{P}_5\overline{P}_6P_7$

$A28=\overline{P}_1P_3P_4\overline{P}_5\overline{P}_6\overline{P}_7$

$A29=\overline{P}_1P_3\overline{P}_4P_5\overline{P}_6\overline{P}_7$

$A30=\overline{P}_1P_3\overline{P}_4\overline{P}_5P_6\overline{P}_7$

$A31=\overline{P}_1P_3\overline{P}_4\overline{P}_5\overline{P}_6P_7$

$A32=\overline{P}_1\overline{P}_3P_4P_5\overline{P}_6\overline{P}_7$

$A33=\overline{P}_1\overline{P}_3P_4\overline{P}_5P_6\overline{P}_7$

$A34=\overline{P}_1\overline{P}_3P_4\overline{P}_5\overline{P}_6P_7$

$A35=\overline{P}_1\overline{P}_3\overline{P}_4P_5P_6\overline{P}_7$

$A36=\overline{P}_1\overline{P}_3\overline{P}_4P_5\overline{P}_6P_7$

$A37=\overline{P}_1\overline{P}_3\overline{P}_4\overline{P}_5P_6P_7$

$A38=P_1P_3P_4\overline{P}_5\overline{P}_6\overline{P}_7$

$A39=P_1P_3\overline{P}_4P_5\overline{P}_6\overline{P}_7$

$A40=P_1P_3\overline{P}_4\overline{P}_5P_6\overline{P}_7$

$A41=P_1\overline{P}_3P_4P_5\overline{P}_6\overline{P}_7$

$A42=P_1\overline{P}_3P_4\overline{P}_5P_6\overline{P}_7$

$A43=P_1\overline{P}_3P_4\overline{P}_5\overline{P}_6P_7$

$A44=P_1\overline{P}_3\overline{P}_4P_5P_6\overline{P}_7$

$A45=P_1\overline{P}_3\overline{P}_4P_5\overline{P}_6P_7$

$A46=\overline{P}_1P_3P_4P_5\overline{P}_6\overline{P}_7$

$A47=\overline{P}_1P_3P_4\overline{P}_5P_6\overline{P}_7$

$A48=\overline{P}_1P_3P_4\overline{P}_5\overline{P}_6P_7$

$A49=\overline{P}_1P_3\overline{P}_4P_5P_6\overline{P}_7$

$A50=\overline{P}_1P_3\overline{P}_4P_5\overline{P}_6P_7$

$A51=\overline{P}_1P_3\overline{P}_4\overline{P}_5P_6P_7$

$A52=\overline{P}_1\overline{P}_3P_4P_5\overline{P}_6P_7$

FAILURE = NOR($A16$, $A17$, $A18$,$A19$, …, $A51$, $A52$)

The selections are made according to the following list.

$A16$ $S1$, $S2$, ..., $S18$

$A17$ $S1$, $S3$, $S7$, $S8$, $S10$, $S11$, $S14$, $S15$

$A18$ $S4$, $S5$, $S7$, $S9$, $S10$, $S12$, $S14$, $S17$, $S18$

$A19$ $S2$, $S3$, $S4$, $S11$, $S12$, $S13$, $S15$, $S16$, $S18$

$A20$ $S1$, $S2$,..., $S9$, $S14$

$A21$ $S1$, $S5$, $S6$, $S10$, $S11$, $S12$, $S13$, $S16$, $S17$

$A22$ $S2$, $S6$, $S8$, $S9$, $S13$, $S15$, $S16$, $S17$, $S18$

$A23$	$S7, S9, S10, S14$	$A33$	$S11, S12, S13, S16$	$A43$	$S15$
$A24$	$S3, S11, S14, S15$	$A34$	$S2, S13, S15, S16, S18$	$A44$	$S1$
$A25$	$S1, S3, S7, S8, S14$	$A35$	$S1, S5, S6$	$A45$	$S8$
$A26$	$S1, S10, S11$	$A36$	$S2, S6, S8$	$A46$	$S4$
$A27$	$S8, S15$	$A37$	$S6, S13, S16, S17$	$A47$	$S18$
$A28$	$S4, S12, S14, 18$	$A38$	$S1$	$A48$	$S5$
$A29$	$S4, S5, S7, S9, S14$	$A39$	$S14$	$A49$	$S9$
$A30$	$S5, S10, S12, S17$	$A40$	$S10$	$A50$	$S17$
$A31$	$S9, S17, S18$	$A41$	$S3$	$A51$	$S2$
$A32$	$S2, S3, S4$	$A42$	$S11$	$A52$	$S13, S16$

REFERENCES

[1] Shannon, C. E., "A Mathematical Theory of Communication", *Bell System Technical Journal (BSTJ)*, Vol.27, 1948, pp.379-423 (Part I), and pp.623-656 (Prt II).

[2] Viterbi, A. J., "Convolutional Codes and Their Performance in Communication Systems", *IEEE Transactions on Communication Technology*, Vol. COM-19 No.10, pp.751-772, October 1971.

[3] Blahut, R. E., *Principles and Practice of Information Transmission*, Addison-Wesley, Reading, MA, 1987.

[4] Turin, W. *Performance Analysis of Digital Transmission Systems*, Computer Science Press, 1990

[5] Lin, S., and P. S. Yiu, "A Hybrid ARQ Scheme with Parity Retransmission For Error Control of Satellite Channels", *IEEE Transactions on Communications*, Vol. COM-30 No.7, pp.1700-1719, July 1982.

[6] Lin, S., D. J. Costello and M. J. Miller, "A Survey of Various ARQ and Hybrid ARQ Schemes, and Error Detection Using Block Codes", *IEEE Communications Magazine*, Vol.22, No.12, December 1984.

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

[8] Krishna, H., and S. D. Morgera, "A New Error Control Scheme For Hybrid ARQ Schemes", *IEEE Transactions on Communications* Vol. COM-35 No.10, pp.981-990, October 1987.

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

[10] Forney, G. D., "Burst Correcting Codes For the Classic Bursty Channel," *IEEE Transactions on Communication Technology*, Vol. COM-19, pp.772-781, 1971.

[11] Tong, S. Y., "Burst Trapping Techniques For a Compund Error Channel", *IEEE Transactions on Inf. Theory*, Vol. IT-15, pp.710-715, 1969.

[12] Sullivan, D. D., "A Generalization of Gallager's Adaptive Error Control Scheme", *IEEE Transactions on Inf. Theory*, Vol. IT-17 pp.727-735, 1971.

[13] Burton, H. O., D. D. Sullivan, and S. Y. Tong, "Generalized Burst-Trapping Codes," *IEEE Transactions on Inf. Theory*, Vol. IT-17 No.11, pp.736-742, Nov. 1971.

[14] Sastry, A. R. K., and L. N. Kanal, "Hybrid Error Control Using Retransmission and Generalized Burst-Trapping Codes," *IEEE Transactions on Communications*, Vol. COM-22 No.4, pp.385-393, April 1976.

[15] Michelson, A. M., and A. H. Levesque, *Error-Control Techniques For Digital Communications*, John Wiley & Sons 1985.

[16] Tsai, S. "Markov Characterization of the HF Channel", *IEEE Transactions on Communications Technology* Vol. COM-17 No.1, pp.24-32, February 1969.

[17] MacManamon, P. "HF Markov Models and Measured Error Averages", *IEEE Transactions on Communications Technology* Vol. COM-18 No.3, pp.201-208, June 1970.

[18] Bhargava, V. K., D. Haccoun, R. Matyas, and P. P. Nuspl, *Digital Communications By Satellite*, John Wiley & Sons 1981.

[19] Young, P. H., *Electronic Communication Techniques, 2nd Edition*, Merrill Publishing, 1990.

[20] Inglis, A. F., *Electronic Communications Handbook*, MicGraw-Hill, 1988

[21] Lavoie, P., D. Haccoun, and Y. Savaria, "New VLSI Architectures for Soft-Decision Threshold Decoders", *IEEE Transactions on Communications*, Vol. COM-39 No.2, pp.200-207, February 1991.

[22] Haccoun, D., P. Lavoie, and Y. Savaria, "New Architectures for Fast Convolutional Encoders and Threshold Decoders", *IEEE Journal of Selected Areas in Communications*, Vol. SAC-6 No.3, pp.547-557, April 1988.

[23] Kanal, L. N., and A. R. Sastry, "Models of Channels with Memory and Their Application to Error Control" *IEEE Proceedings* Vol. 66 No.7, pp.724-744, July 1978.

[24] Casti, J. L., *Alternate Realities: Mathematical Models of Nature and Man*, John Wiley & Sons 1989.

[25] Drukarev, A. I., and K. P. Yiu, "Performance of Error Correcting Codes on Channels with Memory", *IEEE Transactions on Communications* Vol. COM-34 No.6, pp.513-521, June 1986.

[26] Knowles, M. D., and A. I. Drukarev, "Bit Error Rate Estimation For Channels with Memory" *IEEE Transactions on Communications* Vol. COM-36 No.6, pp.767-769, June 1988.

[27] Fritchman, B. D., "A Binary Channel Characterization Using Partitioned Markov Chains", *IEEE Transactions on Information Theory* Vol. IT-13 No.3, pp.221-227, April 1967.

[28] Vucetic, B. "An Adaptive Coding Scheme For Time-Varying Channels", *IEEE Transactions on Communications*, Vol. COM-39 No.5, May 1991.

[29] Chien, R. T., A. Haddad, B. Goldberg, and E. Moyes, "An Analytic Error Model For Real Channels", *IEEE International Conference on Communications (ICC)* Conf. Rec. pp.15-7—15-12.

[30] Haddad, A., S. Tsai, B. Goldberg, and G. Raniery, "Markov Gap Models for Real Communication Channels", *IEEE Transactions on Communications* Vol. COM-23 No.11, pp.1189-1197, Nov. 1975.

[31] Benelli, G., "An ARQ Scheme with Memory and Integrated Modulation", *IEEE Transactions on Communications* Vol. COM-35 No.7, pp.689-697, July 1987.

[32] Kasami, T., T. Tokata, T. Fujiwara, and S. Lin, "A Concatenated Coded Modulation Scheme for Error Control", *IEEE Transactions on Communications*, Vol. COM-38 No.6, pp.752-763, June 1990.

[33] Brilliant, M. B., "Observations of Errors and Error Rates on T1 Digital Repeatered Lines", *Bell System Technical Journal (BSTJ)*, Vol.57, No.3, pp.711-746, March 1978.

[34] Gilbert, E. N., "Capacity of Burst-Noise Channels", *Bell System Technical Journal (BSTJ)*, Vol.39, No.9, pp.1253-1266, Sept. 1960

[35] Elliott, E. O., "A Model For the Switched Telephone Network For Data Communications" *Bell System Technical Journal (BSTJ)*, Vol.44, No.1, pp.89-119, January 1965.

[36] Elliott, E. O., "Estimates of Error Rates for Codes on Burst-Noise Channels", *Bell System Technical Journal (BSTJ)*, Vol.42, No.9, pp.1977-1997, Sept. 1963.

[37] Alexander, A. A., R. M. Gryb, and D. W. Nast, "Capabilities of the Telephone Network For Data Transmission", *Bell System Technical Journal (BSTJ)*, Vol.39, No.3, pp.431-447, 1960.

[38] Leung, C. H. C., Y. Kikumoto, and S. A. Sørensen, "The Throughput Efficiency of the Go-Back-N ARQ Scheme Under Markov and Related Error Structures", *IEEE Transactions on Communications*, Vol. COM-36 No.2, Febraury 1988.

[39] Meyer, M. E., and A. R. K. Sastry, "Performance Analysis of Tandem Burst Links with Applications to ISDN", *IEEE Transactions on Communications*, Vol. COM-33 No.7, pp.646-653, July 1985.

[40] Lai, W. S., "End-to-end Probability of Undetected Error in Computer Networks", *IEEE Transactions on Communications*, Vol. COM-34 No.3, March 1986

[41] Yaglom, A. M., *An Introduction to the Theory of Stationary Random Functions*, Prentice-Hall, Englewood Cliffs, 1962.

[42] Adoul, J. P. A., "Error Intervals and Cluster Density in Channel Modeling", *IEEE Transactions on Information Theory*, Vol.IT-20, pp.125-129, January 1974.

[43] Kosmopoulos, S. A., P. T. Mathiopoulos, and Maria D. Gouta, "Fourier-Bessel Error Performance Analysis and Evaluation of M-ary QAM Schemes in an Impulsive Noise Environment", *IEEE Transactions on Communications*, Vol. COM-39 No.3, pp.398-404, March 1991.

[44] Kosmopoulos, S. A., Maria D. Gouta, and P. T. Mathiopoulos, "Performance Evaluation of M-ary QPRS Schemes in Impulsive Noise Environments", *IEEE Transactions on Communications*, Vol. COM-39 No.3, pp.405-408, March 1991.

[45] Yamamoto, Y., and T. Wright, "Error Performance of the Evolving Digital Networks Including ISDNs", *IEEE Communications Magazine*, Vol. 27 No.4, pp.12-18, April 1989.

[46] Martins, J. A. C., J. C. Alves, "ARQ Protocols with Adaptive Block Size Perform Better Over a Wide Range of Bit Error Rates", *IEEE Transactions on Communications*, Vol. COM-38 No.6, pp.737-739, June 1990.

[47] Schwartz, M., *Telecommunication Networks: Protocols, Modeling, and Analysis*, Addison-Wesley, Reading, MA, 1987.

[48] Metzner, J. J., "Improved Coding Strategies For Meteor Burst Communications", *IEEE Transactions on Communications*, Vol. COM-38 No.2, pp.133-136, February 1990.

[49] Marriott, C., V.K. Oduol, B.P. Tailor, and A. K. Ghosh, "Meteor Burst Communications in the Post 2000 Era", *IEEE Military Communications Conference, MILCOM 90.*

[50] Mui, S., Y., "Coding for Meteor Burst Communications", *IEEE Transactions on Communications*, Vol. COM-39 No.5, May 1991.

[51] Pursley, M. B., and S. D. Sandberg, "Incremental Redundancy Transmission For Meteor Burst Communications", *IEEE Transactions on Communications*, Vol. COM-39 No.5, May 1991.

[52] Fantacci, R., "Performance Evaluation of Some ARQ Schemes Using Efficient Modulation Techniques and Noncoherent Detection", *IEEE Transactions on Communications*, Vol. COM-39 No.3, pp.445-451 March 1991.

[53] Morgera, S. D., and V. K. Oduol, "Soft-Decision Decoding Applied to The Generalized Type-II Hybrid ARQ Scheme", *IEEE Transactions on Communications*, Vol. COM-37 No.4, pp.393-396, April 1989.

[54] Lau, C., and C. Leung, "Performance Analysis of a Memory ARQ with Soft-Decision Detectors", *IEEE Transactions on Communications*, Vol. COM-34 No.8, pp.827-832, August 1986.

[55] Lavoie, P., D. Haccoun, and Y. Savaria, "Performance Analysis of a Memory ARQ with soft-Decision Detectors", *IEEE Transactions on Communications*, Vol. COM-39 No.2, February 1991.

[56] Fantacci, R., "Performance Evaluation of Efficient Continuous ARQ Protocols", *IEEE Transactions on Communications*, Vol. COM-38 No.6, pp.773-781 June 1990.

[57] Chase, D., "Code Combining—— A Maximum Likelihood Decoding Approach For Combining an Arbitrary Number of Packets", *IEEE Transactions on Communications*, Vol. COM-33 No.5, pp.385-393, May 1985.

[58] Chase, D., P. D. Mullers, and J. K. Wolf, "Applications of Code Combining to a Selective Repeat ARQ link", *Proceeding of the IEEE MILCOM*, Conf. Rec. Vol. 1 pp.247-252, 1985.

[59] Kallel, S., "Analysis of Type-II ARQ Scheme with Code Combining", *IEEE Transactions on Communications*, Vol. COM-38 No.8, pp.1133-1137, August 1990.

[60] Kallel, S., and D. Haccoun, "Sequential Decoding with ARQ and Code Combining: A Robust Hybrid FEC/ARQ System", *IEEE Transactions on Communications*, Vol. COM-38 No.7, pp.773-780, July 1990.

[61] Vucetic, B., D. Drajic, and D. Perisic, "Algorithms for Adaptive Error Control System Synthesis", *IEE Proceedings F* Vol. 135 No.1, February 1988.

[62] Drajic, D., and B. Vucetic, "Evaluation of Hybrid Error Control Systems", *IEE Proceedings F* Vol.131, No.2, pp.183-193, April 1984.

[63] Feldman, P. M., and V. K. Li, "An Analysis of Hybrid ARQ Protocol for Continuous-Time Markov Channels" *IEEE Proceedings of the MILCOM Conference* 1987.

[64] Feldman, P. M., and V. K. Li, "Delay Analysis For Adaptive Hybrid ARQ Protocol" *IEEE Proceedings of the MILCOM Conference* 1987.

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

[66] Du, J., M. Kasahara, and T. Namekawa, "Separable Codes on Type-II Hybrid ARQ Systems", *IEEE Transactions on Communications*, Vol. COM-36 No.10, pp.1089-1097, October 1988

[67] Shiozaki, A., K. Okuno, and T. Segawa, "A Hybrid ARQ Scheme with Adaptive Forward Error Correction for Satellite Communications", *IEEE Transactions on Communications*, Vol. COM-39 No.4, pp.482-484, April 1991.

[68] Metzner, J. J., "Improvements in Block Retransmission Schemes", *IEEE Transactions on Communications*, Vol. COM-27 No.2, pp.525-532, February 1979.

[69] Wicker, S., "Adaptive Rate Error Control Through the Use of Diversity Combining and Majority-Logic Decoding in a Hybrid ARQ Protocol", *IEEE Transactions on Communications*, Vol. COM-39 No.3, pp.380-385, March 1991.

[70] Kallel, S., and D. Haccoun, "Generalized Type-II Hybrid ARQ Scheme Using Punctured Convolutional Codes", *IEEE Transactions on Communications*, Vol. COM-38 No.11, pp.1938-1947, November 1990.

[71] Hagenauer, J., "Rate-Compatible Convolutional Codes and Their Applications", *IEEE Transactions on Communications*, Vol. COM-36 No.4, pp.389-400, April 1988

[72] Hagenauer, J., "Hybrid ARQ/FEC Protocols on Fading Channels Using Rate-Compatible Convolutional Codes", *IEEE International Conference on Communications*, Proc. ICC'87, 1987.

[73] Kallel, S., and D. Haccoun, "Sequential Decoding with an Efficient Partial Retransmission ARQ Strategy", *IEEE Transactions on Communications*, Vol. COM-39, No.2, pp.208-213, February 1991.

[74] Forney, G. D., *Concatenated Codes*, Research Monograph No.37, MIT Press, Cambridge, 1966.

[75] Mokrani, K., and S. S. Solimani, "Concatenated Codes Over Fading and Dispersive Channels", *IEEE International Conference on Communications*, Proc. ICC'89, pp.1378-1382, (45.5.1-45.5.5) 1989.

[76] Clark, G. C., and J. B. Cain, *Error-Correction Codes For Digital Communications*, Plenum Press 1983.

[77] Paaske, E., "Improved Decoding for a Concatenated Coding System Recommended by CCSDS", *IEEE Transactions on Communications*, Vol. COM-38 No.8, pp.1138-1144, August 1990.

[78] Kasami, T., T. Fujiwara, and S. Lin, "A Concatenated Coding Scheme For Error Control", *IEEE Transactions on Communications*, Vol. COM-34 No.5, pp.481-488, May 1986.

[79] Kløve, T., and M. Miller, "The Detection of Errors After Error Correction Decoding", *IEEE Transactions on Communications*, Vol. COM-32 No.5, pp.511-517, May 1985.

[80] Deng, R. H., and D. J. Costello, "Reliability and Throughput Analysis of a Concatenated Coding Scheme", *IEEE Transactions on Communications*, Vol. COM-35 No.7, pp.697-705, July 1987.

[81] Hayes, J. F., *Modeling and Analysis of Computer Communication Networks*, Plenum Press, 1984.

[82] Benice, R. J., and A. H. Frey, "An Analysis of Block Retransmission Systems", *IEEE Transactions on Communications Technology*, Vol. COM-12 No.12, pp.135-145, December 1964.

[83] Anagnostou, M. E., and E. N. Protonotarios, "Performance Analysis of the Selective Repeat ARQ Protocol", *IEEE Transactions on Communications*, Vol. COM-34 No.2, pp.127-135, February 1986.

[84] Metzner, J. J., and D. Chang, "Efficient Selective Repeat ARQ Strategies For Very Noisy and Fluctuating Channels", *IEEE Transactions on Communications*, Vol. COM-33 No.5, pp.409-415, May 1985.

[85] Miller, M. J., "The Analysis of Some Selective Repeat ARQ Schemes with Finite Receiver Buffer", *IEEE Transactions on Communications*, Vol. COM-29 No.5, pp.1307-1315, May 1981.

[86] Weldon, E. J., "An Improved ARQ Strategy" *IEEE Transactions on Communications*, Vol. COM-32 No.7, pp.480-486, July 1982.

[87] Wang, Y. M., and S. Lin, "A Modified Selective Type-II Hybrid ARQ System and Its Performance Analysis", *IEEE Transactions on Communications*, Vol. COM-31 No.5, pp.593-607, May 1983.

[88] Dravida, S., M. J. Master, and C. H. Morton, "A Method to Analyze Performance of Digital Connections", *IEEE Transactions on Communications*, Vol. COM-36 No.3, pp.298-305, March 1988.

[89] Haykin, S., *Adaptive Filter Theory*, Prentice-Hall, Englewood Cliffs, 1986.

[90] Giordano, A. A., *Least Squares Estimation with Applications to Digital Signal Processing*, John Wiley & Sons, 1985, pp.63-66.

[91] Levisnon, N. "The Wiener RMS (root-mean-square) error criterion in filter design and prediction," *J. Math Phys.*, vol.25, pp.261-278, 1947

[92] Durbin, J., "The Fitting of Time Series Models", *Rev. Intern. Statist. Inst.*, Vol.28, pp.233-244, 1960.

[93] Schwartz, L., *Principles of Coding, Filtering, and Information Theory*, Spartan Books, 1963.

[94] Oduol, V. K., *A Generalized Type-II Hybrid ARQ Scheme with Soft-Decision Decoding*, M. Eng Thesis, McGill University, March 1987.

[95] Oduol, V.K., and S. D. Morgera, "Performance Evaluation of the GH-II ARQ Scheme with Noisy Feedback", accepted for publication in *The IEEE Transactions on Communications*.