This is to certify that the

thesis entitled
"THROUGHPUT MAXIMIZED PARTIAL RECOVERY CODES"

presented by
SHIRISH S KARANDE

has been accepted towards fulfillment
of the requirements for

MASTER's degree in ELECTRICAL ENGINEERING

DR. HAYDER RADHA
Major professor

Date 05/07/2003

THROUGHPUT MAXIMIZED

PARTIAL RECOVERY CODES

By

SHIRISH S KARANDE

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

2003

ABSTRACT

THROUGHPUT MAXIMIZED PARTIAL RECOVERY CODES

By

Shirish S Karande

In this thesis, we design and optimize codes specifically for real-time multimedia communication over packet-based erasure channels. Based on the constraints and flexibilities of real time applications, we define a performance measure, *message throughput* (τ_m), which is suitable for these applications. We introduce the concept of "*partial recovery*" and investigate the interplay of optimal coding density and channel capacity. Based on this analysis, we introduce a new family of linear block codes, which we refer to as Partial Reed Solomon (PRS) Codes. These codes combine the advantages of lowering the density of a code for near capacity performance with the high decoding efficiency of Reed Solomon (RS) codes. Then, we demonstrate, through an example of a Binary Erasure Channel (BEC), that at near-capacity coding rates, appropriate design of a PRS code can outperform an RS-code. Moreover, as compared with RS codes, the proposed PRS codes provide a significantly improved graceful degradation. This is a highly desirable feature for real-time multimedia applications. Our video simulation results outline that the enhanced erasure recovery yields a profound improvement in the perceived media quality. Finally we investigate the performance of the dividend rendered by PRS codes operating above channel capacity. In particular we define a paradigm for a unique "fixed rate" adaptive FEC scheme based on PRS codes.

To the loving memory of my father and my grandmothers.

ACKNOWLEDGMENTS

"For every one of us that succeeds, it's because there's somebody there to show you the way out." ~ Oprah Winfrey

My advisor Dr. Hayder Radha for his continuous support, his understanding, his patience and tremendous research insight. I thank him many times for taking me under his wings and grooming me as a sincere and able student of science. None of the work in this thesis would have been possible without his encouragement and guidance. The influence of his teachings easily goes beyond the boundaries of pure academics.

My committee members Dr. Pierre and Dr. Hall for their research input and technical evaluation of this work. Dr. Hall for his invaluable guidance in developing a mathematical background in coding theory.

My brother and mother for their love, appreciation and motivation throughout my life. Almost any choice of words to express my gratitude towards them would be an understatement.

Syed Ali Khayam for many things but most of all helping me justify the utility of a nocturnal schedule. Mujahid for cooking the many meals that made focusing on research so much easier. Sunder and Shrini for always being there in times of trouble. Akshay for a number of discussions on Graph Codes.

Mukta for bearing the worst effects of my research frustrations and Ratna for listening patiently to my whining. My roommates and friends Kantha, Prabodh, Gowda, Ranjeet and Vasant who make my life so much smoother. My relatives and especially fami-

lies of Ashwini, Preeti and Bharat Uncle who made settling in United States a much easier task. Aparna, Sharadha, Irti for being great colleagues and friends. Keshav, Parry, Pushkar, Viks for being a big help at varied stages of this work. Last but not the least my colleagues in WAVES who make working there a very pleasurable experience.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The past decade has witnessed a rapid convergence of the telecommunication and entertainment industries. This led to a wide range of WEB-based TV-like services. Consequently, the demand for audiovisual distribution of entertainment content has increased exponentially. Applications such as audio/video telephony over the Internet have been conceptualized, implemented and are in high demand. Concurrent to this Internet-based growth, wireless has emerged as one of the fastest growing sectors of the telecommunication Industry. With the advent of third generation wireless devices, broadband wireless networks have become a reality. All these factors have contributed to the growing interest in wireless multimedia communication.

In a typical television network, real-time data from a single transmitter is delivered to multiple end receivers simultaneously. In order to enable similar service over the Internet, it is required that multimedia data be transmitted real-time over multicast networks. Traditional ARQ strategies, which were originally designed for data transmission over unicast computer networks, have been deemed unsuitable for multicast networks because the transmitter will be overwhelmed by feedback. Furthermore, the Internet telephony applications require protocols with low latency, which again render ARQ strategies ineffectual. Therefore, the impairments in a wireless channel and the need for a minimum Quality of Service guarantee have motivated research into possible alternatives to facilitate

reliable multicast transmission. Forward Error Correction (FEC) schemes serve as one such alternative. Rizzo [1] showed how FEC could be used over multicast networks to facilitate reliable transmission. Thus, design of efficient FEC schemes suitable for real time wired and wireless multimedia communication has become an important area of research.

The problem of designing an efficient error control schemes is almost as old as the field of information theory. The efficiency of an error control scheme is bounded by the channel characteristics. In 1948 Shannon [2] showed that the rate at which information can be reliably transferred over a communication channel is bounded by the channel capacity (denoted by C). Over the past 50 years numerous attempts have been made to design block codes which could achieve this bound. For example, [3],[4],[5], can serve as some well-known contributions to this field.

A block code is usually characterized by three parameters:

- The number of source (or message) symbols K that are transmitted in a coded block.
- The size of the coded block N, which is the total number of symbols in the block. Therefore, the number of redundant symbols, also known as parity symbols, is $N - K$.
- The rate of the code $R = K/N$.

Consequently, in attempting to construct good block codes, the major parameters of interest have been the probability of block decoding error, denoted by P_e, the block

length N, and the rate R. It is widely accepted that for a *given* N and R, the least value of P_e can be obtained by using a (N, K) Reed Solomon (RS) code [6], where $K = N \cdot R$ is the number of message symbols (as mentioned above). RS codes have become very popular for packet loss recovery over packet networks, in general, and the Internet in particular (i.e., over *erasure*[1] channels). In general the process of encoding and decoding for erasure channels can be described by **Figure 1**.

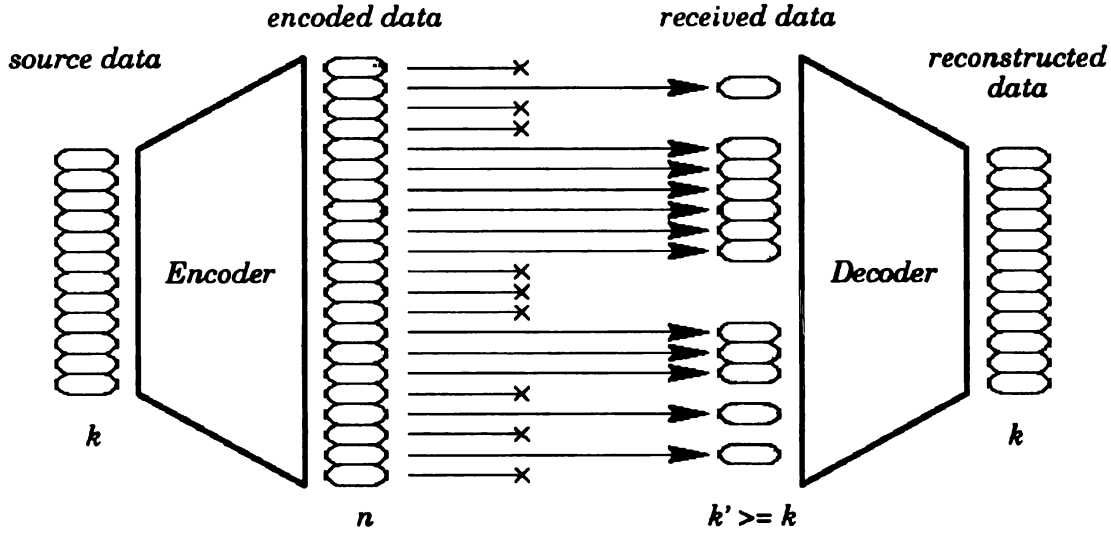


Figure 1 A Graphical representation of the encoding/decoding process [1].

In this thesis we introduce a new approach for design of FEC schemes suitable for real time delivery of multimedia over heterogeneous networks. We identify the limitations of some of the current RS based FEC schemes and show how our proposed "Partial

[1] A channel erasure is an error for which the position of the error is known but the magnitude is not. An erasure in a position i of a received N-tuple can occur as a result of the decoder receiving enough information to decide on a symbol for that coordinate, or information indicating that a particular coordinate value is unreliable. The task of the decoder here is to restore or "fill" the erasure position [7].

Recovery Codes" based scheme can provide improved reliability under certain network conditions. As most FEC schemes at the application layer treat a packet drop as an erasure, the work in this thesis also focuses on erasure channels.

In principle, one can classify the type of applications that employ linear block codes into real-time and non real-time. Each of these application types has its own requirements, constraints, and also flexibilities that can be exploited for a successful deployment of block codes over erasure channels. For example, a powerful and successful usage of the flexibilities and requirements of non real-time applications that demand a reliable transmission of large data files to a large number of receivers has resulted in the recently developed digital fountain approach [8],[9],[10]. This approach provides reliable multicast transmission of a given data file with K message symbols by encoding the desired file into a large-size block of N symbols while requiring the receiver to only acquire a very small overhead symbols (i.e., beyond the minimum required K message symbols). The digital fountain framework is able to achieve this reliable transmission to a very large number of unsynchronized receivers and without feedback while maintaining a very low computational decoding complexity when compared with a similar size RS codes.

The majority of recent proposals for the recovery of lost packets encountered in *real-time* multicast and unicast applications are based on traditional RS codes (e.g., [11],[12]). Some of these approaches are based on employing feedback information regarding the channel condition in realtime [13], [14]. Meanwhile, there are several key requirements and flexibilities imposed/provided by realtime applications that have not been fully considered/utilized when designing block codes that are optimized for these applications. Under this project, we will design and develop new linear block codes that take into con-

sideration key requirements and flexibilities of multimedia applications, in general, and realtime compressed video transmission in particular.

Before proceeding, we outline some of the key aspects of realtime applications and related challenges that motivated the proposed work.

- A fundamental requirement of any realtime application is the transmission of message data at a minimum desired rate R. In general, this minimum rate should be maintained to achieve a certain quality. The minimum rate requirement translates to the transmission of a minimum number of K message symbols within an N-symbol code block: $R=K/N$. Consequently, one of the constraints in the design of linear block codes for realtime applications is the usage of a maximum number $(N-K)$ of parity symbols within the N-symbol block.
- In general, the performance of linear block codes improves with larger values of the code block size N. However, realtime applications can employ a maximum number N depending on the particular application. For example, non-interactive multimedia streaming applications can use larger values of N than interactive (e.g., telephony) applications. Either case, there is a maximum number for the code block size N that needs to be adhered to. Therefore, unlike non-realtime applications that may have the flexibility in selecting N and $R=K/N$, realtime applications, in general, have to employ (adhere to) a block code with a pair-constraint (N, K).
- Performance criteria for LBC codes, which are used for non-realtime data, are not always suitable for realtime applications. For example, a non-realtime LBC code can be evaluated based on the number of symbols needed to *perfectly* recover all

of the original message symbols. In general, for realtime applications, *perfect* recovery, and consequently perfect reconstruction, of the original message symbols is not a hard requirement (as explained further below). Meanwhile, it is crucial to deliver the realtime application layer with the maximum number of the message symbols that are transmitted by the system. Therefore, the probability of a *message* symbol loss (*after* channel decoding) is a key performance parameter. We denote to this probability by p_m. Hence, the parameter $\tau_m=(1-p_m)$, which represents the probability of receiving a message symbol by the realtime application (after channel decoding), is a measure of the end-to-end message symbol throughput. One of the key objectives of the LBC codes to be developed under this proposal is to maximize this throughput measure τ_m. (For the remainder of this proposal, we will refer to τ_m as the *message throughput*.)

- Based on a variety of multimedia processing and compression techniques, a wide range of practical application-layer error concealment methods can be used to compensate for lost data [15],[16],[17],[18],[19]. These techniques, however, usually work well only when the number of losses is limited to a small number of uncovered data. In other words, practical multimedia error concealment and resilience methods usually become useless when the number of losses is beyond a certain threshold. Consequently, it is very crucial for LBC codes to perform well when the number of lost message symbols is large by recovering the majority of these lost message symbols. Meanwhile, and although it is desirable, it is less crucial for these codes to provide perfect recovery when the number of losses (before or after channel decoding) is very small (e.g., one or few symbols) due to the ma-

turity of powerful multimedia processing techniques. Therefore, codes that maintain very low end-to-end (effective) message losses are more desirable than codes that provide perfect recovery under good channel conditions (e.g., under very low loss probability) but provide low recovery under adverse channel conditions. This desirable feature highlights one of the key problems with current LBC codes that are used widely for realtime video. It is well known, for example, that when a RS code block experience a number of losses that is larger than the number of parity symbols, then the code is incapable of recovering any of the lost message data. Experiencing a number of losses that is larger than the number of parity symbols is quite feasible over channels with time-varying characteristics (e.g., the Internet [20], [21] and wireless networks [22], [23]), even if, "on average", the message rate R is lower than the channel capacity. This is particularly true when the message rate R is close to (but may still be lower than) the channel capacity. Moreover, and due to (a) the large amount of data that is inherently needed for representing multimedia (in particular video) signals, and/or (b) the compressed representation of these signals are normally encoded ahead of time at a certain rate that cannot be reduced in realtime by the sender, it is quite often when multimedia applications operate very closely to channel capacity. This phenomenon is quite common for a wide range of applications such as popular streaming applications on the web, IP multimedia telephony, and multicast video.

Consequently, one of the main objectives of the proposed work is the design of LBC codes that are capable of achieving high message throughput τ_m when the rate is close to (but still lower than) channel capacity and when the number of losses L exceeds the num-

ber of parity symbols (N–K). Unlike traditional RS codes, which exhibit a very sharp degradation in their ability to recover lost packets around the point (L=N–K), the proposed LBC codes will provide a graceful transition in their lost-message-recovery capabilities while maintaining a very high message throughput τ_m over this transition point and beyond.

The rest of the thesis is organized as follows:

In Chapter 2 of this thesis we give a brief introduction to coding theory. We elaborate the equivalence between solving a system of equations and decoding for erasures. An overview of some of the basic concepts of graph theory is also provided. This facilitates a discussion on the subject of Graph Codes and decoding algorithms for such codes.

In Chapter 3 we show that though complete recovery of lost information is impossible when the number of losses are greater than the number of parity symbols it is still possible to recover a part of the message. We show initially for binary codes and then for an example of code based on higher fields, that the density of a code graph can be changed to improve performance. We show that depending upon the number of losses there exist an optimal density that can facilitate maximum message throughput. The analysis in this chapter lays the foundation for the work in the rest of the chapters.

In chapter 4 based on the proposed measure, we combine the advantages of lowering the density of a code for near capacity performance with the high decoding efficiency of Reed Solomon (RS) codes, in order to design optimum PRS codes. Then, we demonstrate, through an example of a Binary Erasure Channel (BEC), that at near-capacity cod-

ing rates, appropriate design of a PRS code can outperform an RS-code. We extend this analysis and optimization for a general BEC over a wide range of channel conditions.

As compared with RS codes, the proposed PRS codes provide a significantly improved graceful degradation when the number of losses exceeds the number of parity symbols within the code block. This is a highly desirable feature for realtime communication. Thus in Chapter 5 we investigate the applications of these PRS codes for transmission of real-time video. It will be shown that throughput improvement facilitated by PRS code does indeed translate into an improvement in media quality. In this chapter we also set paradigm for a unique fixed coding rate based adaptive FEC scheme. We compare the performance of such a scheme with other possible RS based adaptive FEC schemes.

In Chapter 6 we summarize our results and conclusions and make a brief remark on the possible future directions to the proposed work.

CHAPTER 2

INTRODUCTION TO CODING THEORY

In this chapter we provide a brief overview of basic channel coding and related information theory material that is relevant to the contributions of this thesis. Discussion throughout this chapter will be limited to linear block codes. The focus of the discussion in this chapter will be on decoding of linear block codes in presence of erasures. A comprehensive treatment of coding theory can be found in [7], [24], [25], [26], [27].

Decoding algorithms with linear time complexity have been described as processes on graphs. Hence, a brief introduction to some concepts of graph theory will be given. This facilitates a discussion on how any system of equations and hence any linear block code can be represented as a bi-partite graph code. Finally we discuss how a graph code in a binary Galois Field, $GF(2)$, can be decoded by using what-is-known as the bit-flipping algorithm. This leads to a discussion on possible extensions for higher order fields.

2.1 Finite Fields and Algebra

We assume that the reader is conversant with the definitions of a group, finite field, vector space, and sub-space, linear dependence, rank of a matrix etc. Interested reader is referred to [28] for an in depth explanation of abstract and linear algebra. Any of [7], [24], [25], [26], [27], should again provide sufficient detail for a reader specifically inter-

ested in the area of error control coding. Here, we try to give a brief overview of some important concepts related to finite fields. Before we proceed further, it should be noted that a finite field is characterized by having finite number of elements. A field is closed under the operations of addition and multiplication. Also, an important property of finite fields is that most of the properties of linear algebra are applicable to finite fields also.

2.1.1 Prime Fields

If p is a prime number[2] the set of integers $\{1, 2, \ldots, p-1\}$ is a commutative group under modulo-p addition. Modulo–p multiplication is distributive over modulo-p addition. Therefore the set $\{1, 2, \ldots, p-1\}$ forms a finite field of order p under modulo–p addition and modulo–p multiplication. A finite field with p elements is denoted by $GF(p)$, where GF stands for "Galois Field". Since p is a prime number we refer to such a field as a prime field.

2.1.2 Extension Fields

It has been shown that finite fields with $q = p^m$ elements exist. (For all $m > 1$). Fields with $q = p^m$ elements where p is a prime number are called extension fields and can be represented[3] by $GF\left(p^m\right)$ or $GF(q)$. The sum and product in the extension fields

---

[2] We use the symbol p for probability of erasure also. The meaning of symbol p will be evident from context. The meaning of the symbol p will be stated explicitly if it is not contextually obvious.

[3] Throughout this thesis the terms $GF(q)$ and F_q are used interchangeably.

are not computed modulo-q. Rather, field elements can be considered as polynomials of degree $m-1$ with coefficients in $GF(p)$. The sum operation is just the sum of coefficients (modulo-p) and the product operation is the product of polynomials, computed modulo an irreducible polynomial[4] of degree m.

An interesting and important property of prime and extension fields is that there exists at least one (i.e., may be more than one) special element, denoted by α, whose powers generate all the non-zero elements of the field. As an example, a generator for $GF(7)$ is 3, whose powers, starting from 3^0, are 3, 2, 6, 5, 4,1, ... Powers of α repeat with a period of length $q-1$. Thus the elements of $GF(2^m)$ can be represented as $\{0,1,\alpha,\alpha^2,\ldots,\alpha^{2^m-1}\}$.

2.1.3 Vector Spaces

F_q^N is a N-dimensional vector space over F_q. The elements of F_q^N are the q^N N-tuples denoted by row vectors $\overline{v}=[v_0,v_1,\ldots,v_{N-1}]$, where each $v_i \in F_q$. The elements

---

[4] Mathematical operations like addition, subtraction, division and multiplication can be carried out on polynomials just as done on numbers. Thus even modulo operation can be carried out on polynomials. An irreducible polynomial cannot be factorized any further and hence is equivalent to a prime number. Thus modulo operation can be carried out on polynomials with respect to these irreducible polynomials. Tables of irreducible polynomial are available in [29]. Chapter 2 of [26] can be referred to get the details on how irreducible polynomials are used for construction of extension fields i.e., $GF\left(2^m\right)$.

of F_q are called scalars. The definition of vector addition and scalar multiplication in this vector space is similar to matrix addition and multiplication.

2.2 Linear Block Codes

A (N,K) linear block code with data (message) word length K and code word length N is a K-dimensional subspace of F_q^N. The rate R of this code is defined as $R = K/N$.

Thus the encoding process of any linear block code can be represented by a matrix operation as $\overline{v} = \overline{u} \cdot G$ where,

$\overline{v} = [v_0, v_1, \ldots, v_{N-1}]$ is a row vector representing the encoded codeword,

$\overline{u} = [u_0, u_1, \ldots, u_{K-1}]$ is a row vector representing the original message data and

$$G = \begin{bmatrix} g_{00} & g_{01} & \cdots & g_{0,N-1} \\ g_{10} & g_{11} & \cdots & g_{1,N-1} \\ \vdots & \vdots & \vdots & \vdots \\ g_{K-1,0} & g_{K-1,1} & \cdots & g_{K-1,N-1} \end{bmatrix}$$ is the KxN generator matrix where, $g_{i,j} \in F_q$.

Thus $\overline{v} = \overline{u} \cdot G$ can be looked upon as a system of equations where each equation is represented by $v_i = \sum_{j=0}^{N-1} u_i \cdot g_{j,i}$, for $i \in [0, N-1]$. Therefore, the ith element of the code vector $\overline{v} = \overline{u} \cdot G$ is a weighted sum of the elements of the message vector $\overline{u} = [u_0, u_1, \ldots, u_{K-1}]$, weighted by the ith column of the generator matrix G.

2.2.1 Systematic Linear Block Codes

A (N, K) linear block code is a systematic code if the encoded vector $\overline{v}$ has a replica of the message vector $\overline{u}$, in particular $v_i = u_i$, $\forall\ i = 0, \ldots, K-1$.

Thus the system of equations $\overline{v} = \overline{u} \cdot G$ consists of K trivial equations and only $(N-K)$ non-trivial equations[5]. We call the non-trivial equations as parity check equations and the elements of $\overline{v}$ which are not exact replicas of a message symbol are called parity symbols. Thus the parity symbols form the $(N-K)$ redundant symbols and the codeword can be broken down into two parts i.e., the message part and the redundant check part as shown in Figure 2.

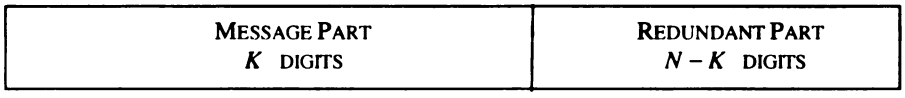


Figure 2 Systematic format of a code word

The generator matrix G of a systematic code can be written as $G = \left[I_K \middle| A_{K \times (N-K)} \right]$ where I_K is an identity matrix of dimension K and the non-trivial part of the generator matrix is given by

---

[5] A trivial equation refers to equation type $v_i = u_i$, while a non-trivial equation will usually refer to an equation of type $v_i = \sum_{j=0}^{N-1} u_i \cdot g_{j,i}$ where atleast two coefficients $g_{j,i}$ are non-zero. Sometimes in very low rate codes, like repetition codes, some of the parity check equations (non-trivial equation) can also assume the form $v_i = u_i$.

$$A_{K\times(N-K)} = \begin{bmatrix} a_{00} & a_{01} & \cdots & a_{0,N-K-1} \\ a_{10} & a_{11} & \cdots & a_{1,N-K-1} \\ \vdots & \vdots & \vdots & \vdots \\ a_{K-1,0} & a_{K-1,1} & \cdots & a_{K-1,N-K-1} \end{bmatrix} \text{ where } a_{i,j} \in F_q .$$

It is a well-known result in Linear Algebra that any matrix G can be column reduced to the form $\left[I_K \middle| A_{K\times(N-K)} \right]$. This implies that any linear block code can be reduced to a systematic block code. Thus without any loss of generality all further work in this thesis focuses on systematic codes.

2.3 Erasure Recovery and Systems of Equation

As has already been shown, the encoding process can be represented by a system of N equations and K unknowns. In the absence of errors, solving this system of equations allows us to reconstruct the message data, which can be considered equivalent to decoding the code. In fact, any error-free system of equations, expressed in term of the message symbols, with a rank K should allow us to completely reconstruct the message data. As each encoded symbol represents an equation in the system of equations, if during transmission an encoded symbol is erased, we cannot use the equation represented by that encoded symbol to solve for the message symbols. If the channel under consideration is such that the only type of possible channel failure is an erasure, then all the equations represented by non-erased encoded symbols can be used to solve for the message symbols. It is noteworthy that if the underlying code is a systematic code then some of the equations will be trivial equations like $v_i = u_i$.

As at least K equations are required to solve for K unknowns, it will be impossible to completely reconstruct the message data if the number of encoded symbols received during transmission is less than K. Thus a decoding algorithm based on solving a linear system of equations cannot recover complete data if the number of erasures L is greater than $N-K$. If the N equations are designed such that any subset of K equations are independent of each other then, such a system of equation guarantees complete recovery of message data, for any random erasure pattern of weight less than or equal to $N-K$.

Thus a code that can recover any random erasure pattern up to a loss of $N-K$ erasures, are called Maximum Distance Separable or Reed Solomon type codes. The polynomial based Berklamp and Euclidean algorithms can also be modified to work for erasures but modern implementations or erasure decoding based on systems of equations have been shown to have lesser time complexity [11].

For a systematic code if the K non-trivial equations are independent and all the coefficients for each equation are non-zero, then any subset of K equations obtained from the N equations of the systematic code will be independent. This is common to the design of RS-Codes for erasures described on the basis of Vander monde matrices in [1] and Cauchy matrices in [11]. If K encoded symbols and hence K equations are available to the receiver then the K equations can be solved by matrix inversion as shown in Figure 3. The time complexity of codes described in [11] is lesser than that in [1] because of novel way of performing finite field calculations but the basic idea (i.e. solving a system of equations) in both algorithms is identical.

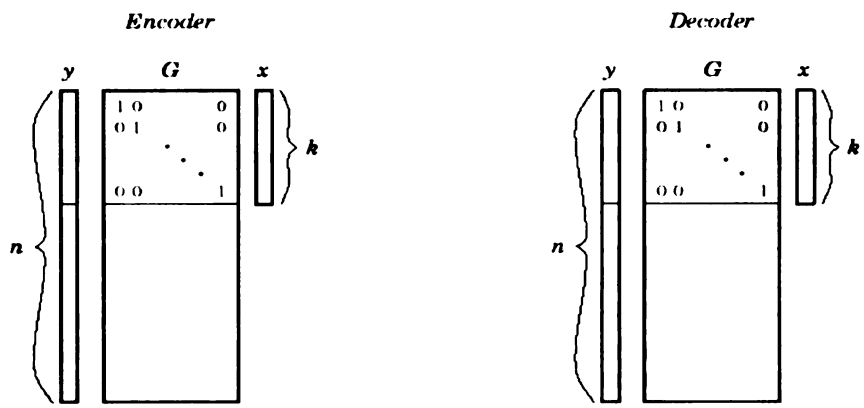


Figure 3 Encoding and Decoding process of as systematic erasure code. The gray area identifies the encoded symbols that have been transmitted successfully and the corresponding equations that can be used for reconstructing the message data [1].

As a systematic code has some trivial equations, the decoding algorithm can concentrate on solving only the non-trivial equations. This further improves the time complexity of the decoding algorithm. In such a system of equations that are formed by the non-trival equations, the message symbols (equal to the trivially encoded symbols) that do not get erased during transmission don't have to be treated as unknowns.

2.4 Concepts in Graph Theory

The last few years has seen an increased interest in the area of coding theory, one of the primary reason for this has been discovery[6] of decoding algorithms with low time

[6] The actual word should be rediscovery as this work was initially presented by Gallager [5] and only was recently rediscovered by Luby, Mackey, et. al.

complexity. These algorithms are based on concepts of graph theory and are described as processes on graphs representing codewords. At this stage, we will give a brief introduction to relevant concepts from Graph Theory by providing some basic definitions. A reader interested in more details can refer to [30], [31].

Definition 2.4.1 A graph[7] G is an ordered pair of disjoint sets (V, E) such that E is a subset of the set $V^{(2)}$ (i.e. w.l.g $V^{(2)} = \{(i, j) \mid i > j \text{ and } v_i, v_j \in V\}$) of unordered pairs of V. The set V is the set of vertices and E is the set of edges.

Definition 2.4.2 A graph G = (V, E) is a *bipartite* graph with vertex classes V_1 and V_2 if $V = V_1 \cup V_2$, $V_1 \cap V_2 = \phi$ and every edge joins a vertex of V_1 to a vertex of V_2.

Definition 2.4.3 A graph G with a number associated to each edge is called a *weighted graph.*

Definition 2.4.4 A graph $G^* = (V^*, E^*)$ is a subgraph of G = (V, E) if $V^* \subset V$ and $E^* \subset E$.

Definition 2.4.5 Two vertices are adjacent to each other if they are connected by an edge. A sequence (without repetitions) of edges forms a path on graph. Two nodes or vertices in a graph are said to be *connected* if there exists a path from one edge to another. A graph is called connected if every vertex in a graph is connected to every other

---

[7] We use G to represent a generator matrix also. This conflict of notation will be persisted with to be consistent with the available literature. The meaning of G if not contextually obvious shall be explicitly specified.

vertex in the graph. A maximal connected[8] subgraph of a graph is called the component of a graph. Thus every graph is a union of the disjoint subgraphs represented by its components

Definition 2.4.6 The *degree* of a node is equal to the number of edges incident on the node. The degree sequence of a graph is the sequence of degrees of each node in the graph. A graph is said to be *regular* if each node has equal degree. A bipartite graph is said to be regular if all nodes belonging to each of V_1 and V_2 have equal degree.

Definition 2.4.7 A *cycle* on a graph is a path that starts and ends on the same node. A *tree* is a subgraph that does not have any cycles. The n-star S_n graph is a tree on $n+1$ with one node having vertex degree n and the others having vertex degree 1.

Definition 2.4.8 A graph is *complete* if all the vertices are adjacent to each other. A bipartite graph is called complete if every vertex belonging to V_1 is adjacent to every vertex belonging to V_2.

Definition 2.4.9 The *density* of a graph (and hence the density of a code defined on a graph) is defined as the ratio of number of edges in the graph to the number of edges in a corresponding complete graph.

---

[8] Here maximal implies the subgraph with the maximum number of nodes. Thus a maximal connected subgraph is a connected subgraph containing maximum possible nodes. Any subgraph with more nodes will not be connected.

2.5 Graph Codes

Most of this work in this field has been constrained to the binary field $GF(2)$ but can be extended to bigger fields $GF(q)$. The advantage of working in $GF(2)$ is that codes field operations can be reduced to simple XOR operations. To begin with we introduce all the concepts for $GF(2)$ and then extend them for $GF(q)$.

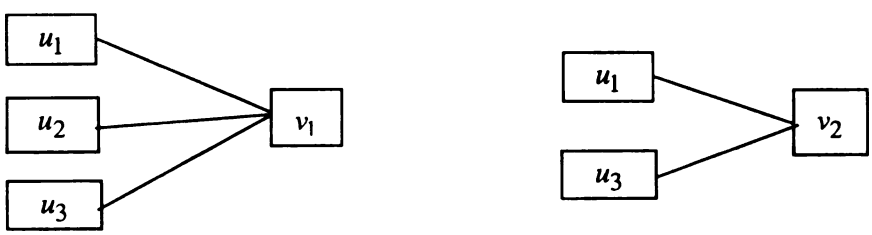


Figure 4 (a) Equation in $GF(2)$ with three unknowns represented by star graph S_3

(b) Equation in $GF(2)$ with two unknowns represented by star graph S_2

Any linear equation in $GF(2)$ can be easily represented by a star – graph. A non-zero coefficient is represented by an edge on the graph. Figure 4 (a) and (b) represents two equations $u_1 + u_2 + u_3 = v_1$ and $u_1 + u_3 = v_2$ respectively. These two equations can be combined and the system of equation can be represented by a bipartite graph as shown in Figure 5.

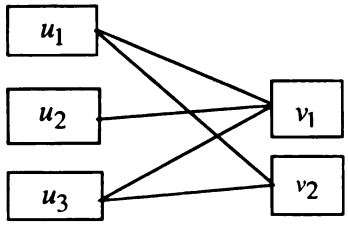


Figure 5 A bipartite graph representing a system of equation over $GF(2)$

Similarly, a system of equations in $GF(q)$ can be represented by a weighted graph, where each edge in the graph is weighted by an element from $GF(q)$. This weight is equal to the appropriate coefficient form the system of equations defining the block code under consideration.

In the previous section it was shown that, decoding linear block codes for erasures is basically equivalent to solving a system of equations. As systems of equations can be represented by graphs, we can solve these equations by defining processes on graphs. In the next section we present an algorithm called the bit-flipping algorithm, which can be used for solving systems of equations over $GF(2)$ and hence can be used for decoding of linear block codes for erasures over $GF(2)$. The algorithm we present is exactly identical to the decoding algorithm used in [8].

2.5.1 Bit Flipping Algorithm

❑ Any equation in a binary field with more than two unknowns cannot be used for erasure recovery. It should be noted that in binary field no two non-trivial simultaneous equations can be independent, i.e. in a binary field no two non-trivial equations can be used to solve for two unknowns.

- As has been already discussed, if an encoded symbol is erased during transmission then the parity equation represented by that symbol cannot be used for any erasure recovery.

- Thus, the decoding algorithm for a systematic code is equivalent to solving the parity equations represented by the un-erased parity symbols, where the unknowns are represented by the message symbols that were erased during transmission. We setup a bipartite graph to represent this system of equations, the left side of the bipartite graph has the *erased* message nodes and the right side of the graph has the un-erased parity nodes. We remove all the isolated parity nodes on the right side of the graph, i.e. parity nodes with degree zero. These parity nodes represent the parity check equations in which there are no unknowns, i.e. all the message symbols in this parity equation have been received un-erased.

- In a parity check equation, if the value of the parity bit and values of all but one message bit are known, then the equation can be solved by setting the value of the unknown bit as equal to the XOR of all known bits in that equation.

- The decoding process can now be described as follows. We begin by looking for parity nodes of degree 1 on the right hand side of the bipartite graph. This represents a parity check equation with a single unknown. On finding such a node, we set its adjacent node equal to the XOR of all the known bits in that equation as described above. We remove the parity node and message node from the graph and again search for a check node with degree 1. This process is repeated till we cannot find any more check nodes of degree 1

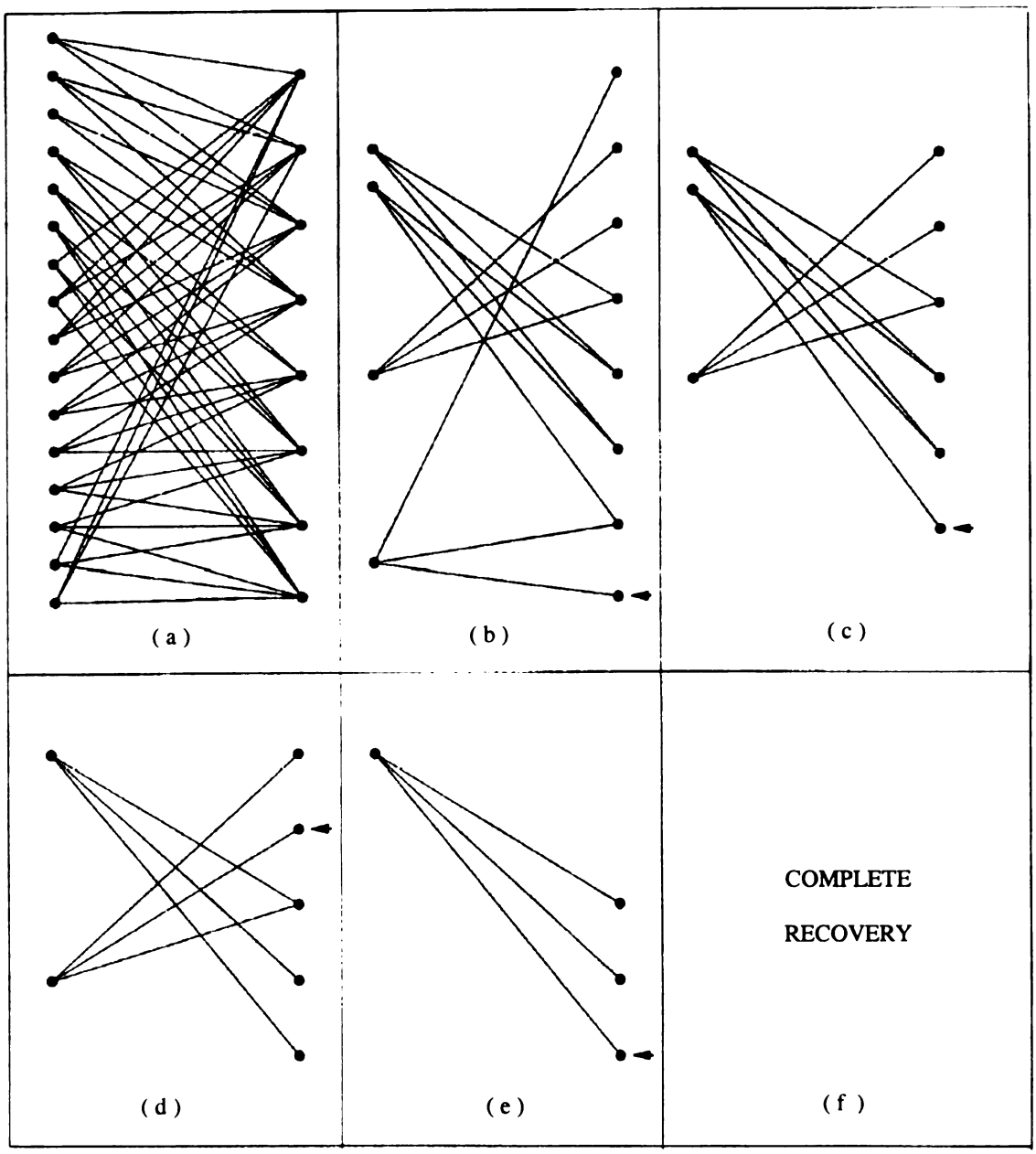


Figure 6 All stages of recovery. (a) Original graph (b) Graph induced by the set of lost nodes on the left. (c)- (f) Recovery process

Figure 6, which has been adopted from [8], describes the entire decoding process through an example. It should be noted that Figure 6 shows a specific example, where complete recovery of message data was possible. This does not have to be the case al-

ways. Sometimes the decoding algorithm could terminate with some isolated nodes on the left side or the decoding process might have to terminate with nodes on both sides. The first case is possible only when all the parity nodes that a message node was dependent on have been erased. The second case is when all the parity nodes left in the graph have degree greater than one.

- It is possible to extend the above algorithm for higher order fields. A possible decoding algorithm could be as follows.

  - Construct an induced graph as in the case of $GF(2)$, but this time the edges in the graph will have weights assigned to them. Use steps identical to the $GF(2)$ algorithm till the algorithm terminates and continue further if there are some un-isolated message nodes on the left hand side.

  - Now look for $K_{2,2}$ (complete bipartite graph with two nodes on each side) sub-graphs on the remaining graph. This represents the case of two equations, two unknowns. Unlike the binary codes, in a higher order field it might be possible to do erasure recovery with two equations and two unknowns. Solve these two equations if they are independent and remove the $K_{2,2}$. If the equations are not independent do not choose this $K_{2,2}$ again. Continue till no $K_{2,2}$ are left or no non-isolated nodes are left on the left hand side of the graph. Again go to Step 1. Go to the next step, if no parity nodes of degree less than 1 exist and neither do any $K_{2,2}$ subgraphs exist, but some non-isolated nodes are still present on the left hand side.

- Repeat the above process with $K_{3,3}$, and so on till $K_{K,K}$.

The above extension of the bit-flipping algorithm to $GF(q)$ is very complex and has a much higher time complexity than the simple bit-flipping algorithm. Matrix inversion which has a time complexity $O(n^3)$ is the most efficient algorithm for a general code on $GF(q)$. This is the primary reason that, in this thesis, we restricted our partial recovery codes to a specific graph structure and do not consider general graph codes based on $GF(q)$. Fast algorithms do exist for decoding of RS type codes. Thus we shall specifically attempt to describe our code design in terms of RS codes. Nevertheless it is important to mention that if we restrict ourselves to solving single equations and single unknowns, the extension of bit flipping for higher order fields will work fine. Mackay [32] infact showed for errors such codes can exhibit record breaking performance. Hence extension of Mackay's work for erasures and design of iterative decoding algorithms for more generalized structures of $GF(q)$ based graph codes are topics for future research.

CHAPTER 3

PARTIAL RECOVERY CODES

The majority of research in channel coding in the past century has attempted to reduce the probability of block decoding error. In chapter 1 we argued in favor of a new measure (message throughput τ_m) for evaluating the efficiency of modern channel coding techniques, especially those designed for real-time communication. As the aim of coding technique is to maximize τ_m and hence not necessarily facilitate complete recovery of message data in a code block each time, it is worth considering codes which are capable of partial recovery of information. Though such codes can be inferior to some commonly known coding techniques, as far as full recovery is concerned, the average message throughput afforded by such codes can be higher. In this chapter, we develop and analyze the performance of new linear block codes that can achieve maximum throughput. In the first part of this chapter we shall consider ensembles of codes on $GF(2)$ and show how these codes can facilitate partial recovery of information even when the number of losses are well above total number of parity bits. Moreover, it will be observed that for different channel conditions there is a different optimal code density. Though the codes we consider in this section are very simple codes, they give some important insight into the design of more complicated codes. We use this insight to design codes on higher fields. A simple coding technique based on RS coding is used to show

how partial recovery can be facilitated by codes on higher fields and conclusions about binary codes can be extended to higher fields.

3.1 Density Dependence of Binary Erasure Codes

Here we present the results of some exhaustive simulations carried out for very short block length codes. Though the codes experimented with are of very short block lengths and, therefore, may not have a lot of practical significance, the observations we make can be applied to larger block length codes. The advantage we have of using a short block – length is that we can run exhaustive simulations, which work as a " definite proof of concept".

For a given (N, K), the number of possible codes or code graphs are $2^{K\times(N-K)}$. It should be noted that for a systematic binary code the number of entries in the non-trivial part of the generator matrix are $N\ x\ (N-K)$. Thus the number of possible generator matrices is $2^{K\times(N-K)}$. Naturally some of the generator matrices are not good code designs and are trivial in nature. Nevertheless they are a part of the code ensemble. This set of all possible codes can be divided into smaller subsets depending on the density of their graphs. The density of a code graph is equal to the number of non-zero entries in the non–trivial part of the generator matrix. Thus for a (N, K) family of codes the density varies from 0 to $K\times(N-K)$. The erasure recovery performance for a given number of erasures is averaged over each subset. Here we assume that all erasure patterns of equal weight are equally likely. As the block length is not large we exhaustively generate each erasure

pattern. Thus the erasure recovery performance is averaged over all possible permutations of erasures of a given weight and over all possible codes of a given density.

The results of a few such exhaustive simulations are provided in this section. The value of N is chosen to be equal to 10 in all the simulations. The coding rate is varied by varying the value of K. The maximum value of code density is 9, 16, 21, 24, 25 for values of $K = 9, 8, 7, 6, 5$ respectively. For each of these combinations of (N, K) the above-described simulation was conducted. The results of such simulations are as shown in the Figure 7, Figure 8, Figure 9, Figure 10, Figure 11. In all figures the performance in terms of message throughput is plotted as a function of density. The number of losses L is used as a parameter for generating the different performance plots. It should be noted that as the number of losses increases the performance deteriorates, thus in each figure the highest curve represents the minimum number of losses while the lowest curve represents the maximum number of losses.

Here, we highlight the following observations:

- It can be observed that the *optimal density* is a function of block-length, the number of losses and the coding rate. Here, the optimal density is the one that provides maximum throughput.

- It can be observed that as the coding rate decreases for a given value of L the performance of the optimal density improves. This is in agreement with our intuition. In other words, as the code operates at rates further away from the channel capacity (i.e., at lower and lower rates), the performance is expected to improve.

- For a given block-length and coding rate, as the number of losses increase the value of the optimal density decreases. This can be explained by considering a *single* parity check equation. (At this point, let us assume that the code is represented by and consists of a single parity equation.) It should be noted that the number of message symbols in a single equation (i.e. the degree of a particular parity check bit) is inversely proportional to the probability of that particular parity check equation (parity check bit) facilitating any recovery. The higher the number of message symbols in an equation, the greater is the number of average number of unknowns in that equation. Thus as the number of losses increases it becomes necessary to decrease the degree of parity check nodes. This naturally translates into an overall reduction in density.

- However it should be noted that excessively decreasing the density of the graph could lead to deterioration in performance. This can be explained by considering the degree of message nodes in a graph. Higher the degree of a message node in a graph, more is the number of parity check bits that depend on that particular message symbol. Thus higher is the number of available parity check equations to recover the message symbol and hence higher is the probability of recovery of that particular symbol. Thus excessively reducing the density of a graph can excessively reduce the protection given to each message symbol and thus deteriorate performance. Thus there is a tradeoff between reducing the density to increase the robustness of the parity check equations against providing adequate amount of protection to all the message symbols. If the coding rate is low (approx ½) and the block-length is large (>5000) the number of parity check equations is high and thus probability of paying a penalty for reducing density beyond a certain threshold is not high. We tried to repeat the above

experiments for certain ensembles of LDPC codes but were unable to capture the effect of density because of the above stated phenomenon. It is a topic of future research to closely study the role density plays in conjunction with channel conditions for large block-length. Finally it is important to note that for all coding rates, even when the number of losses are higher than the available redundant symbols it is possible to recover some information. Complete recovery of lost data is not possible but the values in the figure indicate that there exist codes capable of partial recovery of lost data even when the number of losses are much greater than the redundancy. We shall show later in the thesis that when the coding rate is close to channel capacity such codes can be suitably exploited to outperform codes attempting full recovery.

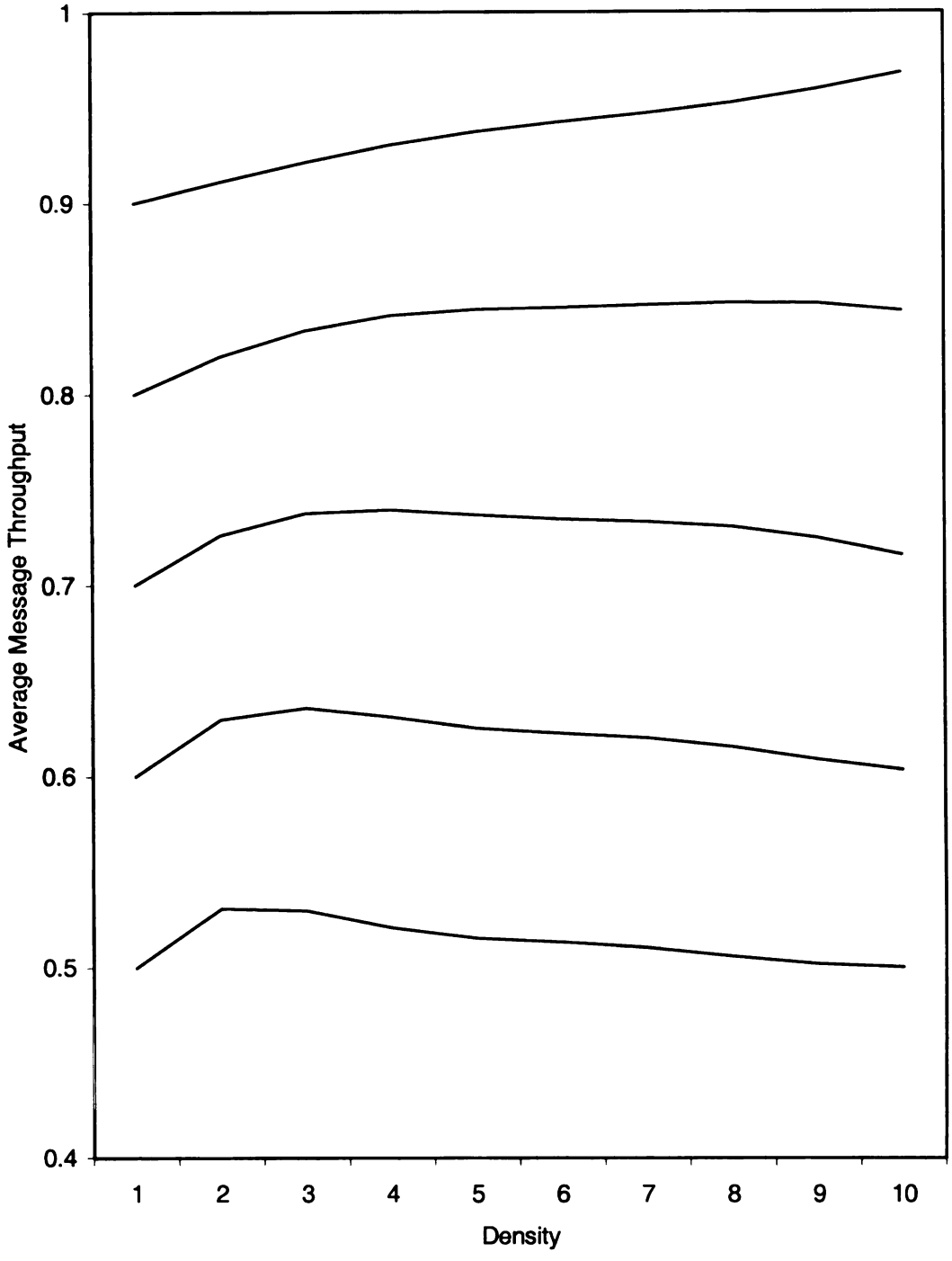


Figure 7 N=10, K=9, L = 2 to 6

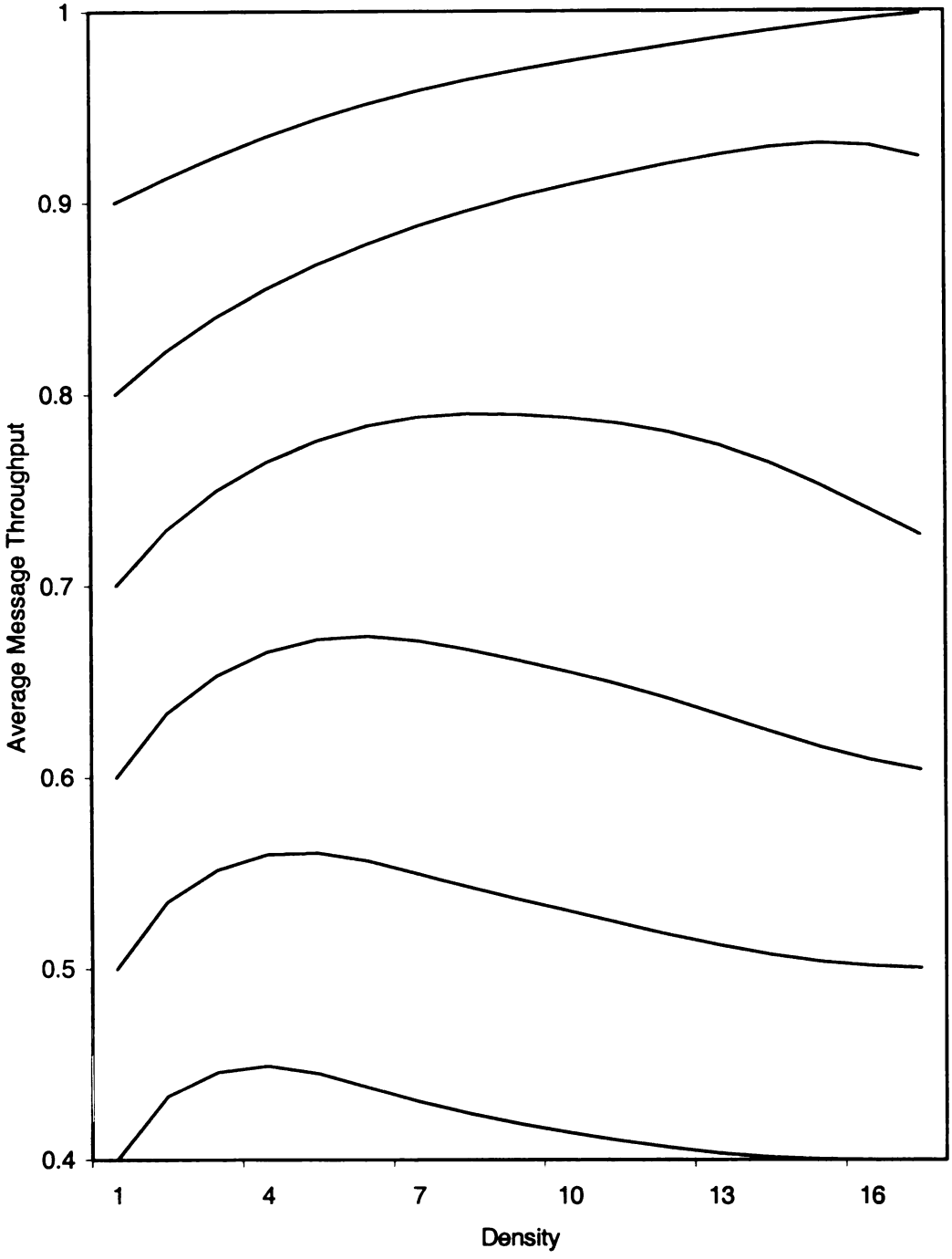


Figure 8 N=10, K=8, L= 2 to 7

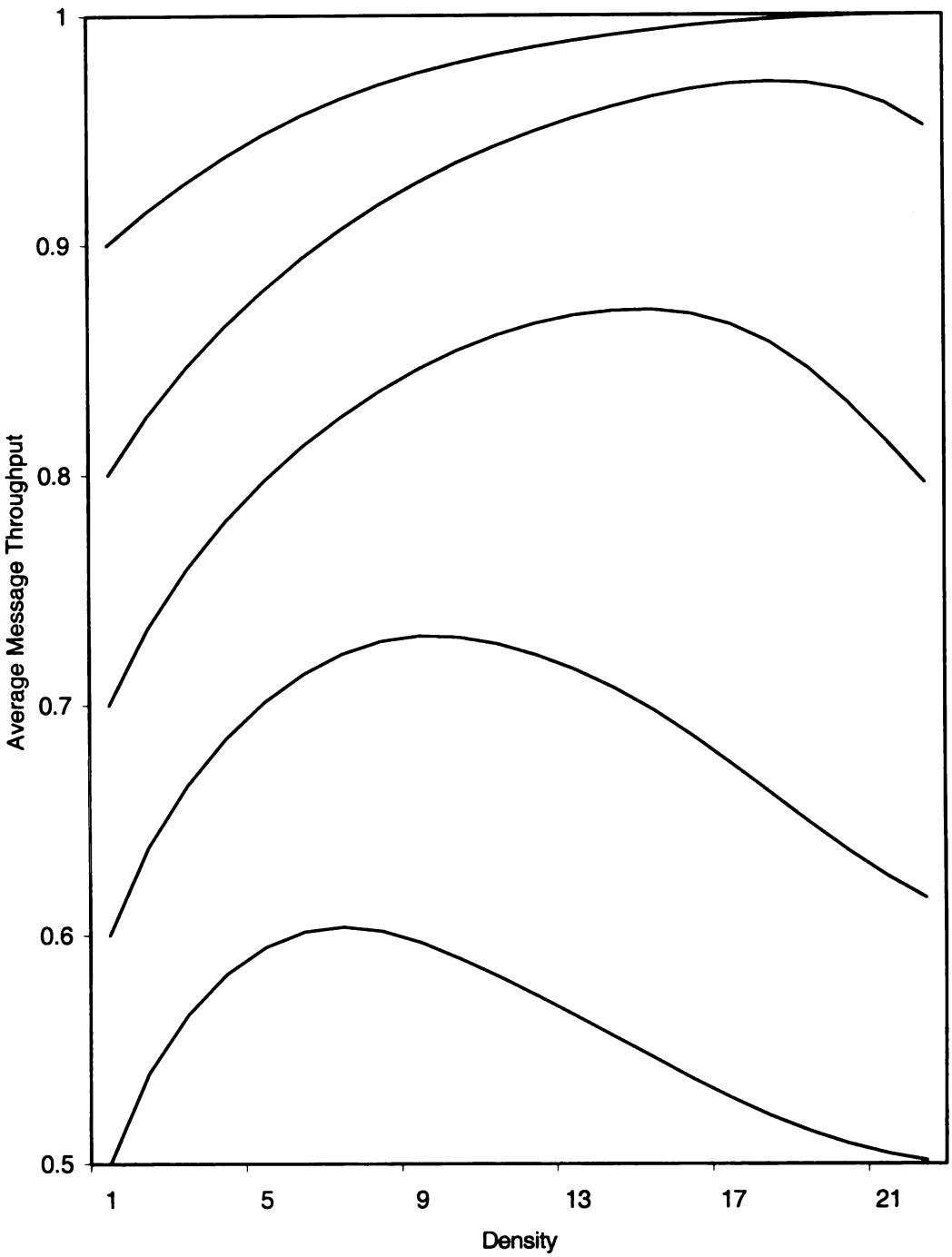


Figure 9 N =10, K= 7, L= 2 to 6

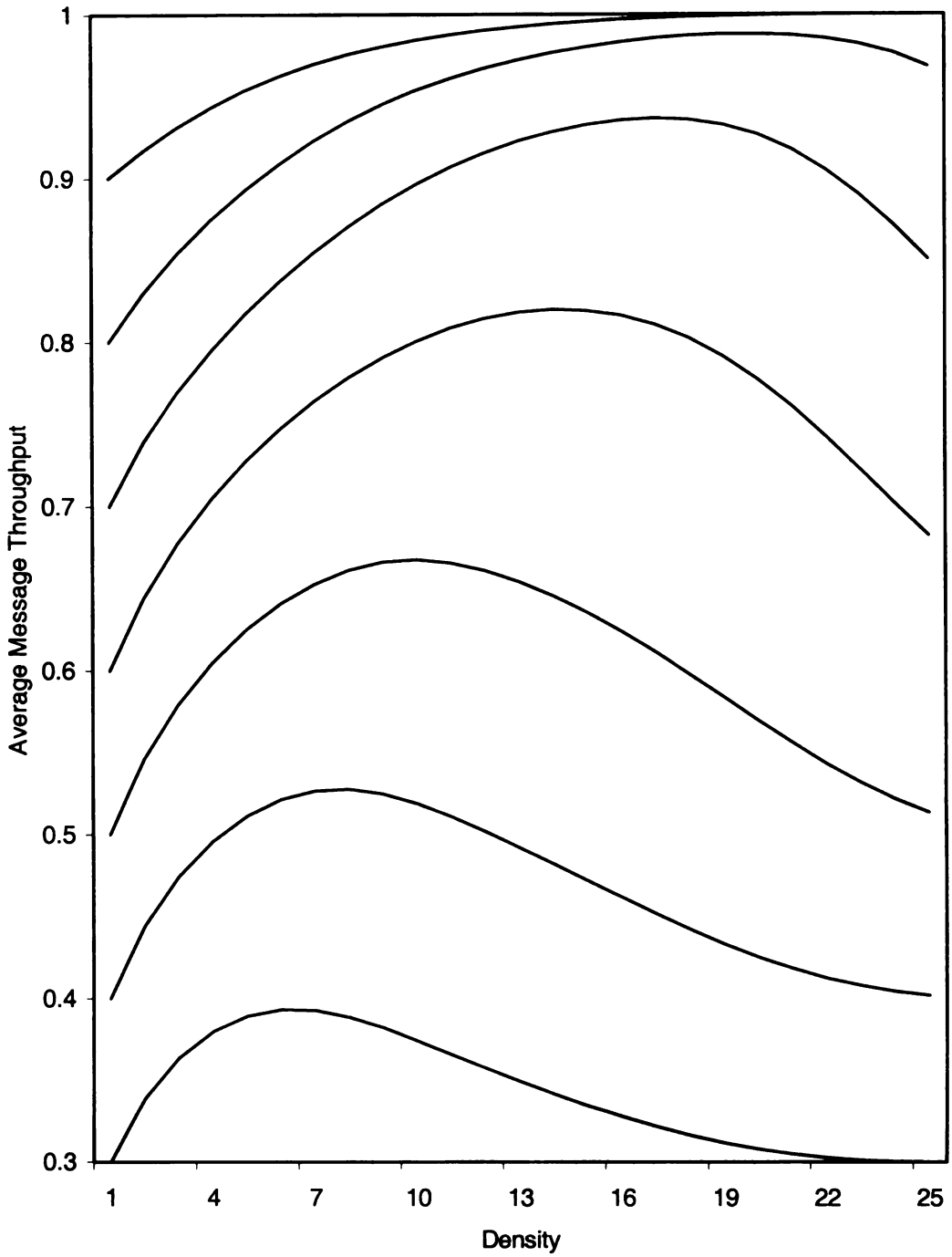


Figure 10 N=10, K =6, L = 2 to 7

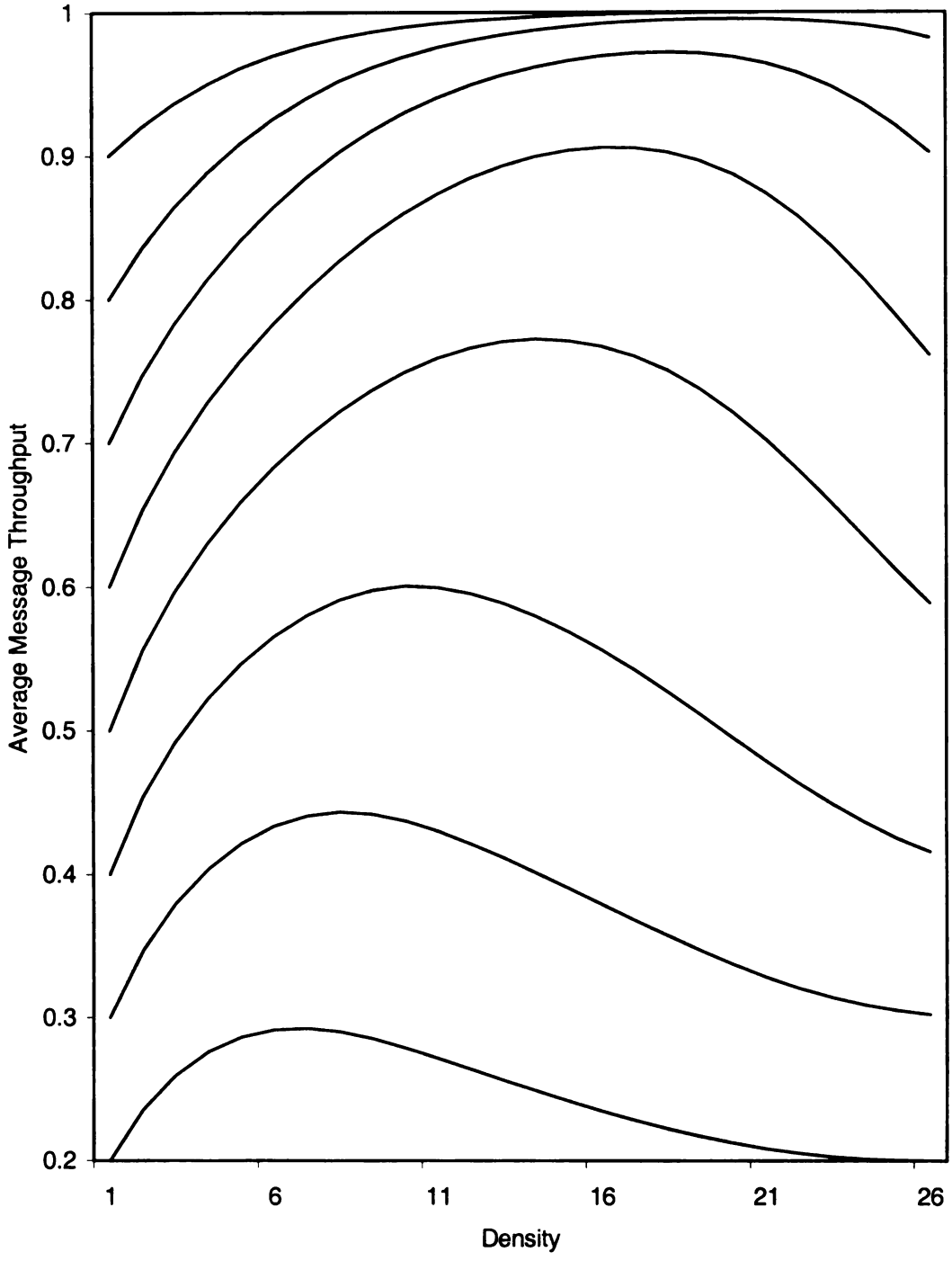


Figure 11 N=10, K =5, L = 2 to 8

3.2 Partial Recovery in Codes on $GF(q)$

The analysis in the previous section can be easily extended to non-binary codes. We shall use a simple code based on RS codes to exhibit the phenomenon of Partial Recovery in codes based on $GF(q)$. It will be shown that in this case too the density can be varied as a function of the number of losses to improve performance. In this scheme, if the number of losses L are greater than the threshold $N-K$, then we use the $N-K$ redundant symbols to protect a smaller subset $K' < K$ number of message symbols. The proposed solution is based on shortening an (N,K) RS-code to an $(N-K+K',K')$ RS-code. We refer to a (N,K) code in which all the redundancy symbols are used to protect only $K' < K$ symbols using a shortened $(N-K+K',K')$ RS-code as a (N,K,K') Partial Reed Solomon code (PRS-code)[9]. As mentioned above, here we assume that all K message symbols in the block of N code symbols are equally important. Consequently, in general, the encoder could select any subset $K' < K$ message symbols to be protected by the parity symbols. The structure of these codes is shown in Figure 12

---

[9] Partial Reed Solomon Codes introduced over here are actually just a special (but important) case of a more general family of codes. We give an introduction to a more generalized family of Partial Reed Solomon Codes in the next chapter. It can be noted then that the code we are discussing here are actually PRS codes of order 1. The notation used in this chapter is slightly different from the one used for the general PRS codes. Never the less we find the notation introduced in this chapter to be more convenient if the discussion is to be limited to PRS – 1 code.

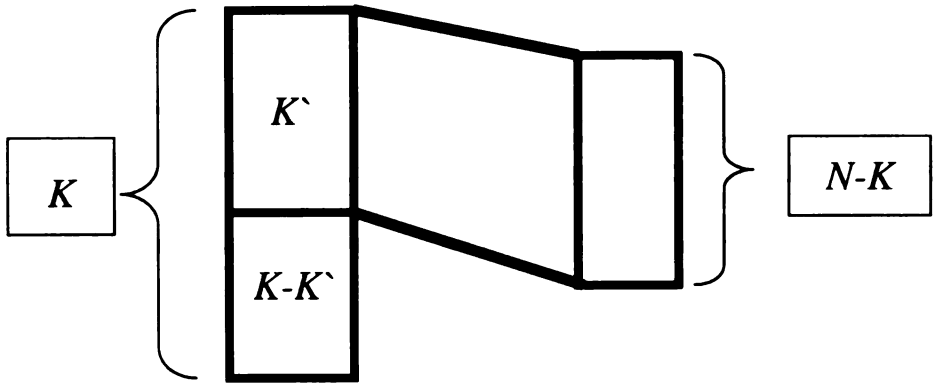


Figure 12 A Simple Partial Reed Solomon Code

It should be noted that if $L-(N-K)>K-K^{'}$ i.e. if $K^{'}>N-L$ then the number of losses in the shortened $(N-K+K^{'},K^{'})$ RS sub-code section of the PRS code will be greater than $N-K$ .In such a scenario even the $(N,K,K^{'})$ PRS code will not be able to do any recovery of the lost message data. Thus in order to achieve recovery of losses the value of $K^{'}$ must satisfy the inequality $N-L\geq K^{'}$.

If the number of losses in the entire block of length N is equal to L then the minimum number of losses in the $(N-K+K^{'},K^{'})$ sub-code section will be equal to the greater number between $L-(K-K^{'})$ and zero. The maximum number of losses in the $(N-K+K^{'},K^{'})$ section, which allows recovery of lost data, will be equal $N-K$. Similarly if the total number of losses in a block is L then maximum number of losses

possible in the $(N-K+K^{'},K^{'})$ subcode is the smaller number among L and $N-K+K^{'}$. Obviously, if any packet among the unprotected $K-K^{'}$ packets is dropped during transmission then that packet cannot be recovered. Finally it should be noted that even if a decoder is unable to recover any erased data, a message packet that has not been dropped could always be forwarded to the source decoder. Thus for a given value of L the message packet throughput is given by

$$\tau_m = \left[(T_1 + T_2) \Big/ K \cdot \binom{N}{L} \right]$$

$$\text{where, } T_1 = \sum_{i=\max(0,L-(K-K^{'}))}^{N-K} \left(K^{'} + (K-K^{'}) - (L-i) \right) \cdot \binom{N-K+K^{'}}{i} \cdot \binom{K-K^{'}}{L-i}$$

$$\text{which } \Rightarrow \quad T_1 = \sum_{i=\max(0,L-(K-K^{'}))}^{N-K} (K-L+i) \cdot \binom{N-K+K^{'}}{i} \cdot \binom{K-K^{'}}{L-i}$$

$$\text{and similarly } T_2 = \sum_{i=N-K}^{\min(L,N-K+K^{'})} ((K^{'}-i) + (K-K^{'}) - (L-i)) \cdot \binom{N-K+K^{'}}{i} \cdot \binom{K-K^{'}}{L-i}$$

$$\text{which} \Rightarrow \quad T_2 = \sum_{i=N-K}^{\min(L,N-K+K')} (K - L) \cdot \binom{N-K+K'}{i} \cdot \binom{K-K'}{L-i}.$$

➢ The term T_1 represents the performance of the code when the decoding of the sub-code is successful (i.e. number of losses in the sub-code are less than $N-K$).

➢ The term T_2 represents the case when the decoder for the sub-code has decoding error, i.e. cannot recover any lost message packet. This will happen when the number of losses in the sub-code section is greater than $N-K$.

➢ The term $K \cdot \binom{N}{L}$ represents the total number of message packets that were transmitted. This term takes into account all possible permutations of L losses with N symbols. For each possible permutation, K message symbols are being transmitted. Since, for given N and L, all of these permutations are equally likely, then the total number of message symbols transmitted under all possible permutations is the product of the two terms.

Figure 13 shows the performance of an example PRS code. The coding rate is fixed at 0.88. If $L > N-K$ then the performance of an (N,K) RS-code will be equal to that of a code in which none of the message packets are protected, i.e. the performance of (N,K,K) and $(N,K,0)$ PRS code will be identical. Any performance improvement over $(N,K,0)$ code would imply that partial recovery improves reliability and affords a better average recovery then RS codes. It can be observed that there indeed exist values of K' for which the performance of (N,K,K') PRS code is better than $(N,K,0)$ PRS code.

Moreover it can be observed for each value of L there exist an optimal value of K'. Thus appropriate choice of K' can help recover substantial number of lost packets even when the number of losses is greater than $N-K$.

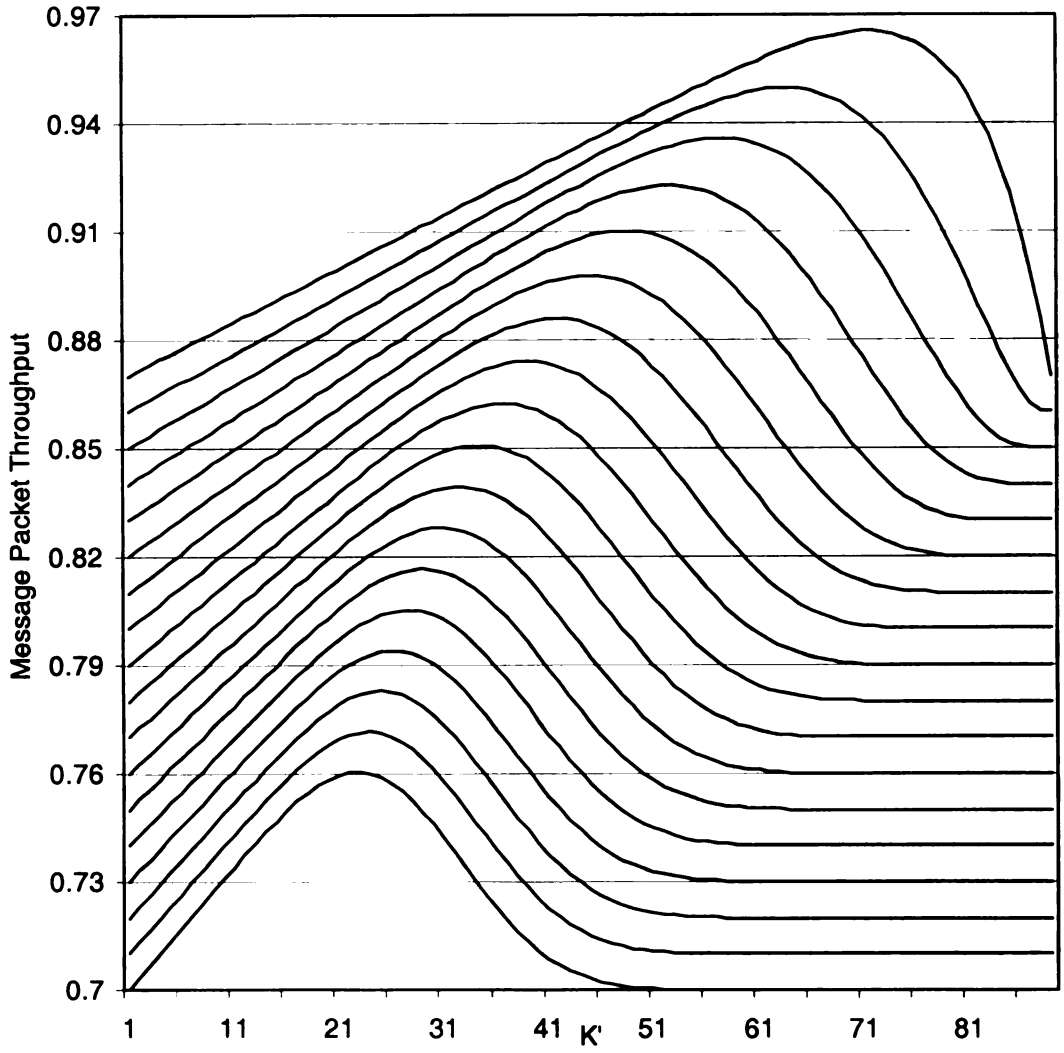


Figure 13 τ_m is plotted as a function of K' with L as a parameter for $N=100$ and $K=88$. K' takes values from 0 to 88, and value of L is varied from 13 to 30. A legend hasn't been included because, for given K' as L decreases the value of τ_m also decreases.

It should be noted that when the value of K' is reduced the density of the graph is reduced. Thus choosing an optimal value of K' is equivalent to choosing an optimal density. Thus the results in the above figure are synergistic with our conclusions for binary codes. Just as observed for binary codes, as the number of losses increase the optimal value of K' (density) reduces. Moreover it can be again observed that an over-reduction in density can deteriorate performance instead of improving it. Thus almost all the conclusions made about binary codes can be extended to these codes based on higher fields.

It can be seen in figure Figure 13 that when 13 packets are dropped (13 is greater than the redundancy 12 in the code) an appropriate choice of K' can improve τ_m by over 0.09. When 13 packets are dropped, "on-average" $0.88 \times 13 = 11.44$ of these packets are message packets. An improvement by 0.09 implies that "on-average" $0.09 \times 88 \approx 8$ out of the dropped 11.44 message packets can be recovered. This can translate into significant improvement in the quality of the perceived media. In fact even when the number of losses are much greater than $N - K$, PRS codes can improve the performance. Similar observations were made for a varied choice of coding rates, block-lengths and losses.

Thus it can bee seen that even a simple scheme based on protecting a subset of the message data can allow us to recover some part of the lost data and thus improve the reliability of the overall scheme. In the next chapter we show that optimal designs of PRS codes can provide a better throughput than RS codes for Binary Erasure Channels.

CHAPTER 4

PARTIAL REED SOLOMON CODES FOR BINARY ERASURE CHANNEL

Before we introduce the family of PRS codes we will outline the motivation for the design of PRS codes. In the previous chapters it was shown that the decoding of a codeword transmitted over an erasure channel is equivalent to solving a system of equations. The erased symbols represent the unknowns in the system of equation. Thus for a given (N, K) and a given graph density representing an LBC code, as the probability of channel erasure p increases, the average number of unknowns in each parity check equation also increase. Also, as the number of unknowns in parity check equation increase, the probability of that equation being successfully solved decreases. Due to this, when the coding rate is near (or above) channel capacity, it becomes necessary to reduce the number of message symbols that are protected by each parity symbol. This is equivalent to reducing the density of the code.

Moreover, the iterative algorithms used for decoding current LDPC codes, limit the decoding process to decoding of graphs without short-cycles. This constraint has influenced the design of most of the current LDPC codes. If a code is based on $GF(2)$, then the above constraint of designing a graph without short cycles is not a severe one. But, for

codes based on $GF(q)$, limiting the code design to graphs without short cycles can be a severe one. This can be explained by the following discussion.

A short cycle in a graph implies the existence of two parity check equations with more than one message symbol in common. Thus in case both the symbols are erased, limiting the decoding the process to "*one equation one unknown*" approach is not going to facilitate any recovery. In case of binary codes this is not a major constraint as simultaneous equations cannot be solved in $GF(2)$. However in higher order fields it is possible to recover erased data by appropriate design. Thus by constraining ourselves to a simple Bit-flipping like decoding scheme, we are reducing the efficiency (with respect to erasure recovery) of the decoding algorithm and also reducing the flexibility of our code design. Since a key objective of our effort is to maximize the message throughput (i.e., lost-symbol recovery), we did not want to constrain our code-design to graphs without cycles.

Meanwhile, decoding algorithms for a general code (with cycles) based on $GF(q)$ can have a very high time complexity. Thus we found it necessary to limit our code design to a family of codes, where the entire codeword could be broken down into sub-codes that resemble RS codes. This allows us to use algorithms developed for efficient decoding of RS codes, for decoding of these RS based sub-codes. Decoding of individual subcodes can facilitate the decoding of the entire codeword. After this brief discussion of the motivation for the proposed PRS codes, we introduce the general structure of these codes.

4.1 Partial Reed Solomon Codes

For a given realtime-pair constraint (N, K), we denote a general PRS code of order s by $(N, K, \Lambda_s)_q$. Here q represents the underlying field[10]. The order of the field is constrained by the equation $q > N$, where N represents the total number of symbols in a single codeword and K represents the number of message symbols in a codeword. Λ_s represents a $2\times(s+1)$ matrix given by $\begin{bmatrix} N_1 \cdots N_{s+1} \\ K_1 \cdots K_{s+1} \end{bmatrix}$. The entries of matrix Λ_s are constrained by the following equations:

$$N_i > K_i \;\forall\, i \in [1,s], \quad K_i > 0 \;\forall\, i \in [1,s], \quad N_{s+1} = K_{s+1}$$

$$\text{and } N = \sum_i N_i,\; K = \sum_i K_i .$$

Thus Λ_s gives an s-partition on the set of parity symbols and a $(s+1)$-partition on the set of message symbols. The code is designed such that, $\forall\, i \in [1,s]$, the pair (N_i, K_i) forms an RS-subcode over $GF(q)$ and the K_{s+1} number of message symbols are transmitted without any protection. Thus the code-graph can be divided into $(s+1)$ disjoint sub-graphs. Obviously such a code graph does not have full density and the density of the overall code has been lowered. It should be noted that an order 1 (s=1) PRS

---

[10] In all further discussion we shall drop q from the notation and assume that the order of the field on which the code is based has been pre-specified.

code with $N_2 = K_2 = 0$ is equivalent to the traditional full density RS code. In general, a PRS code with $N_{s+1} = K_{s+1} = 0$ does not include any subset of message symbols that are not protected. The above description can be clearly understood vis-à-vis Figure 14. Figure 14 shows a second order PRS code.

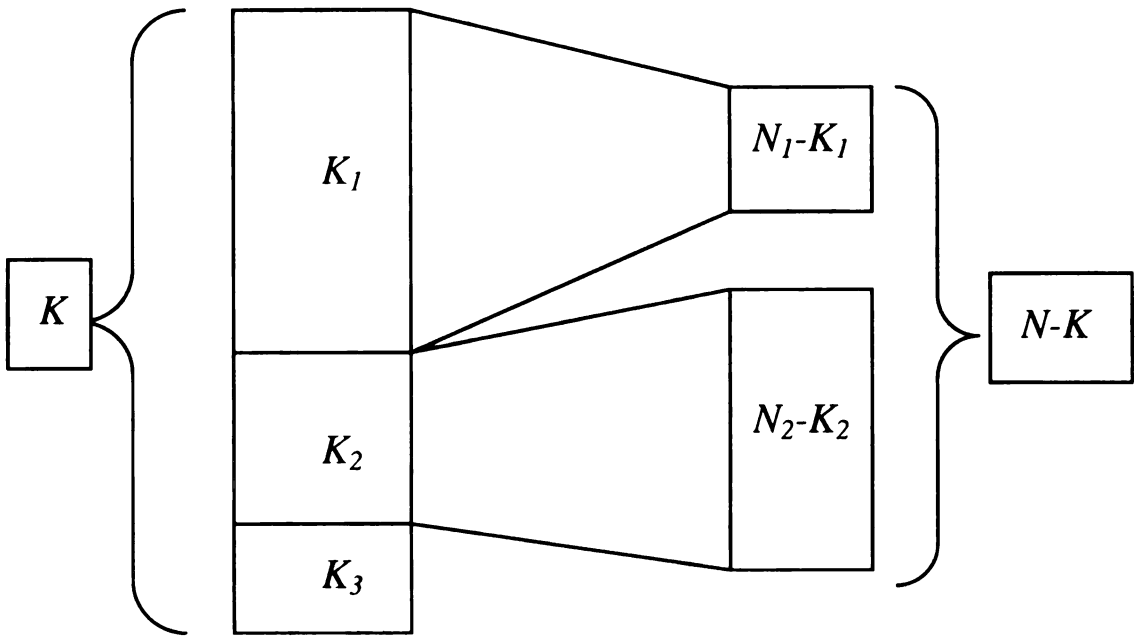
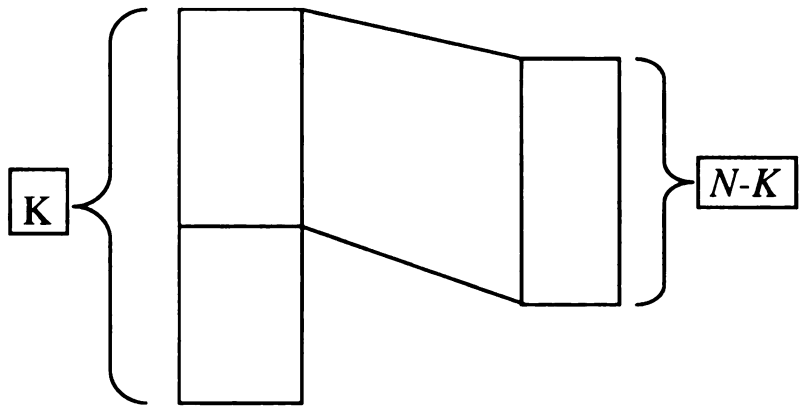


Figure 14 Reduced Density Partial Reed Solomon Code of Order 2

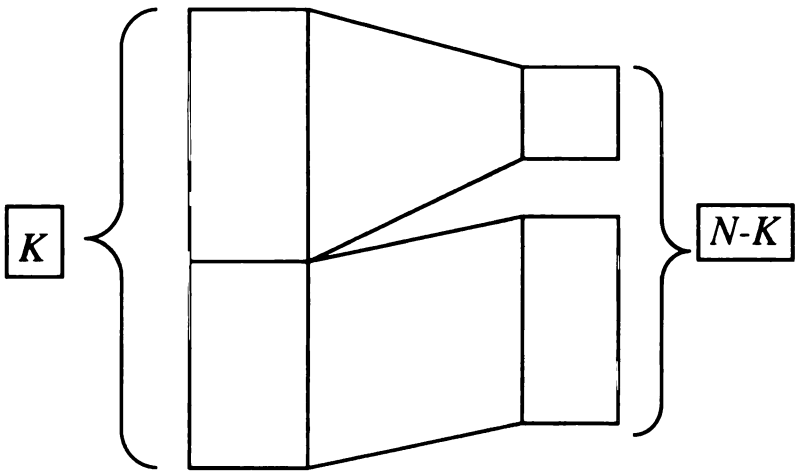
4.2 Optimal Partial Reed Solomon Codes

In this section we identify the class of optimal PRS codes for a Binary Erasure Channel (BEC) based on the message throughput criterion. We show, and with the support of some experimental evidence that, for a BEC, the optimal PRS code is given by an order 1 PRS code (i.e., PRS-1). The parameter used to measure performance of a code

here, is message throughput. Thus a code that maximizes this parameter will be the optimal code. We shall prove two lemmas, these lemmas help us to limit the ensemble of codes we have to consider to find the optima. The following notations and propositions are used by the lemmas.



(a)



(b)

Figure 15 (a) PRS-1 Codes (b) PRS-2 Codes with no symbol unprotected.

Thus the above figures represent the elements of set $\Psi_{N,K,0}$.

- Let Ψ_{N,K,K_3} be a set containing all PRS codes of order 1 with $\Lambda_1 = \begin{bmatrix} N_1 & K_2 \\ K_1 & K_2 \end{bmatrix}$ *and* all PRS codes of order 2 with $\Lambda_2 = \begin{bmatrix} N_1 & N_2 & K_3 \\ K_1 & K_2 & K_3 \end{bmatrix}$. An example of Ψ_{N,K,K_3} is the set $\Psi_{N,K,0}$. In addition to *all* possible PRS codes of order 1, $\Psi_{N,K,0}$ includes only a subset of all PRS codes of order 2 (i.e., PRS-2). This subset represents PRS-2 codes where each message symbol is protected by at least one parity symbol. In other words, no message symbols in this particular PRS-2 subset, which is included in $\Psi_{N,K,0}$, is left unprotected.

- **Proposition 1 (P1):** $\forall\ (N,K)$ the optimal PRS code in the set $\Psi_{N,K,0}$ is an order 1 PRS code.

- **Proposition 2 (P2):** $\forall\ (N,K)$, $\forall\ K_3 < K$ the optimal PRS code in the set Ψ_{N,K,K_3} is an order 1 PRS code.

- **Proposition 3 (P3):** $\forall\ (N,K), \exists$ an order s PRS code, that performs better than all order $(s+1)$ PRS codes.

LEMMA1: For a BEC **P1** $\Rightarrow$ **P2.** In other words, if the optimal code within the set $\Psi_{N,K,0}$ is a PRS-1 code, then the optimal code in the more general set Ψ_{N,K,K_3} $\forall\ K_3 < K$, is also a PRS-1 code.

Proof: Consider the optimal code on the set $\Psi_{(N-K_3),(K-K_3),0}$. **P1** implies that the optimal PRS code on this set is a PRS code of order 1. Since adding unprotected symbols

to a block will not change the relative performance of two codes on a BEC, the optimal PRS code in the set Ψ_{N,K,K_1} is also an order 1 PRS code. Thus for a BEC **P1** $\Rightarrow$ **P2.**

LEMMA 2: For a BEC **P1** $\Rightarrow$ **P3.**

Proof: Let the optimal PRS code of order $(s+1)$ be given by (N,K,Λ_{s+1}), such that $\Lambda_{s+1}=\begin{bmatrix} N_1 \cdots N_{s+2} \\ K_1 \cdots K_{s+2} \end{bmatrix}$. Using **P1** we can conclude that optimal PRS code in $\Psi_{(N_1+N_2),(K_1+K_2),0}$ is an order 1 PRS code. For a BEC the relative performance of two codewords is not going to change due to addition of identical code sections. This implies that there exists $K^* < (K_1+K_2)$ such that, the performance of (N,K,Λ_s) PRS code with $\Lambda_s=\begin{bmatrix} (N_1+N_2) & N_3 & \cdots & N_{s+1} & (K_{s+1}+K_1+K_2-K^*) \\ K^* & K_3 & \cdots & K_{s+1} & (K_{s+1}+K_1+K_2-K^*) \end{bmatrix}$ will be better than any PRS code of order $(s+1)$. Thus we can conclude that for a BEC **P1** $\Rightarrow$ **P3.**

Lemma's 1 and 2 reduce the ensemble of codes over which we need to search for an optimal code to the set $\Psi_{N,K,0}$. Now, we present experimental evidence, which allows us to formulate the following conjecture.

CONJECTURE 1: For a BEC channel P1 is true

We verified the validity of conjecture 1 for different values of N, K and p. Here, we present some results for $N=100$ and $K=88$. Any PRS code of order 2 belonging to the set $\Psi_{100,88,0}$ can be represented by $\Lambda_2=\begin{bmatrix} N_1 & N-N_1 & 0 \\ K_1 & K-K_1 & 0 \end{bmatrix}$. Furthermore,

the search space to find the optimal PRS code can be further reduced by noting that the performance of the above PRS code will be unchanged even if $\Lambda_2 = \begin{bmatrix} N-N_1 & N_1 & 0 \\ K-K_1 & K_1 & 0 \end{bmatrix}$. Thus we constraint the values of N_1 and K_1 by the following equations: $(N_1 - K_1) > (N-K)/2$ and $K_1 > K/2$.

Thus in all the figures in this section the x-axis shows the value of $(N_1 - K_1)$, the y-axis shows the value of K_1 and the z-axis shows the message throughput of the corresponding code. In each figure, **P1** is validated if the code that has maximum message throughput satisfies $N_1 - K_1 = N - K$. This represents a PRS-1 code since all of the parity codes are being allocated to protect only one subset (with K_1 elements) of the message symbols. The other subset of message symbols (with $K - K_1$ elements) is either empty (i.e., $K - K_1 = 0$) or not protected at all. In the case when $K - K_1 = 0$, we have a traditional RS code where all of the message symbols are protected by all of the parity symbols.

Figure 16 and Figure 17 show the experimental results for $p = 0.05$ and $p = 0.1$, where the channel capacity is 0.95 and 0.90, respectively. It should be noted that in both of these cases the coding rate (0.88) is below channel capacity. It can be seen in the above figures the optimal PRS code for a BEC in $\Psi_{100,88,0}$ is an order 1 PRS code. Thus using lemma's 1 and 2 it can be concluded that for a BEC the optimal code is given by PRS code of order 1. In Figure 16 it can be seen that the optimal code is actually a RS code. Thus it is possible that the optimal PRS code turns out to be a RS code depending

on the channel condition. Meanwhile, it should be noted that in Figure 17, though the coding rate is lower than the channel capacity, the optimal code is given by a PRS code of order 1 that is not equivalent to a RS code.

It has been explained in chapter 1, that though "on-average" the coding rate is lower than channel capacity, the time varying nature of a channel can make the scenarios when the number of losses are greater than $N-K$, or when the coding rate is higher than channel capacity possible. A possible way to mitigate this problem is to use some feed-back information to adapt the channel code. Thus to help in design of adaptive FEC codes, analysis of PRS codes with rates greater than channel capacity is an important topic. In the above two figures it can be observed that even when the coding rate is greater than channel capacity the optimal PRS code is a PRS code of order 1.

We also tried to find the structure of the optimal PRS codes when the numbers of losses were known. In the previous chapter we constrained our analysis when the number of losses were known to only PRS codes. Thus we wanted to investigate whether more complicated code designs could yield better performance. Thus we again considered the set $\Psi_{N,K,0}$ for our analysis. Though a thorough investigation of this problem is still a topic under study, our simulation results allow us to conclude the optimal PRS codes in this case to are order 1 PRS codes.

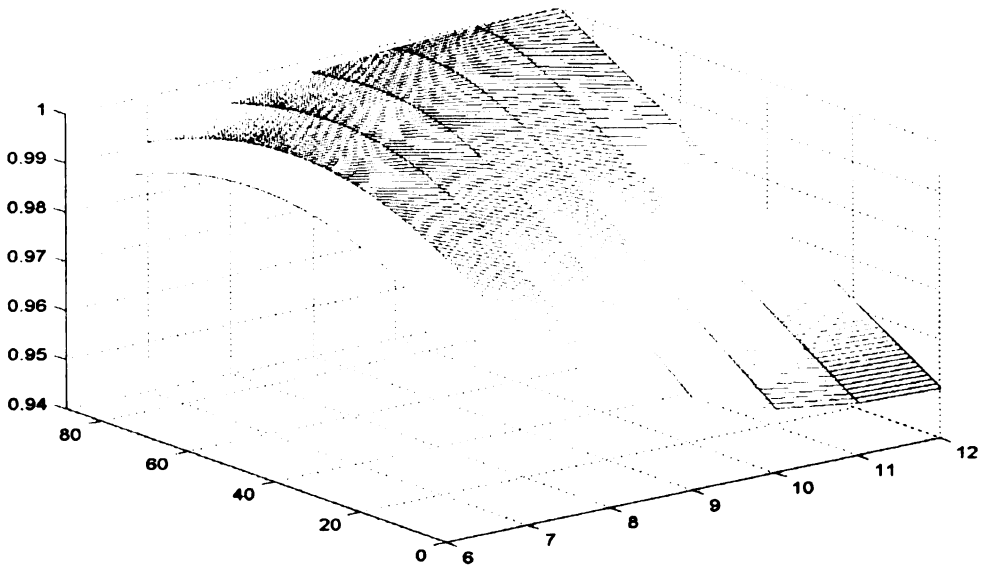


Figure 16 $N = 100,\ K = 88,\ p = 0.05$

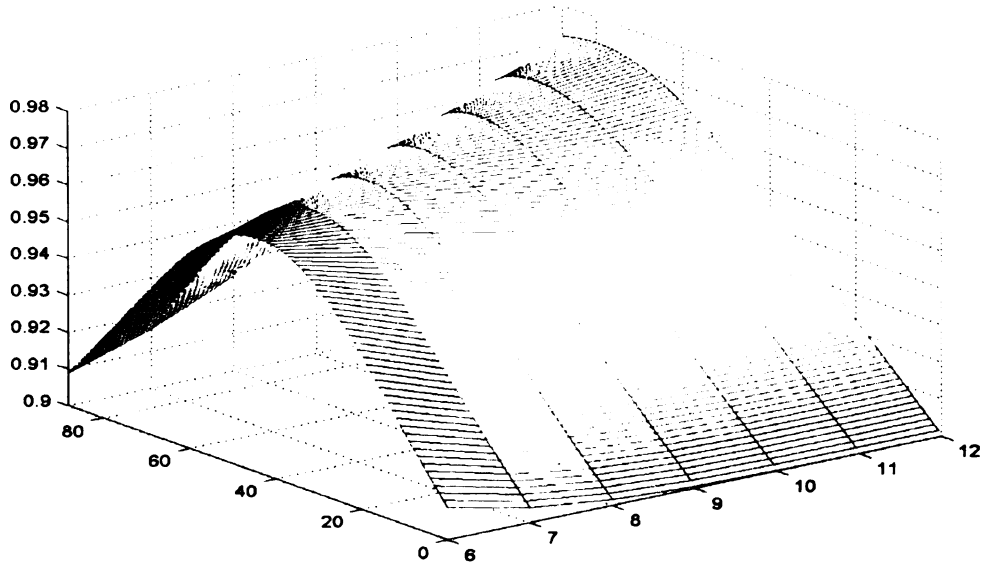


Figure 17 $N = 100,\ K = 88,\ p = 0.1$

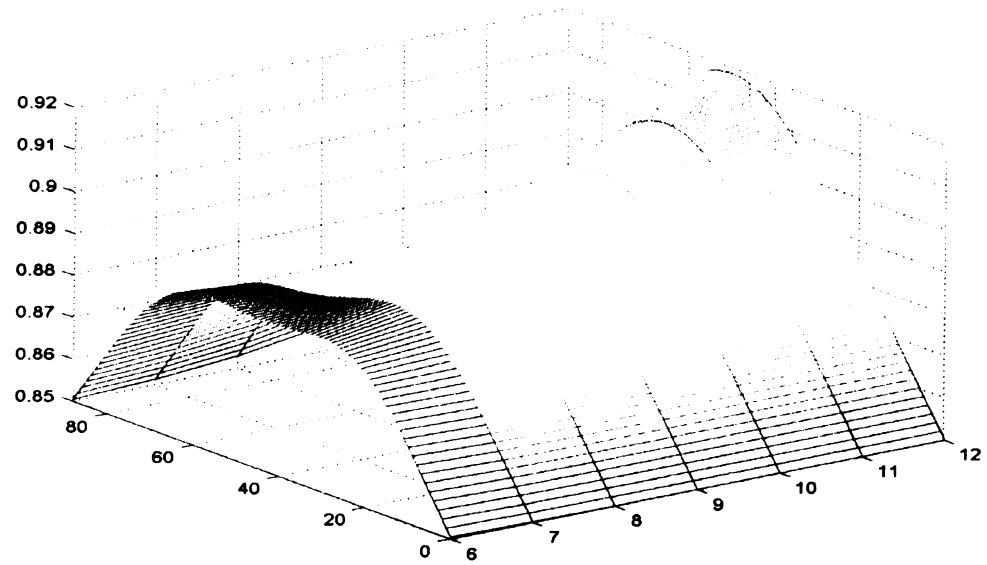


Figure 18 $N = 100,\ K = 88,\ p = 0.15$

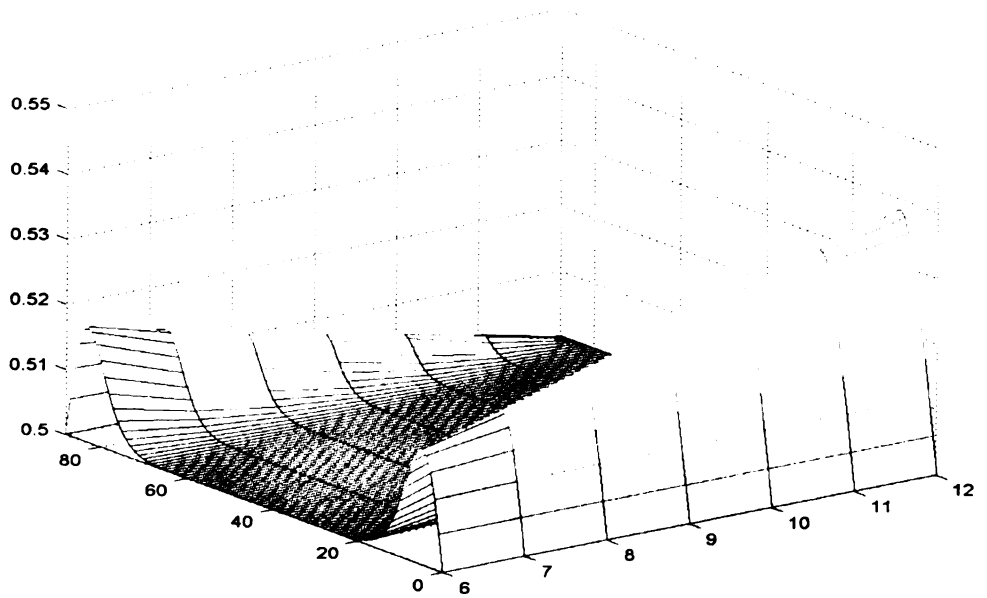


Figure 19 $N = 100,\ K = 88,\ p = 0.2$

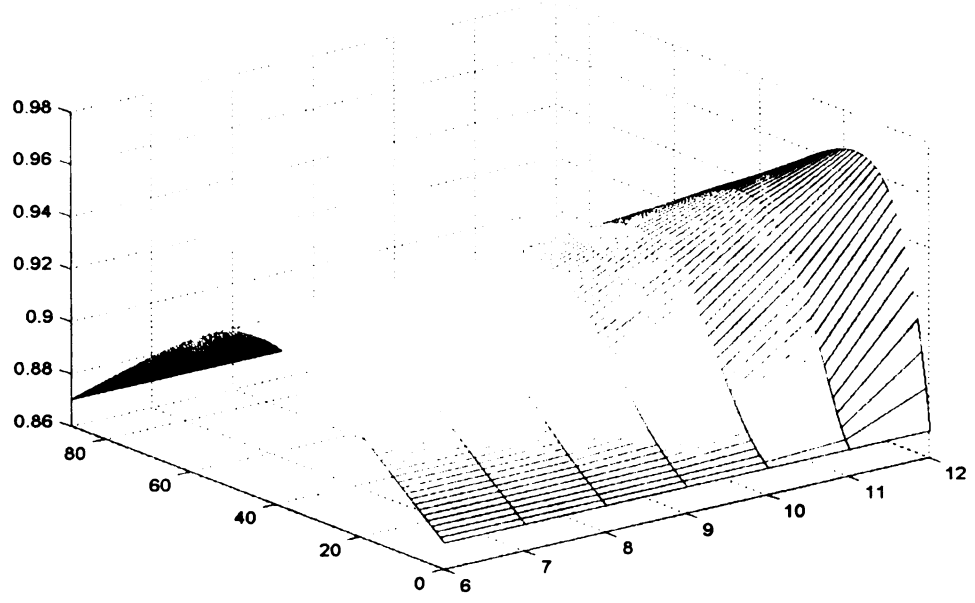


Figure 20 $N=100,\ K=88, L=13$

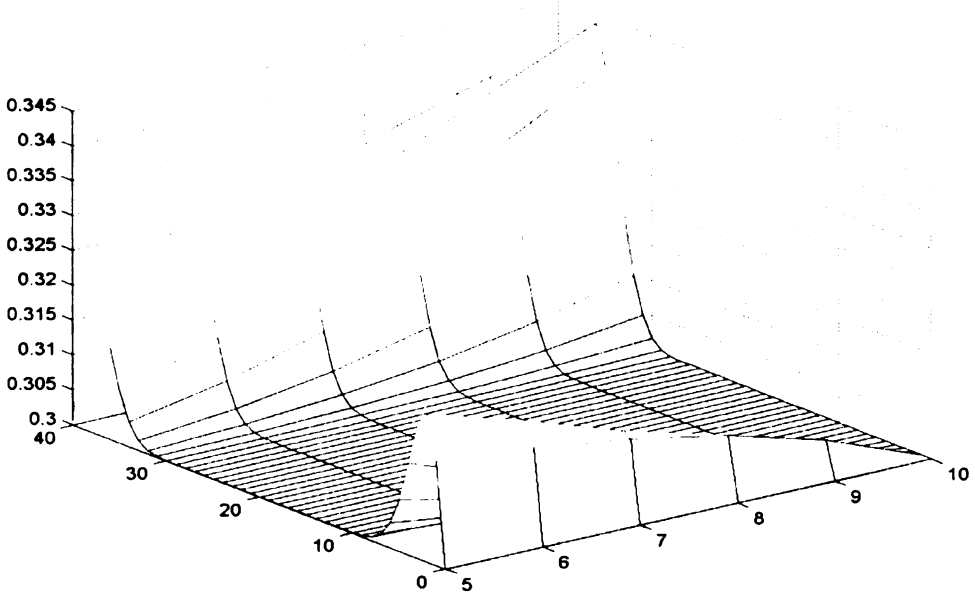


Figure 21 $N=50,\ K=40, L=35$

Figure 20 and Figure 21 show the results of some example simulations. It can be seen that for a varied block-length, coding rate and channel conditions the optimal code is a PRS – 1 code. If we consider a channel model where the receiver always knows the number of losses, then as $N \to \infty$, $p \to (L/N)$ and thus for large N. Thus for large N the channel can be approximated by a BEC channel. Thus the results obtained in Figure 20 and Figure 21 should not be surprising and are in accordance with our conclusions for a BEC.

4.3 Performance Comparison of PRS v/s RS

In this section we further evaluate and analyze the performance of PRS codes of order 1 (PRS –1). As the design of a PRS code is completely determined by our choice of K_1, we use a shortened notation for order 1 PRS code. Thus a PRS code denoted by (N, K, K_1) is equivalent to a PRS code denoted by (N, K, Λ_1) where $\Lambda_1 = \begin{bmatrix} N-K+K_1 & K-K_1 \\ K_1 & K-K_1 \end{bmatrix}$. Thus the optimal PRS code will be obtained by choosing an optimal value of K_1, denoted by K^*. It should be noted that the probability of a *message* symbol loss (*after* channel decoding) for a (N, K, K_1) PRS-1 code over a BEC with probability of erasure p is given by

$$p_m = (1/K) \cdot \left(\begin{array}{l} (K-K_1) \cdot p \; + \left(\dfrac{K_1}{(N-K)+K_1} \right) \cdot \\ \left(\displaystyle\sum_{i=(N-K)+1}^{N_1} i \cdot \binom{N_1}{i} \cdot p^i \cdot (1-p)^{(N-K)+K_1-i} \right) \end{array} \right) \quad \textbf{Equation 1}$$

The optimal value of K_1 can be obtained by minimizing Equation-1. Since $\tau_m=(1-p_m)$, this is equivalent to maximizing the message throughput. For Figures 22 – 24 we choose the block-length of the code as $N=100$. For this block-length the behavior of the performance of optimal PRS –1 code and the behavior of the optimal value of K_1 is observed. In all the three figures y-axis shows the coding rate $R=K/N$ and the x-axis shows the probability of erasure p.

In Figure 22 the z-axis shows the message throughput for the optimal PRS –1 code, while in Figure 23 the z-axis shows the ratio K^*/N for the corresponding optimal codes. It can be seen that for given N the dependence of K^* (and thus the performance of the optimal PRS –1 code) on the coding rate and $(1-p)$ is symmetrical. It can be observed that for a given loss probability p, as the coding rate increases, the message throughput decreases. . For coding rates below channel capacity the decrease in message throughput with increase in coding rate is very gradual, and the drop in performance when the coding rate is beyond channel capacity is much severe. Nevertheless, it can be observed that even for coding rates beyond channel capacity it is possible to get a reasonable message throughput and drop in performance that is graceful.

In Figure 23 it can be observed that for coding rates less than channel capacity, the optimal PRS code is a RS code, since $(K^*/N)=R$. For coding rates beyond channel capacity the ratio K^*/N decreases at a fast rate. Thus as the coding rate increases the density of the code needs to be decreased to facilitate optimal decoding efficiency. It can be observed that the decrease in the value of K^*/N with increasing coding rate despite be-

ing very fast maintains its gracefulness. This property could be utilized to obtain a closed form approximation of the dependence of K^* / N on the coding rate and probability of erasure. A closed form approximation could facilitate a fast encoding scheme for near optimal PRS codes.

The z-axis in Figure 24 shows the difference in performance of RS code and an optimal PRS code in terms of message throughput. Thus it can be clearly observed that near and above channel capacity the performance of PRS - 1 code can be better (may be much better) than an RS code of a similar rate. Thus an adaptive scheme based on PRS codes could take advantage of this to improve the overall efficiency of the FEC scheme.

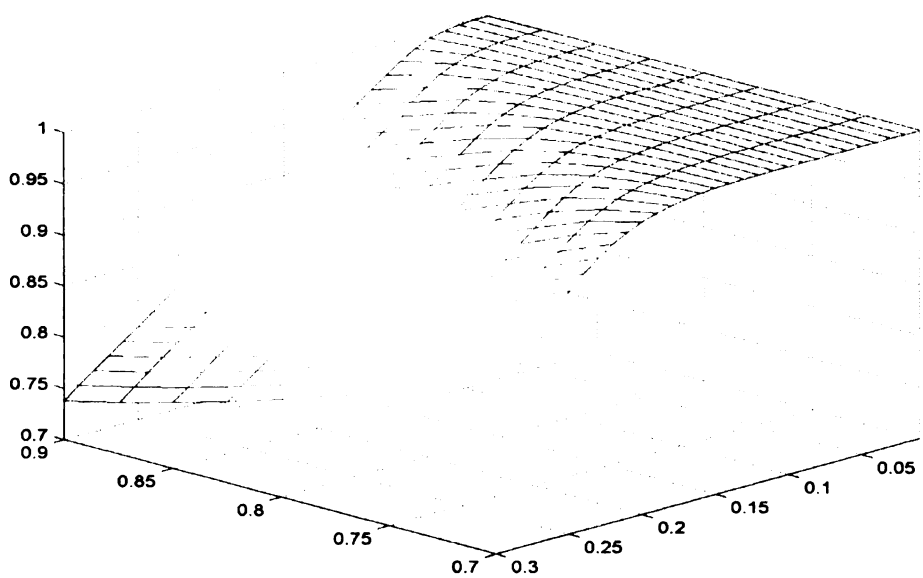


Figure 22 $N = 100$**:** Performance of optimal PRS –1 code

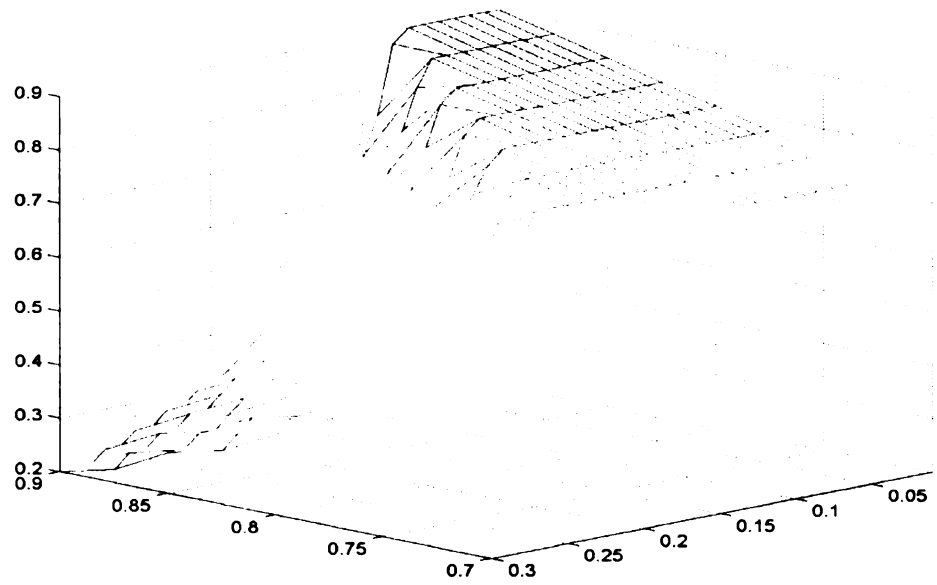


Figure 23 $N = 100$: Dependence of optimal $K^{'}$

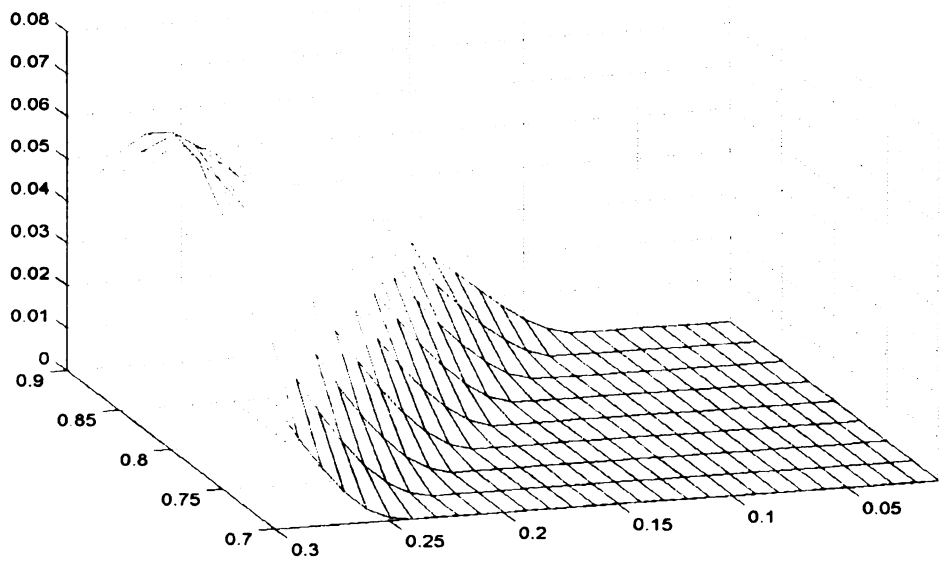


Figure 24 $N = 100$: Difference between RS and PRS-1

At this stage it is important to emphasize the fact that there exist coding rates below channel capacity for which an optimal PRS code can outperform an RS code of similar rate and block-length. Figure 25 shows one such example. Message throughput performance of optimum (N, K, K^*) = (100, 88, K^*) PRS codes is compared with the RS (N, K)=(100,88) code over different Binary Erasure Channel (BEC) conditions. The coding rate K/N = 0.88 is lower than the channel capacities. It is clear that the optimum PRS codes are maintaining better overall message throughput under these conditions. Figure 26 shows the optimum value of K' as a function of p. It can be observed that as value of p increases the optimal value of K' decreases. This is equivalent to reduction in density.

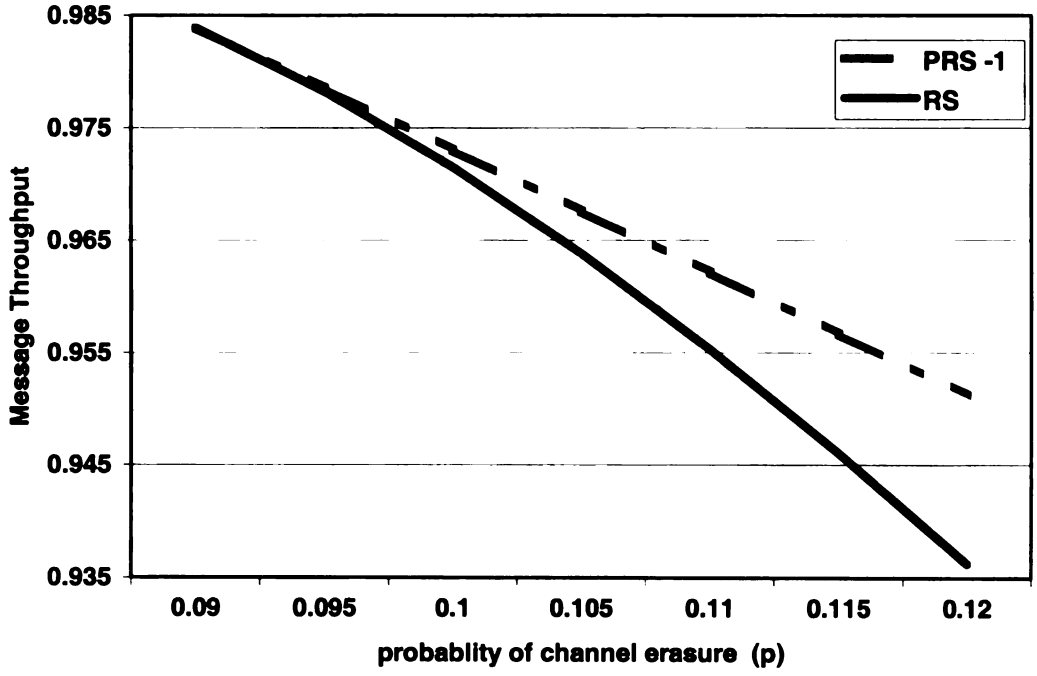


Figure 25 (100, 88, K^*) PRS codes as compared with the RS (N, K)=(100,88) code over different Binary Erasure Channel (BEC) conditions.

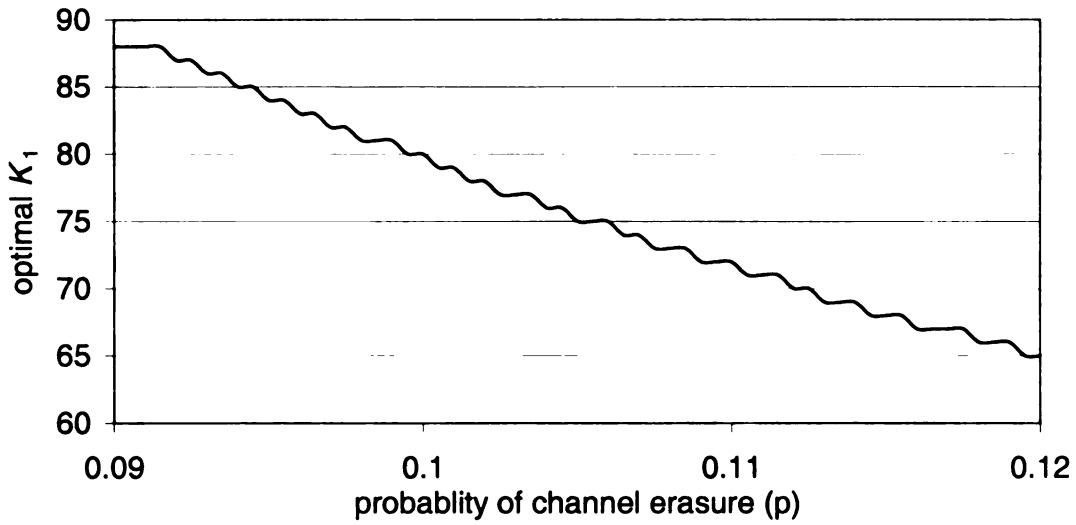


Figure 26 Optimum value of K_1

As the dependence of optimal PRS codes on channel capacity and coding rate is symmetric, it can be concluded that for a given probability of erasure and block-length there exists a critical coding rate lesser than channel capacity, such that, for all coding rates above this critical value, there exists an optimal PRS code that can outperform the traditional RS code. Moreover, it can be shown for the PRS-1 codes,

$$\text{As } N \to \infty,\ p_m \to 1 - \left(\frac{K_1 + (1-p)(K - K_1)}{K} \right) \Rightarrow,$$

$$\text{As } N \to \infty,\ K_1 \to (1-p) \cdot (K_1 + (N-K)) \Rightarrow,$$

$$\text{As } N \to \infty,\ p_m \to 1 - \left(\frac{(1-p) \cdot (K_1 + (N-K)) + (1-p) \cdot (K - K_1)}{K} \right)$$

$$\text{i.e. as } N \to \infty,\ p_m \to 1 - \left(\frac{(1-p) \cdot N}{K} \right),$$

Thus, since $N \to \infty$, $C \to (1-p)$ and $R = K/N$, we can conclude that as $N \to \infty$, $p_m \to 1-\left(C/R\right)$.

By combining the (inverse of the) channel coding theorem with this result, we can conclude that as $N \to \infty$, the critical rate becomes equal to the channel capacity of the BEC.

Thus in this chapter it has been clearly shown that for a Binary Erasure Channel, if the coding rate is close to channel capacity then the erasure recovery performance of PRS codes is much better than RS codes. Moreover it was shown that the optimal PRS codes are simple order 1 PRS codes. In the next chapter we look at the applications of these codes. In particular we investigate with some multimedia examples whether the improvement in throughput performance does indeed translate into improvement in media quality. We shall also investigate the role PRS codes could play in adaptive FEC schemes.

CHAPTER 5

APPLICATION OF PRS CODES

In this chapter, we extend our work, which employed Partial Reed-Solomon (PRS) Codes at coding rates near channel capacity on a Binary Erasure Channel (BEC). We demonstrated that an appropriately designed PRS code outperforms the classical Reed-Solomon (RS) code for a performance criterion tailored for realtime applications. In this chapter we shall illustrate that PRS codes exhibit a graceful degradation in erasure recovery performance and, hence, are suitable for multimedia communication. Our video simulation results will outline that the enhanced erasure recovery yields a profound improvement in the perceived media quality. Finally we investigate the performance of the dividend rendered by PRS codes operating above channel capacity. In particular we define a paradigm for a unique "fixed rate" adaptive FEC scheme based on PRS codes.

5.1 Graceful Degradation in Performance

Figure 27 shows the comparative performance of $(100,88)$ codes of rate $R = 0.88$ as a function of number of packet losses (L). It should be noted that the avg. no. of packets dropped = $R \cdot L$. The performance of an RS code is compared with PRS – 1 code optimized for various erasure probabilities. It can be observed that when a RS code block experiences a number of losses that is larger than the number of parity symbols, then the code is incapable of recovering any of the lost message data. Experiencing a number of

losses that is larger than the number of parity symbols is quite feasible, even if, "on average", the message rate R is lower than the channel capacity. This is particularly true when the message rate R is close to (but may still be lower than) the channel capacity. On the contrary the performance of PRS – 1 code shows a graceful degradation in performance. Depending on the channel conditions, this property can be suitably exploited to provide better packet recovery than an RS based FEC scheme. The above phenomenon is also responsible for PRS-1 codes showing better performance than RS codes in Figure 25. Video simulations provided in the next section shall further illustrate the significance of a graceful degradation in performance

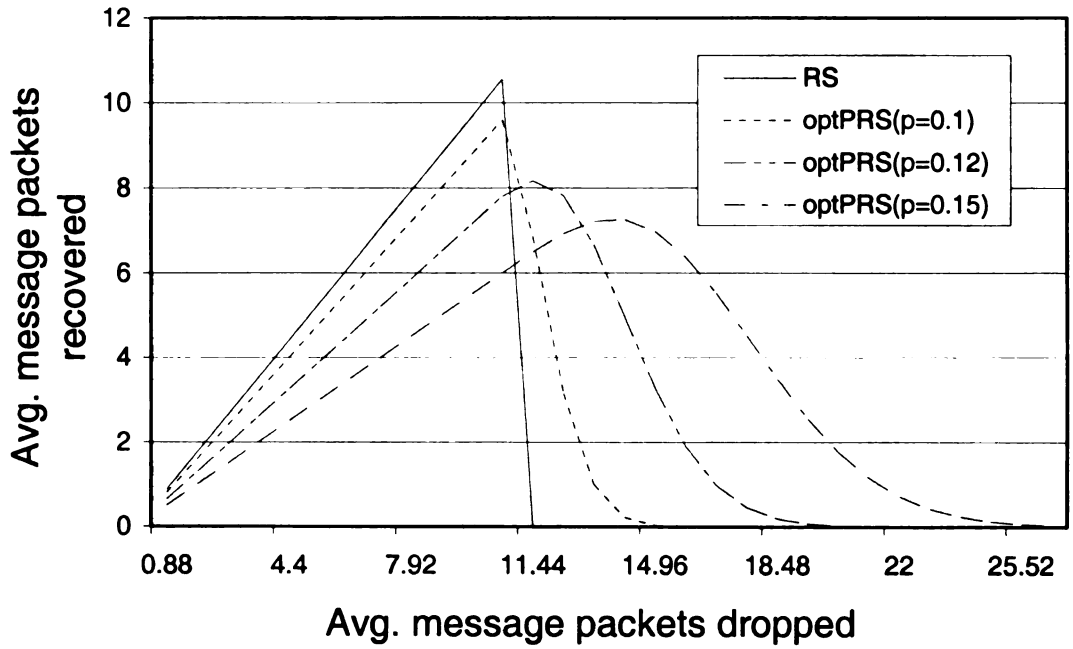


Figure 27 Recovery capability of codes optimized for different channel conditions.

5.2 VIDEO SIMULATIONS

The overall performance due to the graceful degradation in performance of PRS codes, as the number of losses in a code block increase, is further improved when the performance is measured in terms of perceptual image quality instead of message throughput. This can be attributed to the limitations of error concealment algorithms, which are effective only when the numbers of losses (after channel decoding) are not substantial. We used the newly emerging JVT standard [33] as an underlying video coding technique to compare the performance of RS and PRS channel coding schemes under identical channel conditions and identical loss patterns.

We use the standard test sequence *foreman* to present our results. The sequence was coded at 1MBps at 30 HZ. A GOP size of 15 with a frame sequence IPPP was used. A packet size of 512 bytes and slice size of 512 bytes were used for the purpose of our simulations. Figure 28 and Figure 29 just show instances in a particular ensemble of the simulations, but similar results were observed for numerous repetitions of the experiments. These figures show the results obtained by using (100,88) RS and (100,88,72) PRS-1 (optimized for p=0.11) codes. When the number of losses in a code block is less than N-K the performance of RS codes is better than that of the PRS code. The difference in performance between the two schemes is the maximum when L=N-K. As against this the performance of a PRS code is better than an RS code when the number of losses are greater than N-K. The improvement due to a PRS code is the least significant when the number of losses $L = N$-K+1.

Figure 28 Clockwise an instance in the foreman Sequence for

L= 12 RS code, L=12 PRS – 1 code optimized for p=0.11,

L=13 PRS – 1 code optimized for p=0.11, L = 13 RS code.

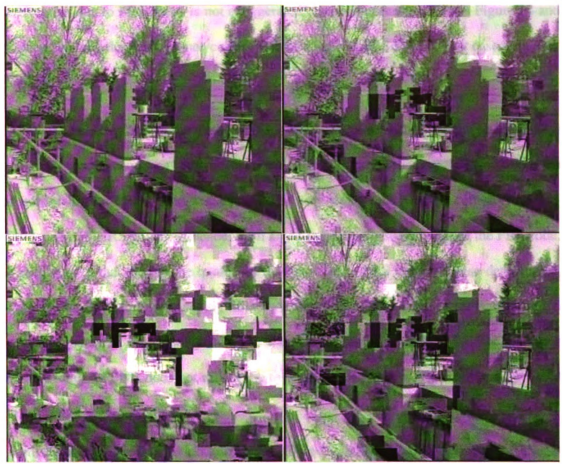


Figure 29 Clockwise an instance in the foreman Sequence for

L= 12 RS code, L=12 PRS – 1 code optimized for p=0.11,

L=13 PRS – 1 code optimized for p=0.11, L = 13 RS code.

In our simulations we forced the number of losses in each code block to be equal to L. The Figures shown here present the results for the cases when $L = N\text{-}K$ and $L = N\text{-}K+1$. Moreover for $L = N\text{-}K$ these figures show the comparison of the worst affected frames for a PRS coded sequence. In addition, for $L=N\text{-}K+1$, comparison of frames when the improvement due to PRS codes is not exaggerated[11] has been presented. Thus Figure 28 and Figure 29 show the performance comparison of a RS and PRS for a "worst case scenario" for PRS.

It can be clearly seen in the above mentioned figures that when $L=12$ the image quality for an RS coded sequence is better than that of a PRS coded sequence. Nevertheless the distortion in the PRS coded sequence is not very significant. On the contrary the performance of the RS coded sequence when $L=13$ is much worse than that of the PRS code. It can be seen that though the quality of the image for a PRS sequence also deteriorates, the increase in distortion is not significant. However the increase in distortion for an RS coded sequence is high enough to almost make the frame unintelligible. For such low quality images PSNR does not reflect the true quality of the image and hence only visual results have been presented.

In the above experiments no knowledge about the source model was used for allocation of parity symbols i.e. the symbols to be protected in a PRS code block were chosen without taking into consideration the importance of I frames or the temporal proximity of P frames to a particular I frame. Thus we are not attempting to provide a new unequal

---

[11] There were many instances when a particular frame in an RS coded sequence was significantly distorted but a PRS coded sequence had absolutely no artifacts, we avoid presenting such comparisons.

error protection scheme, however in this case the best PRS code for a BEC is an unequal distribution of parity. A more appropriate interpretation of such a code would be to recognize it as an irregular graph code [34]. In addition the error robustness features in the standard were kept at a minimal. i.e. features such as forced intra coded blocks, data partitioning, use of B-frames etc. were turned off. Taking all the above features into consideration can significantly improve the performance of PRS codes, but even without these features and even for worst cases the performance improvement of PRS codes is significant.

5.3 PRS-1 BASED ADAPTIVE FEC SCHEMES

Over channels with time-varying characteristics multiple code blocks can experience a number of losses that are larger than the number of parity symbols. Thus, though "on-average" coding rate is lesser than channel capacity, it is possible for the coding rate to be greater than the channel capacity for a period of time. If the change in channel conditions is slow enough and if a channel can provide some feedback information about the channel conditions, then the underlying error control code in an FEC scheme can be changed to adapt to the channel conditions. The feedback information can be provided to the transmitter using many possible approaches depending on the application. Also, the particular approach used by the system to use this information for channel coding purposes can be achieved in several ways. For example, the parity symbols can be transmitted in a delayed and synchronized way relative to the original message symbols and in response to the feedback information. Also, the number of losses L may represent some form of a "current average" of losses that being experienced by the channel over a recent history.

This way, the feedback information may be updated periodically and not necessarily for every block of N transmitted symbols. This approach could be feasible for channels that change relatively slowly. In this case, L/N would represent a current (updated) average for the packet loss ratio.

Most of the current FEC schemes adapt to the channel conditions by changing the coding rate R. If the loss probability increases the number of parity symbols are also increased (thus the rate is adapted to always transmit below channel capacity). For a real–time application this is equivalent to increasing the transmission bit-rate. Increasing the transmission bit-rate is not always feasible and thus changing the coding rate in an adaptive FEC scheme is not always suitable.

Using a PRS code based adaptive FEC scheme can mitigate the above problem. In such a scheme the coding rate is kept fixed, but the underlying PRS -1 code can be changed. The feedback information about the erasure probability from the channel can be used to optimise the design of the underlying PRS –1 code. It should be noted that the coding rate of the PRS code could be greater than channel capacity for a limited period of time. Thus a performance analysis of PRS codes with rates greater than channel capacity is required. Figure 30 shows such a analysis. It compares the performance of (100,88) PRS – 1 codes optimised for different channel conditions, with the performance of (100,88) RS code. It can be observed that the PRS – 1 codes perform significantly better than an RS code and can recover more than 85% of the lost message information even when the coding rate is well above channel capacity.

It should be realized though, that it is possible to design an RS based fixed transmission rate adaptive FEC scheme. This can be achieved by changing the rate of a code without changing the block-length and transmission rate. The two possible ways to achieve this are

(a) Transmitting only a subset of K^* message packets out of the K message packets and protecting these K^* message packets by $N-K^*$ parity packets instead of $N-K$.

(b) Transmitting only a subset $N\cdot(1-p)$ message packets out of the K message packets and protecting these $N\cdot(1-p)$ message packets by $N\cdot p$ parity packets instead of $N-K$.

Figure 30 shows that performance of scheme (a) is much worse than optimal PRS –1 code. The performance of scheme (b) is better than RS code but still inferior to that of an optimal PRS code. Never the less we believe that it is possible to get performance comparable to the optimal PRS –1 codes by optimally dropping packets before transmission and decreasing rate as described in (a) and (b). Even such a hypothetical scheme, on account of being an RS based scheme will not exhibit graceful degradation. This can be explained by noting that, the feedback about channel conditions is an estimate over multiple code blocks, it is possible for an RS code to be ill designed for individual blocks. In such an event the performance of a PRS-1 code will not deteriorate as rapidly as an RS based code.

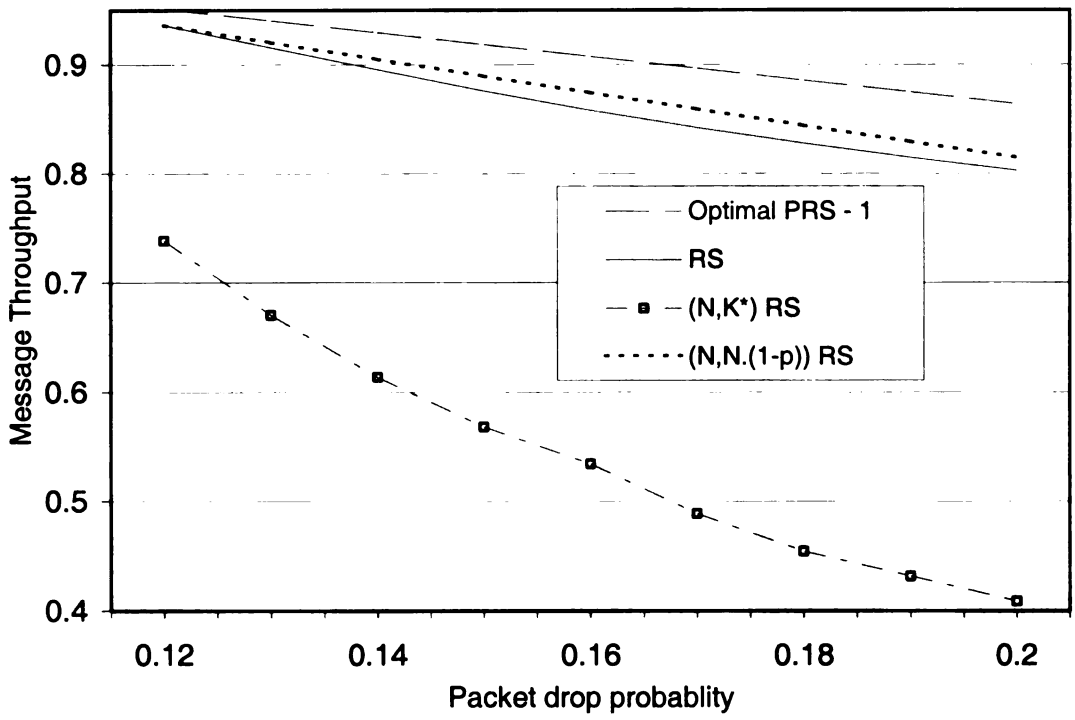


Figure 30 Comparison of (100,88) optimal PRS –1 with (100,88) RS, (100,K*) RS and (100,100·p) RS for coding rate greater than channel capacity.

Thus it can be appreciated that despite a comparatively simple design, PRS codes can be used for real-time multimedia applications. More generalized code structures and application of such coding schemes to problems other than Multimedia streaming are topics of future research.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

In this thesis, we studied the interplay of density of a good channel code and channel conditions. We introduced the concept of Partial Recovery. The existence of codes that could facilitate partial recovery of information, in adverse channel conditions, when full recovery is impossible, was exhibited. We introduced a new family of linear block codes, which we refer to as Partial Reed Solomon (PRS) Codes. These codes are specifically designed and optimized for real-time multimedia communication over packet-based erasure channels. Based on the constraints and flexibilities of real time applications, we define a performance measure, *message throughput* (τ_m), which is suitable for these applications. This measure differentiates the notion of optimum codes for the target multimedia applications as compared to performance measures that are used for non-realtime data. Based on the proposed measure, we combined the advantages of lowering the density of a code for near capacity performance with the high decoding efficiency of Reed Solomon (RS) codes, in order to design optimum PRS codes. Then, we demonstrated, through an example of a Binary Erasure Channel (BEC), that at near-capacity coding rates, appropriate design of a PRS code can outperform an RS-code. We extended this analysis and optimization for a general BEC over a wide range of channel conditions. Moreover, as compared with RS codes, the proposed PRS codes provide a significantly improved graceful degradation when the number of losses exceeds the number of parity symbols within the code block. This is a highly desirable feature for realtime multimedia

applications. . Our video simulation results outlined that the enhanced erasure recovery and graceful degradation in performances yields a profound improvement in the perceived media quality. In particular we defined a paradigm for a unique "fixed rate" adaptive FEC scheme based on PRS codes.

Future research direction of the presented work can be summarized as follows:

- More generalized constructions of codes.
- Designing of faster decoding algorithms and better decoding efficiency for short block-length codes.
- Increase gracefulness in performance degradation of the code and better adaptability to channel conditions.
- Generalization of current schemes to more generalized channels. Specifically design of codes for channels with memory and for channels with errors where the location of the error is not known.
- Optimizing the performance of codes from a multi-user point of view
- Extension the proposed work to prioritized data streams. In case of video, this would be equivalent to designing codes for multi-resolution streams.
- Explore possible applications in Joint-Source Channel Coding scenarios.

REFERENCES

[1] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," *ACM Computer Communication Review*, April 1997.

[2] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379-423 and 623-656, July and October 1948.

[3] R. W. Hamming, R. W. (1950, April). "Error Detecting and Error Correcting Codes," *Bell System Technical Journal*, vol. 29, pp. 235-237, April 1950.

[4] P. Elias, "Coding for two noisy channels. Information Theory," Third London Symposium, pp. 61-76, 1955.

[5] R. G. Gallager, "Low Density Parity-Check Codes," *MIT Press*, Cambridge, MA, 1963.

[6] I. S. Reed, G. Solomon, "Polynomial Codes over Certain Finite Fields," *J. Soc. Ind. Appl. Math.*, vol. 8, pp. 300-304, June 1960.

[7] S. A. Vanstone, P. C. van Oorschot, "An Introduction to Error Correcting Codes with Applications," *Kluwer Academic Publishers*, 1989.

[8] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, "Efficient Erasure Correcting Codes," *IEEE Transactions on Information Theory*, vol. 47, pp 569 – 584, February 2001.

[9] J.W. Byers, M. Luby, M. Mitzenmacher, "A Digital Fountain Approach to Asynchronous Reliable Multicast," *IEEE Journal on Selected Areas In Communications*, vol. 20, no. 8, pp 1528 – 1540, October 2002.

[10] J.W. Byers, M. Luby, M. Mitzenmacher, A. Rege, "A Digital Fountain Approach to Reliable Distribution of Bulk Data," *ACM SIGCOMM '98*, September 1998.

[11] J. Bloemer, M. Kalfane, M. Karpinski, R. Karp, M. Luby, D. Zuckerman, "An XOR-Based Erasure-Resilient Coding Scheme," *ICSI TR-95-048*, University of California, Berkeley, August 1995.

[12] A. McAuley, "Reliable Broadband Communication Using a Burst Erasure Correcting Code," *ACM SIGCOMM'90*, Philadelphia, PA, U.S.A, September 1990.

[13] J. C. Bolot, S. Fosse-Parisis, D. Towsley, "Adaptive FEC-based error control for Internet telephony," *Proceedings of IEEE INFOCOM '99*, vol. 3, pp. 1453-1460, March 1999.

[14] P. A. Chou, A. E. Mohr, A. Wang, S. Mehrotra, "Error control for receiver-driven layered multicast of audio and video," *IEEE Transactions on Multimedia*, vol. 3, pp. 108-122, March 2001.

[15] Y. Wang, Q. Zhu, "Error Control and Concealment for Video Communications: A Review," *Proceedings of the IEEE ICIP*, May 1998.

[16] Y. Wang, M. Hannuksela, V. Varsa, A. Hourunranta, M. Gabbouj, "The Error Concealment Feature in H.26L Test Model," *Proceedings of the IEEE ICIP*, September 2002.

[17] P. Bansal, M. R. Narendran, N. K. Murli Manohar, "Improved Error Detection and Localization Techniques for MPEG-4 Video," *Proceedings of the IEEE ICIP*, September 2002.

[18] H. Radha, M. van der Schaar, Y. Chen, "The MPEG-4 Fine-Grained Scalable Video Coding Method for Multimedia Streaming over IP," *IEEE Transactions on Multimedia*, March 2001

[19] M. van der Schaar, H. Radha, "Packet-loss resilient Internet video using MPEG-4 Fine Granular Scalability," *Proceedings of the IEEE ICIP*, September 2002.

[20] D. Loguinov, H. Radha, "End-to-End Internet Video Traffic Dynamics: Statistical Study and Analysis," *IEEE INFOCOM*, June 2002.

[21] M. Yajnik, J. Kurose, D. Towsley, "Packet Loss Correlation in the MBone Multicast Network," *Proceedings of IEEE Global Internet Conference*, November 1996.

[22] S. Khayam, S. Karande, H. Radha, and D. Loguinov, "Performance Analysis and Modeling of Errors and Losses over 802.11b LANs for High-Bitrate Real-Time Multimedia," to appear in *Signal Processing: Image Communication Journal*, 2003.

[23] S. Karande, S. Khayam, M. Krappel, and H. Radha, "Analysis and Modeling of Errors at the 802.11b Link Layer," to appear in *Proceedings of IEEE Conference of Multimedia and Expo (ICME)*, July 2003.

[24] R.E. Blahut, "Theory and practice of error control codes," *Addison-Wesley*, 1983.

[25] O. Pretzel, "Error Correcting codes and finite fields," *Oxford University Press*, 1992.

[26] S. Lin, JR. D. J. Costello, "Error Control Coding: Fundamentals and Applications," *Prentice Hall*, 1983.

[27] J. I. Hall, "Notes on Coding Theory," *http://www.mth.msu.edu/~jhall/.*

[28] M. Artin, "Algebra," *Prentice Hall*, 1991.

[29] R. W. Marsh, "Table of irreducible polynomials over GF(2) through degree 19," *NSA*, Washington, D.C., 1957.

[30] B. Bollobas, "Modern Graph Theory," *Springer*, 1998.

[31] R. Diestel, "Graph Theory," *Electronic Edition Springer-Verlag*, 2000.

[32] M. C. Davey, D. J. C. Mackay, "Low Density Parity Check Codes over GF(q)," *IEEE Communication Letters*, June 1998.

[33] T. Wiegand, G. Sullivan, "Draft ITU-T Recommendation and Final Draft international Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC)," *JVT of ISO/IEC MPEG & ITU-T VCEG 7th Meeting*, Pattaya, Thailand, March 2003.

[34] M.G. Luby, M. Mitzenmacher, A. Shokrollahi, D. A. Spielman, "Improved Low-Density Parity-Check Codes Using Irregular Graphs," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 585-598. February 2001.