Robust H.264/AVC Video Transmission in 3G Packet-Switched Networks

by

Katayoun Farrahi
B.A.Sc., University of Toronto, 2002

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

MASTER OF APPLIED SCIENCE

in the Department of Electrical and Computer Engineering

Supervisor: Dr. T.A. Gulliver

ABSTRACT

Real time video transmission in lossy packet switched environments is a challenging task requiring error recovery techniques that have very strict delay constraints. The mobile environment is characterized by harsh transmission conditions which can severely degrade the quality of a video stream. Many sophisticated radio link features are used to reduce bit error and packet loss. However, these techniques may not be sufficient to provide acceptable video quality.

The main objective of this work was to provide a set of error correction tools which are likely to be used in wireless environments with a focus on the most challenging application, wireless conversational services. Third generation (3G) Wideband-CDMA (W-CDMA) networks were used for simulating channel conditions. For transmission, RTP packetized H.264/AVC video, including some error resilience and concealment tools, were used. The RTP packet loss rate was reduced significantly with the use of forward error correction coding, specifically Reed Solomon coding, applied to the link layer frames. This was found to be an effective way of reducing packet loss and increasing decoded PSNR without increasing the overall system delay. A significant reduction in packet loss was obtained by combining Reed Solomon coding with interleaving. However, this increased the overall delay and this is more suitable for streaming applications.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

2.5G Second and a Half Generation

3G Third Generation

3GPP Third Generation Partnership Project

AVC Advanced Video Coding

AWGN Additive White Gaussian Noise

BER Bit Error Rate

CABAC Context Adaptive Binary Arithmetic Coding

CRC Cyclic Redundancy Check

DSL Digital Subscriber Line

ETSI European Telecommunications Standards Institute

FQI Frame Quality Indicator

GSM Global System for Mobile Communications

HD High Definition

IEC International Electrotechnical Commission

IP Internet Protocol

ISDN Integrated Services Digital Network

ISO International Organization for Standardization

ITU-T International Telecommunications Union Telecommunication Standardization Sector

LAN Local Area Network

LTU Logical Transmission Unit

MMS Multiple Messaging Services

MPEG Moving Picture Experts Group

NAL Network Abstraction Layer

PCS Packet-switched Conversational Services

PDU Protocol Data Unit

PSS Packet-switched Streaming Services

RGB Red/Green/Blue colour space

RLC Radio Link Control

RLP Radio Link Protocol

RTP Real-time Transport Protocol

SD Standard Definition

SDP Session Description Protocol

SDU Service Data Unit

SEI Supplementary Enhancement Information

SIP Session Initialization Protocol

TCP Transmission Control Protocol

UDP User Datagram Protocol

UMTS Universal Mobile Telecommunications System

VCEG Video Coding Experts Group

VCL Video Coding Layer

W-CDMA Wideband Code Division Multiple Access

Acknowledgement

I would first like to thank my supervisor, Professor Aaron Gulliver, for all his guidance, assistance and time.

I would also like to acknowledge the financial support of NSERC and Sierra Wireless Inc., and the technical assistance of Sierra Wireless principal engineer Pete McConnell.

Many thanks to Houman, Shirin, and my parents, for all of their support. Last but not least, thank you to my friends in Electrical Engineering at UVic and UBC.

Chapter 1

Introduction

Work on the emerging Advanced Video Coding (AVC) standard, now known as ITU-T Recommendation H.264 and as ISO/IEC 14496 (MPEG-4 part 10), has dominated the video coding standardization community for roughly the past four years [2]. The development of H.264/AVC video coding standard has been carried out by the Joint Video Team, a joint collaboration of the ISO/IEC and ITU-T standards organizations. The MPEG-2 video coding standard (also known as ITU-T H.262), which was developed about 10 years ago, was an enabling technology for digital television systems worldwide. It is widely used for the transmission of Standard Definition (SD) and High Definition (HD) TV signals over satellite, cable, and the storage of video signals onto DVD.

However, an increasing number of services are creating greater needs for higher coding efficiency. Moreover, enhanced coding efficiency can enable the tranmission of more video channels or higher quality video within existing digital transmission capacities. Video coding for telecommunication applications has diversified from wireline networks, such as ISDN, to mobile wireless networks, LANs and Internet network delivery. Throughout this evolution, efforts have been made to maximize coding efficiency while dealing with the diversification of network types and their characteristic formatting and loss/error robustness requirements [3].

The new H.264/AVC standard is designed to provide a technical solution appropriate for a broad range of applications, including:

- Broadcast over cable, satellite, cable modem, DSL, terrestrial.

- Interactive or serial storage on optical and magnetic storage devices, DVD, etc.
- Conversational services over ISDN, Ethernet, LAN, DSL, wireless and mobile networks, modems.
- Video-on-demand or multimedia streaming services over cable modem, DSL, ISDN, LAN, wireless networks.
- Multimedia messaging services over DSL, ISDN.

The range of bit rates and picture sizes supported by H.264/AVC is extensive, addressing video coding capabilities ranging from very low bit rate resolution video for mobile and dial-up devices, through to entertainment quality standard definition television services, and beyond.

1.1 3G W-CDMA (UMTS) Networks

The Universal Mobile Telecommunications System (UMTS), or Wideband-CDMA (W-CDMA), is an air interface standard for third generation (3G) wireless telecommunications that has evolved since late 1996 under the European Telecommunications Standards Institute (ETSI). In 1998, UMTS was submitted by ETSI for consideration as a world standard. Around the turn of the century, several other competing W-CDMA proposals agreed to merge into a single W-CDMA standard, and this resulting standard is now called UMTS.

UMTS assures backward compatability with many of the second generation and 2.5G technologies, such as GSM. The new W-CDMA air interface can provide additional capacity and bandwidth compared to second generation systems. Today, W-CDMA is the primary focus of the 3GPP world standard body, and while ETSI remains the organizational body that coordinates the W-CDMA standards effort, W-CDMA development now involves leading manufacturers, carriers, engineers, and regulators from around the world within the 3GPP community. W-CDMA is developing for both wide area mobile cellular coverage as well as indoor cordless type applications. By 2010, it is likely that W-CDMA will be fully installed [4].

1.2 Significance of Research

Video transmission for mobile terminals is likely to be a major application in emerging 3G systems and may be a key factor in their success. In general, the available bandwidth and therefore the bit rate over the radio link are limited, and the costs for a user are expected to be proportional to the reserved bit rate or the number of transmitted bits over the radio link. Thus, low bit rates are likely to be typical, and compression efficiency a key requirement, making H.264/AVC a prime candidate for the use in wireless systems.

In addition, the mobile environment is characterized by harsh transmission conditions in terms of attenuation, shadowing, fading, and multi-user interference, which result in time and location varying channel conditions. Many highly sophisticated radio link features such as diversity, space-time coding, multiple antenna systems, and forward error correction, just to name a few, are used in 3G systems to reduce variations in channel conditions. However, only for fast moving users and relatively large tolerated maximum delay can these techniques provide a negligible bit error and radio packet loss rate [5]. Therefore, in addition to high compression efficiency and reasonable complexity, a video coding standard for conversational services in wireless environments has to be error resilient.

In this thesis, forward error correction using Reed-Solomon codes is applied to the link layer frames of RTP packetized H.264/AVC video to reduce packet loss with minimal delay. Error resilience and concealment tools are also used to reduce the visual degradation in quality where packet loss occurs. Finally, interleaving is applied to significantly reduce packet loss for applications with less strict delay constraints, such as streaming services.

1.3 Thesis Outline

This thesis is organized as follows. Chapter 2 provides some basic background information on video compression, and video signals. Chapter 3 presents an overview of the H.264/AVC video coding standard, the packetization of H.264/AVC video into RTP pack-

ets, as well as the protocol environment and transport of H.264/AVC video over 3G W-CDMA networks. The error control techniques applied to reduce packet loss are covered in Chapter 4, including Reed-Solomon coding, interleaving, and the error resilience and concealment techniques employed. Chapter 5 discusses the simulation environment and the results obtained for reducing packet loss and improving the decoded PSNR. Finally, Chapter 6 presents the conclusions and some suggestions for future work.

Chapter 2

Properties of Video Signals

This chapter examines the structure and characteristics of video signals and introduces concepts such as sampling formats and quality metrics that are helpful to understand video coding.

2.1 Representation of Video Signals

For each instant in time, an uncompressed colour video frame can be thought of as three two dimensional (2-D) arrays. Each of these 2-D arrays contains intensity values, in the form of pixels, for a particular colour: red, green, or blue. This colour format is called RGB. The combination of these colour components in different variations can produce any other colour. Figure 2.1 illustrates the arrays of colour components [6].

For convenience, this multidimensional signal is represented in two dimensions. For each 3 colour pixel vector, the red component is followed by the green then the blue. These 3 components are read in raster scan order (from left to right and top to bottom) as shown in Figure 2.2.

2.1.1 The YCbCr Colour Format

The RGB colour format is the most well known, but it is not the most suitable for video coding. It is not very efficient since it uses equal bandwidths to represent all three colour components. The human visual system perceives scene content in terms of brightness

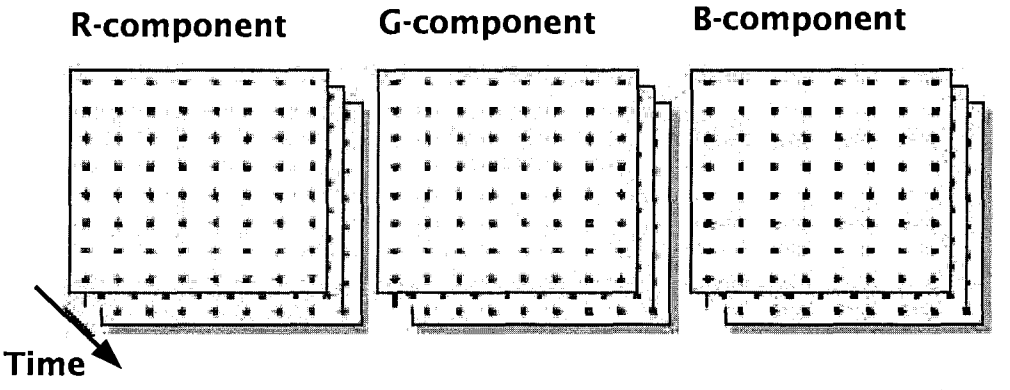

Figure 2.1. *Video Signal Representation*

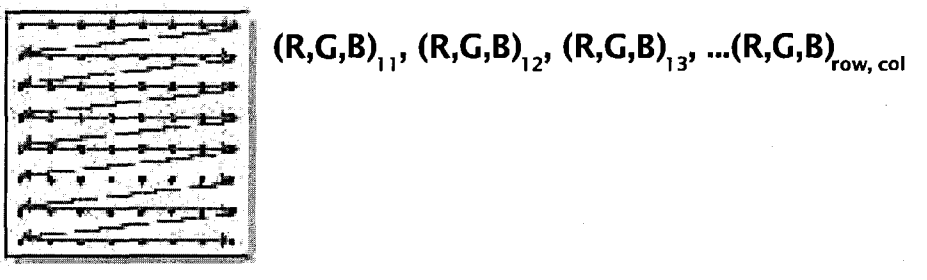

Figure 2.2. *Two Dimensional Signal Representation*

and colour information separately, and with greater sensitivity to the details of brightness than colour. Video transmission systems can be designed to take advantage of this. In H.264/AVC, as in prior video coding standards, this is done by using a YCbCr colour space together with a reduced sampling resolution of the Cb and Cr chroma information [3].

The YCbCr colour format, which is derived from the RGB format, is usually used for video coding. The Y component is the black and white component, called the luminance (or luma). The human visual system perceives luma as brightness. The Cb and Cr components, called the chrominance (or chroma) represent the extent to which the colour deviates from grey towards blue and red, respectively. The conversion from RGB to YCbCr can be performed as follows:

RGB to YCbCr conversion:

$Y = 0.299R + 0.587G + 0.114B$

$Cb = 0.564(B - Y)$

$Cr = 0.713(R - Y)$

YCbCr to RGB conversion:

$R = Y + 1.402Cr$

$G = Y - 0.344Cb - 0.714Cr$

$B = Y + 1.772Cb$

2.1.2 4:2:0 Sampling

Since the eye is less sensitive to the chroma gradient than it is to the luma gradient, it is sampled at a lower rate to reduce bandwidth requirements. A common sampling structure for video is the 4:2:0 sampling format where the chroma sampling rate is half the luma sampling rate in both horizontal and vertical directions [7]. With this sampling, chroma component will have one fourth the number of samples of the corresponding luma component.

Each sample has 8 bits of precision, so 1 byte of data is used to represent each colour

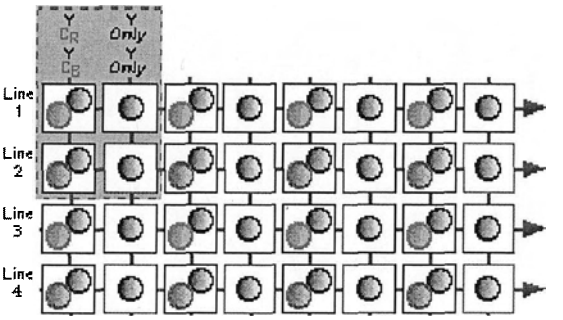

Figure 2.3. *4:2:0 Sampling: Chroma and Luma Components*

component. Thus, there is a scale of 0 to 255 to represent each individual luma or chroma component.

2.1.3 Video Formats

Video compression standards, such as H.264/AVC, can compress a wide variety of video frame formats. The Common Intermediate Format (CIF) for H.264 video conferencing is the basis for a popular set of formats. The choice of frame resolution depends on the application and availabe storage or transmission quality. Quarter Common Intermediate Format (QCIF) is popular for videoconferencing applications as well as mobile multimedia applications where the display resolution and the bitrate are limited. It is a videoconferencing format that specifies data rates of 30 frames per second (fps), with each frame containing 144 lines and 176 pixels per line. Therefore, there are 144x176 luminance pixels and 2x72x88 chrominance pixels. The uncompressed bit rate at 30fps for QCIF picture format is 9.124Mb/s (= 30frames/s * ((176x144) + 2(72x88))pixels/frame * 8bits/pixel).

2.1.4 Picture Organization

A coded video sequence consists of a sequence of coded pictures or frames. A picture is partitioned into fixed sized macroblocks that cover a rectangular picture area. Macroblocks are the basic building blocks of video compression. The size of a macroblock varies from

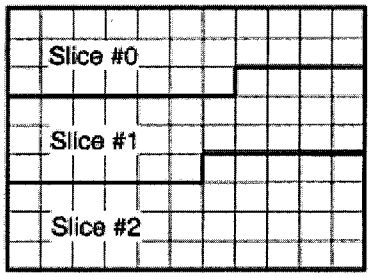

Figure 2.4. *Subdivision of a Picture into Slices*

one video compression standard to the next. However, a macroblock consists of a rectangular array of luminance (Y) samples and two corresponding array of chrominance (Cb and Cr) samples. The macroblock size for H.264/AVC is given in Subsection 3.1.1. They are addressed in raster scan order within a frame.

The macroblocks are organized into slices, which represent subsets of a given picture that can be decoded independently. A video picture is coded as one or more slices. A slice consists of an arbitrary number of successive macroblocks. There is minimal interdependency between coded slices which can help to limit the propagation of errors. A picture may be split into one or several slices as shown in Figure 2.4. Slices are self-contained; the values of the sample in the area of the picture that the slice represents can be correctly decoded without the use of data from other slices.

2.2 Frame Rate and Bit Rate

Video signals are composed of a sequence of successive still images called frames. The perception of moving objects is obtained by projecting in time a sequence of frames containing graphical objects which vary slightly in position between frames. The frame rate is the number of frames projected per second.

Two conditions must be met in order to represent a visual reality through a sequence of video frames. First, the rate of repetition of the images must be high enough to ensure continuity of movements (smooth transition) from frame to frame. That is, the frame rate

must be high enough so that an object will appear to move smoothly to the human eye. Second, the rate must be high enough that continuity of perception is not disturbed by the dark intervals between frames. Continuity of perception is achieved when the viewer does not see the black spaces between frames. This is refered to as flickering. For video, a minimum of 10-15 frames per second (fps or Hz) is required for minimal motion perception [8] [6]. Smooth video motion is achieved at 30fps [9]. In order for the human eye not to perceive any flicker between video frames, 50fps must be transmitted [8]. However, normal frame rates for standard broadcast video systems are 25 or 30fps; technical measures are used to allow lower refresh rates.

The bit rate, also known as the coding rate, is an important parameter in video compression and is usually expressed in bits per second. The bit rate in the H.264/AVC video codec is defined as:

Bit rate = (total number of bits) * (frame rate) / (number of frames transmitted)

2.3 Types of Redundancy in Video

Compression essentially identifies and eliminates redundancies in a video signal and provides instructions for reconstituting the bit stream into a picture when the bits are uncompressed. The basic types of redundancy are spatial, temporal, psycho-visual, and statistical.

Spatial redundancy represents the statistical correlation between pixels within an image frame. It is called intra-frame redundancy since it is within a frame. Spatial redundancy implies that the intensity value of a pixel can be approximated from the intensity of its neighbouring pixels. Techniques to exploit spatial redundancy are:

-Transform Coding

-Predictive Coding

Temporal redundancy is concerned with the statistical correlation between pixels from successive frames in a video sequence. Therefore, it is called inter-frame redundancy.

Video compression techniques depend heavily on the removal of temporal redundancy to achieve high compression ratios. Techniques to exploit temporal redundancy are:

-Motion Compensation

-Predictive Coding

Psycho-visual redundancy originates from the characteristics of the human visual system (HVS). One example of psychovisual redundancy is 4:2:0 sampling. Since the human eye is less sensitive to changes in chroma, it is sampled at a lower rate. Techniques to exploit psycho-visual redundancy are:

-Sampling

-Quantization

Finally, statistical redundancy uses a more compact representation for elements that frequently recur in a bitstream, thus reducing the overall size of the compressed signal. Entropy coding techniques are typically employed.

Modern video compression techniques follow a common set of operations. First, they segment the video frame into blocks of pixels called macroblocks. Then, intra or inter prediction is applied. Following this, an algorithm based on DCT (discrete cosine transform) decorrelates the motion-compensated data as well as the intra-coded data to produce an expression with the lowest number of coefficients. The video-compression scheme then quantizes the DCT coefficients based on a psycho-visual redundancy model. Source coding then removes statistical redundancy, reducing the average number of bits necessary to represent the compressed video.

2.4 Picture Quality Assessment

In order to specify, evaluate and compare video communication systems, it is necessary to determine the quality of the video images displayed to the viewer. Measuring video quality is a difficult and often imprecise art because there are so many factors that can

affect the results. Visual quality is inherently subjective and is influenced by many factors that make it difficult to obtain a completely accurate measure of quality. Measuring visual quality using objective criteria gives accurate, repeatable results but as yet there are no objective measurement systems that completely reproduce the subjective experience of a human observer watching a video display.

In subjective visual quality measurement, observers are asked to evaluate the visual quality of a video sequence. For objective quality evaluation mean square error (MSE) and peak signal to noise ratio (PSNR) are used. For video and images, PSNR is the most widely used. MSE and PSNR are given by

$$MSE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (f(x,y) - g(x,y))^2,$$

and

$$PSNR = 10 \log_{10}(\frac{255^2}{MSE}),$$

respectively, where M and N are the dimensions of the video frame in width and height, respectively, and f and g are the original and reconstructed pixel luminance or chrominance values at position (x, y). Typically, the PSNR calculation is shown for the luminance samples only, since the luma component is perceived most by the human visual system and forms the fundamental part of the picture. Variations in the picture colour are not as noticeable by the human eye. However, it is also possible to find the PSNR of the Cb or Cr chrominance samples.

In this thesis, luma PSNR is used for quality assessment at the encoder. However, since real-time transport protocol (RTP) packetized video is transmitted, packet loss and decoded luma PSNR are used as an evaluation of quality at the receiver.

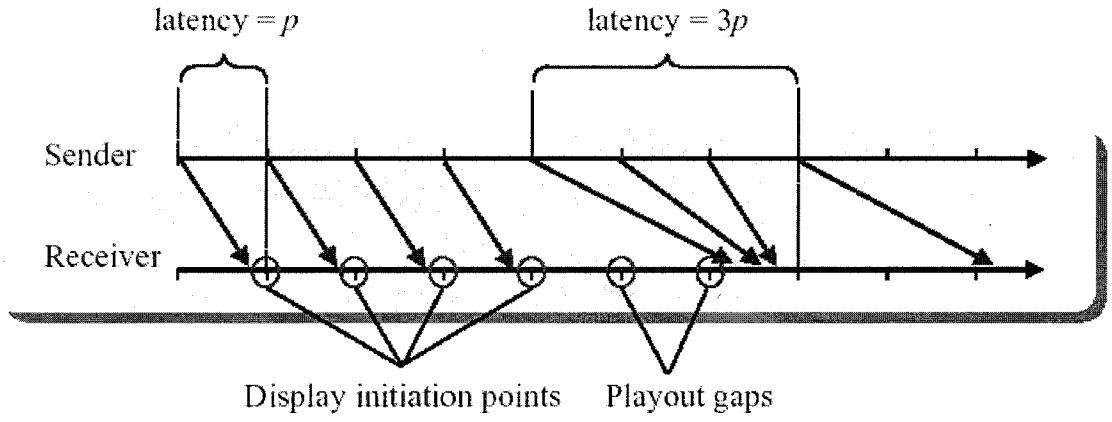

Figure 2.5. *Delay Jitter*

2.5 Performance Requirements

The following parameters play a vital role in enabling the transmission of video over a digital network.

1) *Latency:* End-to-end delay, or latency, is the time duration between the acquisition of a signal and its display. The latency requirement depends on the application and is discussed further in later chapters. For real-time applications, less than 250ms one-way delay is generally necessary [6].

2) *Delay jitter:* Delay jitter is variation in end-to-end latency, as shown in Figure 2.5. It leads to "gaps "in the playout of media and increases playout latency. Playout is the viewing (or playing) of received video packets.

3) *Error Rate:* Another important paramater for multimedia networks is the error rate. This can be defined in a number of ways. One is bit error rate (BER). Another is packet error rate or packet loss, which is the parameter considered in this thesis since erroneous packets are typically dropped by receivers. BER is typically used when working on the physical layer.

4) *Throughput:* The throughput of a network is its effective bit rate, or effective bandwidth. Throughput is defined to be the physical link bit rate minus the overheads

that pertain to the transmission technologies employed [8]. In many instances, the overhead is deemed to be implicit, and throughput is simply the bit rate of the system [8].

Chapter 3

H.264/AVC Video in 3G Packet-Switched Networks

This chapter presents background information on H.264/AVC video, third generation (3G) protocol environments for packet-switched networks, as well as the packetization and transport mechanism of video over these networks.

3.1 Overview of the H.264/AVC Video Coding Standard

H.264/AVC is the current video standardization project of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG). The H.264/AVC video coding standard is known as ITU-T Recommendation H.264 and ISO/IEC International Standard 14496 Part 10 (or MPEG-4 Advanced Video Coding). The main goals of this standardization effort are to develop a simple video coding design, with enhanced compression performance, and to provide improved network adaptation to address both "conversational "(video telephony) and "non-conversational "(storage, broadcast, or streaming) applications. H.264/AVC consists of a video coding layer (VCL) and a network abstraction layer (NAL). The interface between the VCL and the NAL is conceptual and helps in describing and separating the tasks of the VCL and the NAL [3] [10].

3.1.1 H.264/AVC VCL

The VCL is designed to efficiently represent the video content. It performs signal processing tasks and generates bit strings containing coded macroblocks. The VCL has been designed to be as network independent as possible.

Each picture of a video sequence is partitioned into fixed sized macroblocks that cover a rectangular picture area of 16x16 samples of the luma component and 8x8 samples of each of the two chroma components.

The VCL design follows the *block-based hybrid* video coding approach. The basic source coding algorithm is a hybrid of *inter-picture prediction*, to exploit the temporal statistical dependencies, and *intra-picture coding* to exploit the spatial statistical dependencies [11]. The VCL encoder outputs slices. Each slice can be coded using different coding types as follows:

- I slice: A slice in which all macroblocks of the slice are coded using intra prediction.
- P slice: In addition to the coding types of the I slice, some macroblocks of the P slice can also be coded using inter prediction with at most *one* motion compensated prediction signal per macroblock.
- B slice: In addition to the coding types available in a P slice, some macroblocks of the B slice can also be coded using inter-picture prediction with *two* motion compensated prediction signals per macroblock.

SP and SI slices are two other slice types which are new in H.264/AVC. They are specially coded slices that enable efficient switching between video streams and efficient random access for video decoders [2]. A common requirement in a streaming application is for a video decoder to switch between one of several encoded streams. For example, the same video material can be coded at multiple bitrates for transmission across the Internet. A decoder can attempt to decode the highest bitrate stream it can receive, but may require switching automatically to a lower bitrate stream if the data throughput drops. Readers are referred to [12] for more information on these slice types.

All luma and chroma samples of a macroblock are either spatially or temporally predicted. In intra-picture coding mode, the macroblocks are first processed by a transform for bit rate reduction. The resulting coefficients are quantized and entropy coded. An optional rate control algorithm is used to adapt the quantizer step size or decide the number of frames to be skipped. In inter-picture prediction mode, the same operations are applied to the motion predicted difference between the current macroblocks of a frame and those of a previous frame. A deblocking filter can be used to reduce the visual artifacts of encoding.

3.1.2 H.264/AVC NAL

The network abstraction layer defines the interface between the video codec itself and the outside world. It is designed to provide "network friendliness "to enable simple and effective customization of the VCL for a broad variety of systems. The NAL encoder encapsulates the slice output of the VCL encoder into network abstraction layer units (NAL units), which gives support for the packet-based approach of most existing networks. At the NAL decoder interface it is assumed that the NAL units are delivered in transmission order and that packets are either received correctly, are lost, or an appropriate error flag in the NAL unit header is set if the payload contains bit errors.

Each NAL unit is effectively a packet that contains an integer number of bytes. The first byte of each NAL unit is a header byte that contains an indication of the type of data in the NAL unit. The remaining bytes contain payload data representing the macroblocks of a slice [3]. The header byte itself consists of an error flag, a NAL unit flag, and the NAL unit type [13]. The NAL unit flag can be used to signal whether that NAL unit is used as a reference for inter picture prediction or not. It is used to reduce the effects of error propagation. The NAL payload type indicates the VCL "slice type "used, e.g. Intra slice (or I-slice), P-slice, or B-slice [13].

A NAL unit specifies a generic format for use in both packet-oriented and bitstream systems. Systems using packet networks can employ NAL units directly by mapping them into RTP packets. The details of RTP packetization are covered in the following section. In

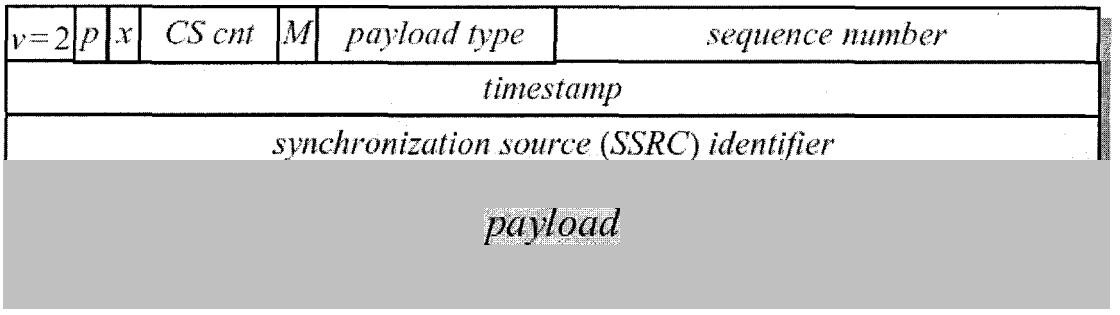

Figure 3.1. *RTP Packet*

stream-oriented formats the NAL units are encapsulated by start codes. Otherwise, the format of NAL units for both packet-oriented transport and bitstream delivery is identical. The start code prefix is a specific pattern of two or three bytes used to identify the boundaries of the NAL unit.

3.2 H.264/AVC RTP-Packetization

The real-time transport protocol (RTP) provides end-to-end delivery services for data with real-time characteristics. The RTP packetization of H.264/AVC using the NAL follows the general principles outlined in RFC1889 [14] and are described in this section. The H.264/AVC video codec outputs RTP video packets in the following format. Each RTP packet consists of an RTP header and the payload, as shown in Figure 3.1. The RTP header contains the following:

Version (V): A 2 bit version number, which is equal to 2 for RFC 3550 [15][16]. It identifes the version of RTP being used. The value of 1 is used for the first draft version of RTP [14].

Padding (P): A padding bit, which is either 1 or 0 (yes or no) according to whether padding bits are used or not [6][16]. Padding may be needed by some encryption algorithms.

Extension (X): The extension bit is either 1 or 0 (yes or no) according to whether optional

payload headers are used or not [6].

Contributed Source count (CS cnt): The CS cnt is a 4 bit value between 0-15 indicating the number of CSRC identifiers that follow the fixed header. This is 0 in our case.

Marker Bit (M): The marker bit usually indicates the end or a group of packets with the same timestamp. It can be used to quickly detect the end of a group of related packets without having to wait for the next packet to arrive.

Payload Type: The payload type is 7 bits, and it identifies the media codec of the payload.

Sequence Number: The sequence number is 16 bits. It is incremented by one for each packet sent in a session and is used for packet-loss detection. The sequence number can also be used to remove duplicate packets and reorder packets.

Timestamp: The 32 bit timestamp contains timing information relative to the establishment of the session. It is used for the synchronization of more than one media stream, e.g. to provide synchronization between audio and video. The timestamp is normally the sampling instant of the last bit of media in the RTP payload [13]. It is not an absolute time, but coded relative to a random offset that is chosen during the session establishment. A clock rate of 90KHz must be used [16].

Synchronization source (SSRC) identifier: The SSRC is a randomly chosen 32-bit ID which identifies a source of RTP packets for a given session. This identifier is chosen randomly, with the intent that no two synchronization sources within the same RTP session will have the same SSRC identifier.

3.3 Video in Mobile Networks

Video transmission for mobile terminals is likely to be a major application in emerging 3G systems and may be a key factor in their success [5]. Three major service catergories were identified in the H.264/AVC standardization process [5] and are illustrated in Figure 3.2.

1) Circuit-switched and packet-switched conversational services (PCS) for video telephony

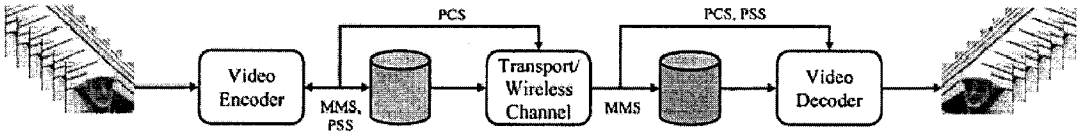

Figure 3.2. *Wireless video applications: MMS, PSS, and PCS*

and conferencing.

2) Live or pre-recorded video packet-switched streaming services (PSS).

3) Video in multimedia messaging services (MMS).

The transmission requirements for the three identified applications can be distinguished with respect to data rate, maximum end-to-end delay, and maximum delay jitter.

MMS does not include any real-time constraints since encoding, transport, and decoding are completely separate. The video signal is recorded and stored prior to transmission. The decoding process at the receiver is not generally started until the complete video sequence has been downloaded.

In PSS applications, the user typically requests pre-recorded sequences, which are stored at the server. Whereas encoding and transmission are separate, decoding and display are started during transmission to minimize the initial delay and memory usage in mobile devices. The allowable end-to-end delay is variable, but is on the order of seconds or tens of seconds [6].

Finally, in conversational applications, the end-to-end delay has to be minimized to avoid perceptual disturbances and to maintain synchronicity of audio and video. Encoding, transmission, and decoding is performed simultaneously in real-time. Conversational video transmission is by far the most challenging of the three applications identified here. Stockhammer et al. [5] restrict the end-to-end delay for conversational services to less than 250ms.

For real-time video services over 3G mobile networks, two protocol stacks are of major interest. A protocol stack is a set of network protocol layers that work together. 3GPP has specified a multimedia telephony service for circuit-switched channels based on ITU-T

Recommendation H.324M [5]. For IP-based packet-switched communication, 3GPP has chosen to use SIP and SDP [17] for call control and RTP for media transport. While the H.324 and the RTP/UDP/IP stacks have different roots and a completely different switching philosophy, the loss and delay effects on the media data when transmitting over wireless dedicated channels are very similar. Dedicated channels are those in which one user gets assigned a fixed data rate for the entire transmission interval.

For packet-switched conversational and streaming applications there is only one single commonly used protocol hierarchy which is discussed in Sections 3.4 and 3.5.

This thesis will mainly deal with conversational and streaming video, for both live and prerecorded applications. Messaging applications are much easier since there are no time constraints, and can use protocols which are used for conventional data applications.

3.4 Protocol Environment

Video over IP is usually implemented either by downloading complete bit streams, or by real-time transmission. The latter one for conversational or streaming applications over IP networks usually employs IP on the network layer, UDP on the transport layer, and RTP and accompanying RTP payload specifications on the application layer. IP and UDP together offer an unreliable datagram transport service. RTP adds to this functionality a few features that make the transport of media possible.

1) *Physical and Link Layer:* IP networks may operate over a variety of physical and link layer protocols and are generally designed to abstract from those underlying protocols. The physical layer specifies the wireless transmission technology. In our simulation model, we will be using 3G WCDMA standards. The link layer provides medium access control, which includes best effort delivery (retransmission), optional quality-of-service (QoS) control, resource allocation between services and multiplexing of the traffic channels onto the physical layer channels [18].

2) *Network Layer:* On the network layer, IP networks use the Internet Protocol (IP). IP packets are transported individually from the sender through a set of routers to the receiver. The transmission time of a packet varies from packet to packet. Routers can discard packets at any time, resulting in packet loss. The IP header size is 20 bytes and is protected by a checksum. No protection of the payload is performed. The maximum size of an IP packet allowed by the protocol specification is 64KBytes [10], however, this size is rarely used. IP packets are susceptible to network congestion in routers. The network layer's main tasks are routing of packets and congestion (or flow) control [18]. Network congestion occurs when packets arrive too fast for the router to process so that fills its buffer. Routers will try to discard or reroute any incoming packets for which there is no available buffer space. Rerouting results in delay jitter and out-of-order packet delivery [19].

3) *Transport Layer:* On the transport layer, IP networks commonly employ two protocols: TCP [20] and UDP [21]. TCP offers a byte-oriented, guaranteed transport service, which is based on re-transmission and timeout mechanisms for error control. It is not suitable for real-time communications due to its delay characteristics. UDP offers a simple, unreliable, datagram transport service. The UDP header contains a checksum, which can be used to detect and remove packets containing bit errors. Packets may be lost, duplicated, or re-ordered on their way from the source to destination. The transport layer's main tasks are:

- segmentation and packetization of the video stream in a format suitable for the network layer,
- multiplexing of the packets of the video connection with packets from other applications, and
- flow control, if needed at this stage [18].

The UDP header size is 8 bytes.

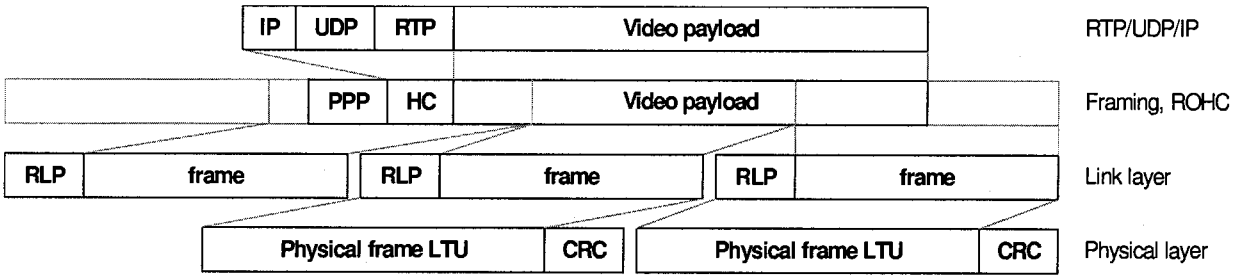

Figure 3.3. *Packetization through the user protocol stack.*

4) *Application Layer Transport:* The real-time transport protocol (RTP) is implemented in the application layer, above IP/UDP. RTP provides end-to-end delivery services for data with real-time characteristics. It is session oriented, and a session is associated with a transport address. The RTP header size is 12 bytes. Section 3.2 describes the RTP packet format and its components in further detail.

5) *Application Layer Control Protocols:* Control protocols, such as H.245, SIP with SDP, or RTSP, are used to announce the availability of a media stream, to establish the connection, to negotiate the capabilities of the sender/receiver(s), and to control a running session [10].

3.5 Transport of H.264/AVC Video in UMTS

This section gives a basic overview of the 3GPP air interface user plane protocol. 3GPP defines the user plane protocols for the UMTS specification between the mobile station and the radio base station. Figure 3.3 shows the application packets in the user plane protocol stack.

A physical layer frame can be divided into a number of logical transmission units (LTU). The LTU is the smallest unit with a cyclic redundancy check (CRC) to detect possible bit errors. The LTU size is fixed for a given session. It is assumed that one link layer frame, called a protocol data unit (PDU), is packed into one physical layer LTU. This means that each link layer frame has a frame quality indicator (FQI) that indicates if there are bit

errors detected by the physical layer CRC. The link layer frame is thus an appropriate unit for forward error correction.

Some extra header information is added to every RTP/UDP/IP packet, and this new packet is called a service data unit (SDU). Video packets output by the video codec in RTP packetized format can vary in length, resulting in variable length SDUs. However, the PDU size is fixed and is typically smaller than the SDU size. The radio link control (RLC) layer segments the SDUs into several PDUs. Figure 3.3 uses RLP instead of RLC. This is simply the CDMA-2000 terminology for RLC. Since an SDU may not fit into an integer number of PDUs, more than one SDU associated with more than one RTP packet may fill a PDU.

During transmission, if one or more PDUs (or frames) associated with an SDU (or RTP packet) are received in error, the entire RTP packet (or SDU) is discarded. Since more than one SDU may fill a PDU, a single frame error may result in the loss of more than one RTP packet. It is important to select appropriate-sized SDUs that match the channel conditions. For error prone channels, the SDU size is minimized considering the overhead associated with each SDU. The SDU size can be maximized to minimize overhead for error-free channels.

The RLC layer can perform re-transmissions depending on the protocol mode selected. The protocol can operate in the following three modes:

1) transparent,

2) unacknowledged, and

3) acknowledged mode [22].

The transparent and unacknowledged modes are defined to be unidirectional; the acknowledged mode is bi-directional. In the transparent mode no protocol overhead is added to higher layer data. Erroneous PDUs can be discarded or marked erroneous. In the unacknowledged mode, no retransmission protocol is in use and data delivery is not guaranteed. Received erroneous data is either marked or discarded depending on the configuration. The functionality of the transparent mode is similar to unachnowledged mode but no proto-

col information is appended to the PDU. In the acknowledged mode, an automatic repeat request mechanism is used for backward error correction. The maximum number of retransmissions is specified for a connection [22].

Chapter 4

Error Control Techniques

Video communication systems need to be robust in order to transmit good quality video streams. Compressed video sequences need to be protected for transmission over noisy wireless channels. There are two types of errors which can occur in a wireless network, bit errors and packet loss. Bit errors occur over circuit-switched networks, whereas both bit errors and packet loss occurs over packet-switched networks. Most applications transport over packet-switched networks, and thus, errors due to packet loss are considered in this thesis. However, all of the work done in this thesis could be applied to circuit-switched networks.

Errors due to packet loss can cause significant degradation to a video stream. Due to the motion compensation techniques used in video compression, packet loss can propagate resulting in poor video quality of many decoded frames. This impact of transmission errors is called spatio-temporal error propagation, and is shown in Figure 4.1. Further, loss of video header information can result in the complete loss of a segment of video, or perhaps complete loss of the entire video sequence.

The basic mechanisms available to recover from packet loss or corruption of data packets are error resilience and concealment techniques, forward error correction (FEC), and automatic repeat request (ARQ). In ARQ systems, the receiver simply detects whether the transmitted packet includes errors or not. When errors are detected, a retransmission request is transmitted to the sender. In this system, we can achieve error-free conditions if we can accept long transmission delay time. On the other hand, the idea of FEC is to get

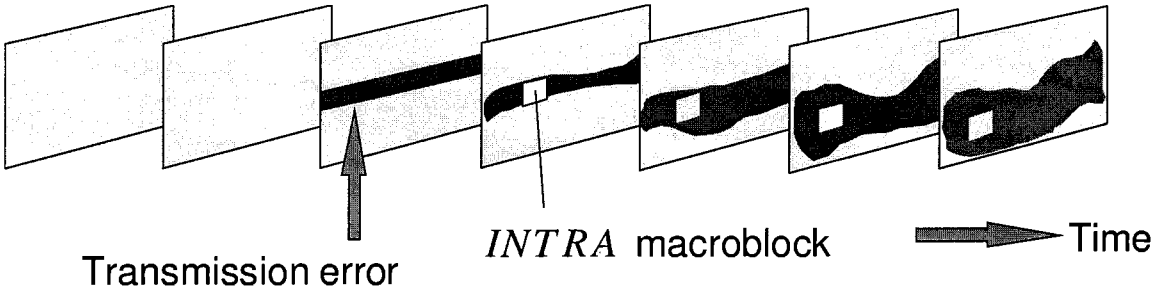

Figure 4.1. *Spatio-Temporal Error Propagation*

the transmission right the first time. FEC transmits, with the original data, some redundant data, called parity, to allow reconstruction of lost packets at the receiver. Error resilience and concealment techniques are built into the video codec itself. They are used to reduce the visual effects of data loss in the decoded video stream. The appropriate choice of error control technique to apply is dependent on the given application.

Conversational services are limited by low bit rates ($< 100kbit/s$) and low delay requirements ($< 250ms$) [23]. A method for improvement of video quality in real-time applications are to reduce the number of errors by using forward error correction (FEC). Another method is to mitigate the impact or errors using error resilience and concealment. For real time applications, retransmissions are not used due to the strict delay requirements [23].

Video streaming services have medium bit rates ($100 - 300kbit/s$) and high delays can be tolerated (2-3s) [23]. Methods for improving the received video quality are similar to those for conversational services, except retransmissions can be used to reduce the number of errors.

In this thesis, we focus on FEC combined with interleaving (which will be discussed in Section 4.4) to reduce the effect of burst errors. This can be applied to both conversational and streaming applications. Error concealment is used at the decoder. Error resilient tools, including intra macroblock refresh, periodic insertion of I-frames, and multiple reference pictures, are also used to reduce the effects of error propagation. These error control

techniques are further explained in this chapter.

4.1 Error Resilience and Concealment Techniques

A variety of techniques have been proposed to enhance the robustness of video communication systems to packet loss. Error resilience is the ability to protect video data against network losses in the video encoding/decoding process. Error concealment is the ability to make the errors induced less visible to the human eye. The use of multiple reference pictures, INTRA picture coding, and INTRA macroblock refresh, can mitigate the effects of error propagation. Apart from the error resilience tools mentioned, error concealment can lead to further improvements in video quality.

It is widely recognized that intra-coding is an important tool for mitigating the effects of packet loss. By using intra prediction for macroblocks of a given frame, the reproduced macroblocks are no longer dependent on past frames and error propagation is stopped. However, the robustness provided by intra coding may be costly, as it requires a higher bit rate than inter coding. Too many intra coded macroblocks will significantly degrade the compression performance. By switching between intra coding and inter coding, the right balance between compression efficiency and robustness can be achieved. Periodic intra coding of random macroblocks can also be used to reduce the effects of error propagation. This technique is called intra macroblock refresh. By periodically using intra coded macroblocks, error propagation can be reduced using fewer bits than if an entire frame was intra coded. The third error resilience tool used to reduce error propagation, is the reference picture selection mode which allows the use of multiple reference pictures. In this mode, an H.264 encoder may use one or two of a number of previously encoded pictures as a reference for inter-picture prediction. This enables the encoder to search for the best 'match 'for the current macroblock from a wider set of pictures than just the previously encoded picture.

The severity of the error caused by discarded packets can be reduced if error conceal-

ment techniques are employed to hide visible distortion. In our simulation environment, we employ the simple and most common approach called *previous frame concealment*, where the corrupted image content is replaced by corresponding pixels from the previous frame. This approach yields good concealment results for sequences with little motion. However, severe distortions may be introduced for image regions containing heavy motion.

4.2 Reed-Solomon Codes

Among the various types of linear block codes, the Reed-Solomon codes are one of the most important for practical applications. They comprise a subset of the BCH codes, which are a class of cyclic codes.

Reed-Solomon codes are nonbinary codes in which the elements of the code words are selected from an alphabet of q symbols, denoted by {0,1,...,q-1}. Usually, $q = 2^m$, so that m information bits are mapped into one of the q symbols. The minimum distance of the nonbinary code is denoted by D_{min}. D_{min} represents the smallest value of the set of *Hamming distances* of the code. The Hamming distance is the number of elements in which two code words differ. A systematic (N, K) block code consists of K information symbols and N - K parity check symbols. The parameter, r, represents the code rate or redundancy. The Reed-Solomon codes considered in this thesis can be described by the parameters:

$$N = q - 1 = 2^m - 1$$

$$K = 1, 2, 3, \cdots, N - 1$$

$$D_{min} = N - K + 1$$

$$r = K/N$$

Such a code is guaranteed to correct up to and including

$$t = \left\lfloor \frac{1}{2}(D_{min} - 1) \right\rfloor$$

$$= \left\lfloor \frac{1}{2}(N-K) \right\rfloor$$

symbol errors.

These codes may be extended or shortened. Extended Reed-Solomon codes were used in this thesis for simulation. For an (*N*, *K*) code with D_{min}, we can construct a singly-extended (*N+1*, *K*) code by appending one additional parity symbol to each code word. This symbol can be added such that the minimum distance by 1 [24].

4.3 Probability of Decoder Error

If a received word is within the minimum Hamming distance of a codeword, the decoder selects that codeword as the most likely to have been sent. If the selected codeword is not the one that was sent, a **decoder error** has occured. In this section, the probability of decoder error is shown to be very small, and approaches zero quickly as t increases. This is significant, since decoder errors are not corrected or detected by the Reed-Solomon decoder.

For a block code over memoryless channels (i.e. channels in which independent noise acts on each use), the probability of decoder error derived in [25] is as follows:

$$P(E) = \sum_{j=D_{min}}^{n} A_j \sum_{k=0}^{\lfloor \frac{d_{min}-1}{2} \rfloor} P_k^j,$$

where

$$P_k^j = \sum_{r=0}^{k} \binom{j}{k-r} \binom{n-j}{r} p^{j-k+r}(1-p)^{k-r} s^{n-j-r}(1-s)^r.$$

The weight distribution $\{A_j\}$ of Reed-Solomon codes is known. The number of codewords of weight i is

$$A_i = \binom{N}{i} (q-1) \sum_{j=0}^{i-D_{min}} (-1)^j \binom{i-1}{j} q^{i-j-D_{min}}, i \geq D_{min},$$

where $q = 2^m$, p is the probability that a *particular* incorrect symbol is received, and s is the probability that a symbol is correctly received. Therefore

$$(1 - s) = (q^m - 1)p.$$

Figure 4.2 illustrates the probability of decoder error for the noise files which have a BER of $2 * 10^{-4}$. Figure 4.3 illustrates the probability of decoder error for the noise files which have a BER of $5 * 10^{-4}$. The simulations assume that Reed-Solomon coding is used with $N = 255$ and 8 bits per symbol.

The curve labelled *Independent Assumption*, corresponds to the probability of decoder error found assuming bit errors are independently distributed. In this case

$$(1 - P_s) = (1 - P_b)^m,$$

where P_s is the probability of symbol error, P_b is the probability of bit error, and m is the number of bits per symbol.

The other two curves, labelled *actual symbol error rate*, corresponds to $P(E)$ calculated using the actual symbol error rate of the noise files.

In Figures 4.2 and 4.3, the independent assumption curve results in a greater probability of decoder error than the actual symbol error rates of the noise files. This is because the errors in the noise file are not completely independent and are bursty in nature. Thus, the use of an interleaver can reduce the packet loss significantly.

4.4 Interleaving

Most of the well-known codes that have been devised are effective when the errors caused by the channel are statistically independent. This is the case for the Additive White Guassian Noise (AWGN) channel. However, many channels exhibit bursty error characteristics. Channels characterized by multipath fading exhibit bursty error characteristics. Signal fading due to time-variant multipath propagation often causes the SNR to fall significantly, resulting in a large number of errors.

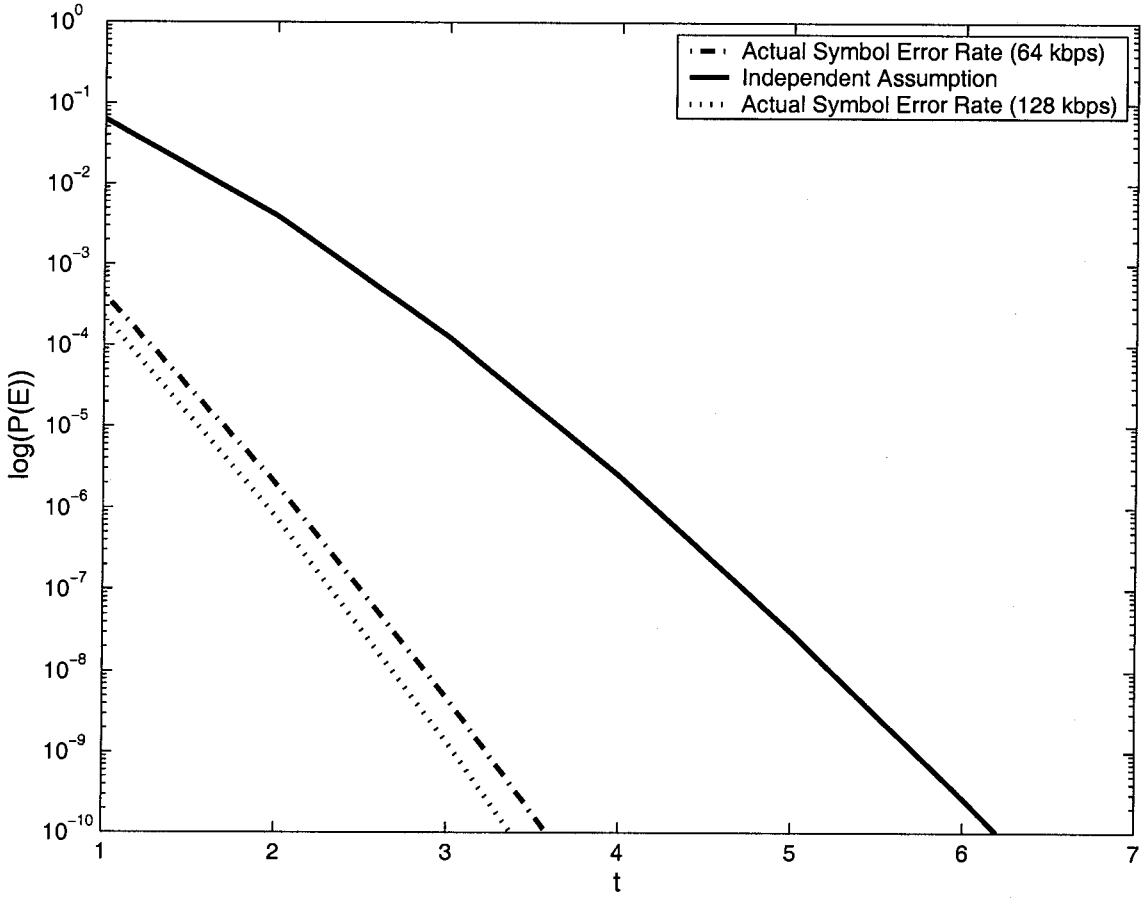

Figure 4.2. *Probability of Decoder Error for a BER of* $2 * 10^{-4}$

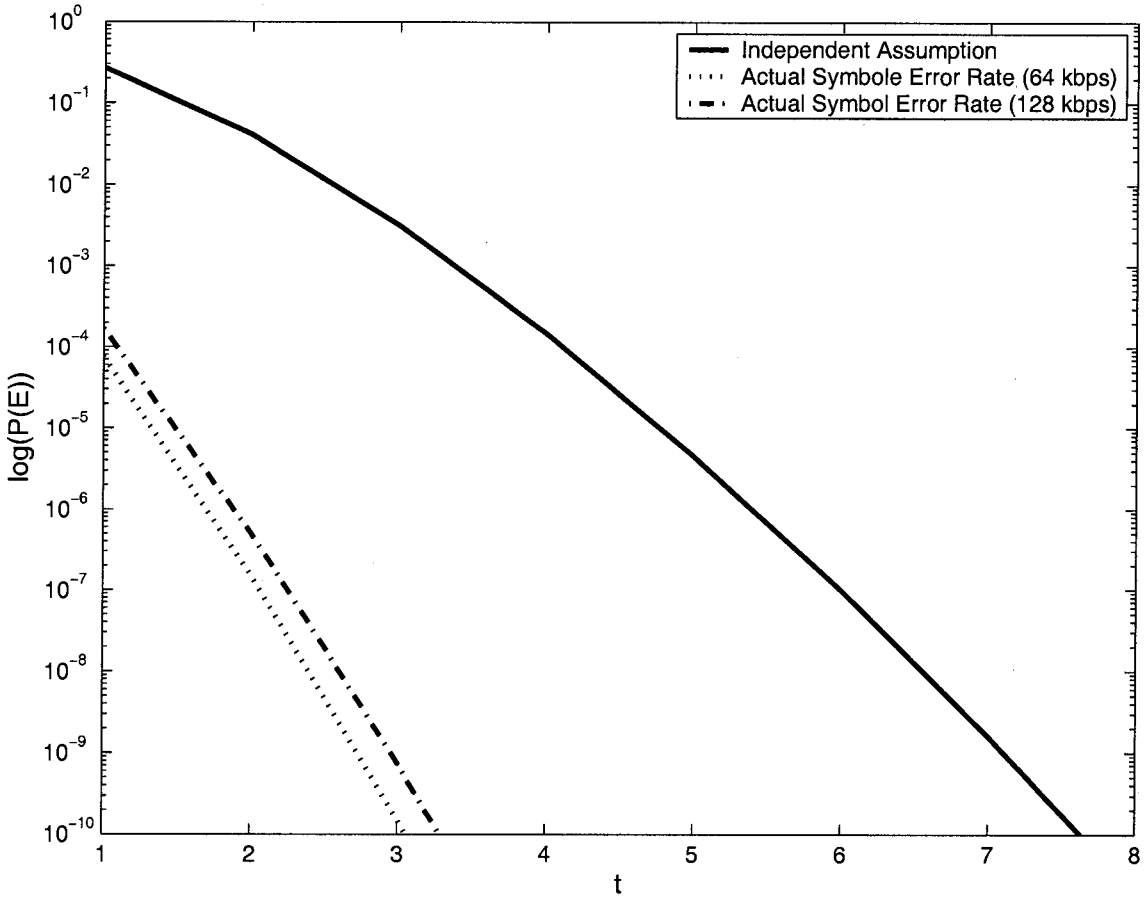

Figure 4.3. *Probability of Decoder Error for a BER of* $5 * 10^{-4}$

A burst of errors of length b is defined as a sequence of b-symbols in which the first and last symbols are in error, with several other errors in between. The *burst error correction capability* of a code is defined as one less than the length of the shortest uncorrectable burst. For an (N, K) code, $b \leq \lfloor \frac{1}{2}(N-K) \rfloor$ [24]. An effective method for dealing with burst error channels is to interleave the coded data in such a way that the bursty channel is transformed into a channel having independent errors. The encoded data can be reordered by an interleaver and transmitted over the channel. At the receiver, a deinterleaver puts the data in the original sequence and passes it to the decoder. As a result of interleaving/deinterleaving, error bursts are spread out in time so that errors within a codeword appear to be independent.

The interleaver structure used in this thesis has a block structure. A block interleaver formats the encoded data in a rectangular array of m rows and n columns. Each row of the array constitutes a codeword of length N. Figure 4.4 shows an example of an interleaver. The symbols are read out column-wise and transmitted over the channel. In the case of Figure 4.4, the transmitted data sequence is given by

$$(A_1, B_1, C_1, D_1, A_2, B_2, C_2, D_2, A_3, B_3, C_3, D_3, A_4, B_4, C_4, D_4).$$

At the receiver, the deinterleaver stores the data in the same rectangular array format, but it is read out row-wise, one codeword at a time. As a result of this reordering of the data during transmission, a burst of errors of length b=ml is broken up into m bursts of length l [24]. In the case of Figure 4.5, after deinterleaving, three consecutive errors are distributed into 3 bursts of length 1.

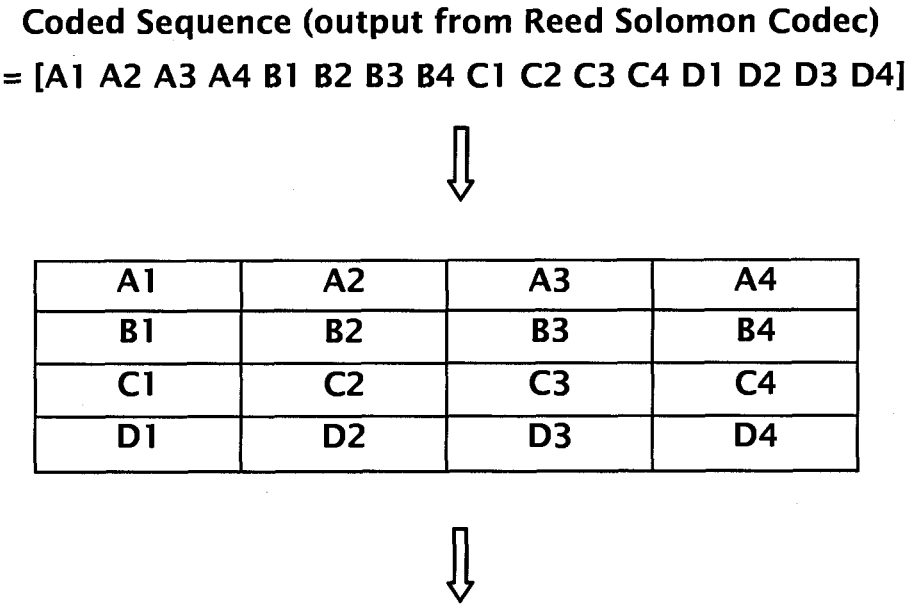

Figure 4.4. *Interleaving Process*

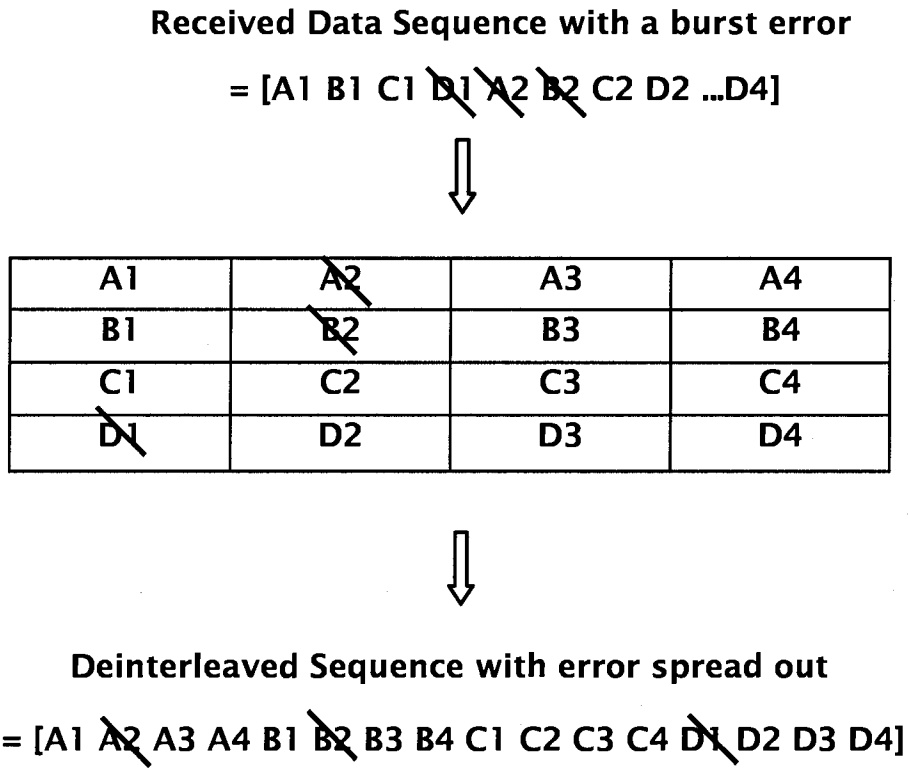

Figure 4.5. *Deinterleaving Process*

4.4.1 Interleaver Delay

The drawback of using an interleaver is the delay. All m interleaved segments of a code-word need to be received before Reed-Solomon decoding can be applied. The delay increases as m increases, however it is also dependent on the channel bit rate. This can be seen in Figures 4.6 and 4.7, which shows the interleaver delay for various block sizes, and data rates. A Reed-Solomon block size of N=256 results in larger delays since more bits are transmitted, however, in Chapter 5 it is shown that they give superior performance than block sizes of N=128 or N=64. The interleaver delay for a channel bit rate of 128 kbps is exactly half of that for a channel bit rate of 64 kbps. For real time systems, interleaving is more practical if higher bit rate (such as 128 kbps) are available.

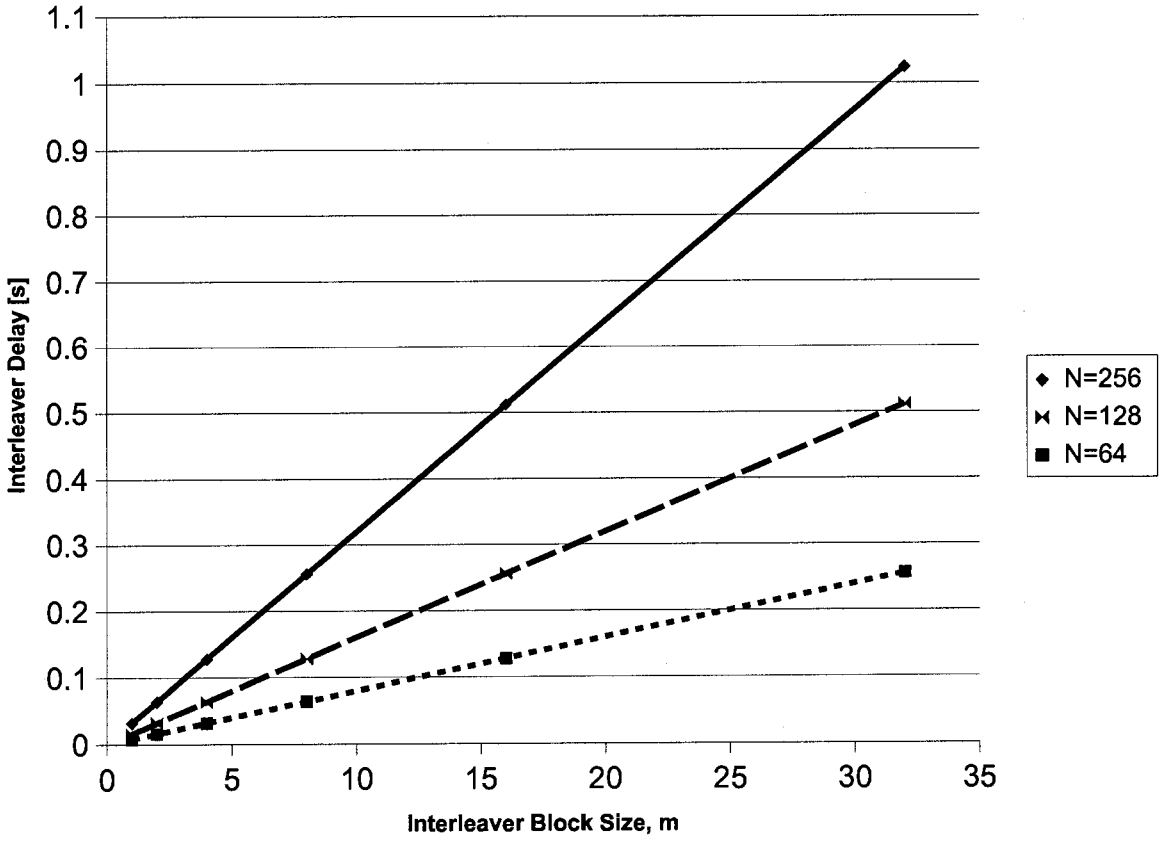

Figure 4.6. *Interleaver Delay at 64Kbps*

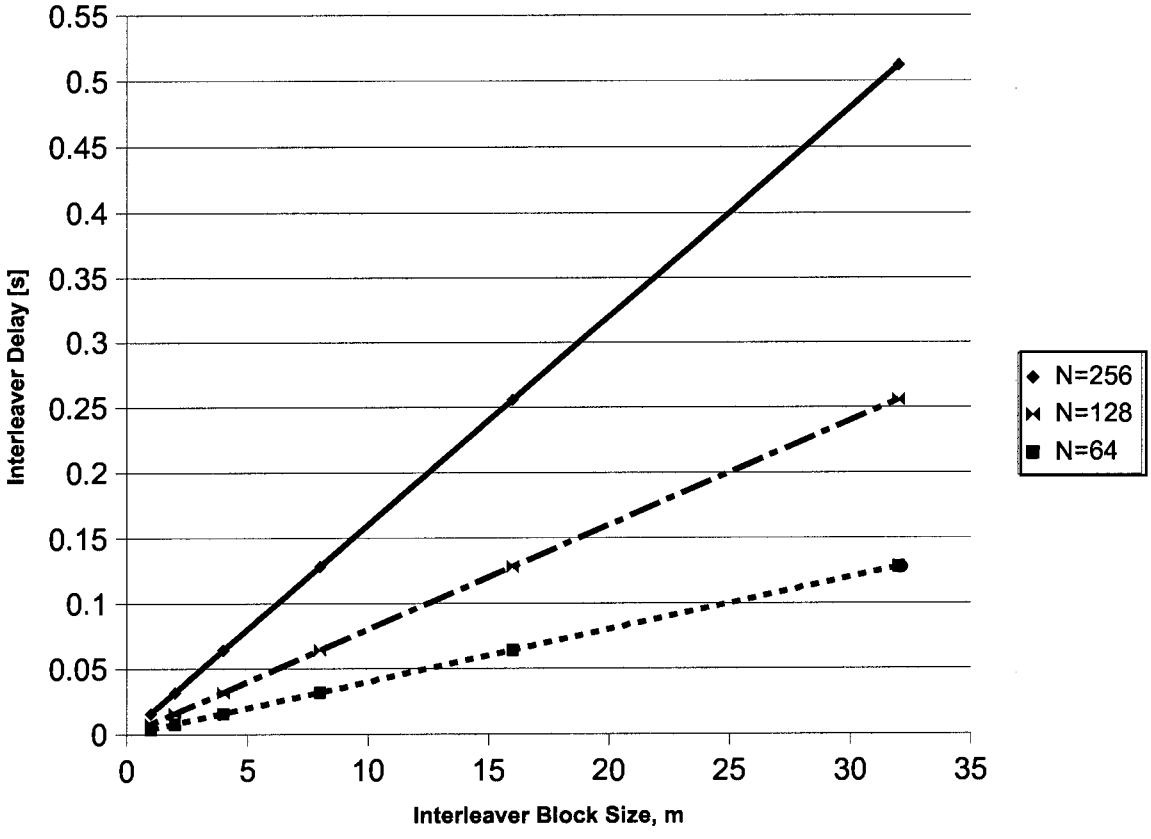

Figure 4.7. *Interleaver Delay at 128Kbps*

Chapter 5

Simulation Results

5.1 Simulation Environment

The simulation environment used to obtain the performance results is as follows. For all simulations performed, the H.264/AVC test model software version JM7.5c [26] was used for encoding and version JM7.3 [27] was used for decoding. The QCIF test sequences Foreman and Carphone were encoded at 10 frames/sec using 400 frames at 30Hz for a bit rate of 64kbps. These sequences were encoded at 15 frames/sec using 400 frames at 30Hz for a bit rate of 128kbps. The encoded frame sequence was I-P-P-P..., and no B-pictures were used. An I-frame occurred every second. Thus, at 64kbps, an I frame was inserted every 10 frames, and for 128kbps bitstreams, an I frame was inserted every 15 frames. Intra macroblock refresh was used to randomly insert intra coded macroblocks in every frame. Context Adaptive Binary Arithmetic Coding (CABAC) entropy coding and Rate-Distortion (RD)-Optimization were used. Two previous frames were used for inter-motion search using a search range of 16 macroblocks. The quantization parameter (QP) of the frames were varied to achieve the desired bit rates.

The mobile channel simulator was called MobileIP [28]. The software simulated the user plane protocol stack for 3G WCDMA networks. It required RTP packetized data as input, broke these packets into frames, and dropped the associated RTP frame if a frame was received in error.

Results were obtained for channel bit rates of R_c= 64 kbps and R_c= 128 kbps. An

Table 5.1. *Noise File Parameters*

File Name	Bit Rate	Bit Error Rate	Mobile Speed
Noise File 3	64 kbps	0.00051	3 kph
Noise File 4	64 kbps	0.00017	50 kph
Noise File 5	128 kbps	0.0005	3 kph
Noise File 6	128 kbps	0.0002	50 kph

(N, K) Reed-Solomon code with a block size of N symbols with 8 bits per symbol was used. Three values of N, 256, 128, and 64, equal to the PDU size, were employed.

For simulating radio channel conditions bit-error patterns are used, that were captured in different real or emulated mobile radio channels. The bit-error patterns are captured above the physical layer and below the RLC/RLP layer, such that in practice they act as the physical layer simulation. It is assumed, that the statistical parameters (e.g. average residual BER and burstiness) of the bit-error pattern are similar for the whole transmission [28]. The packet loss rate was determined by averaging the loss over 100 transmissions. The 100 start positions for the error patterns were randomly chosen, but for comparison purposes, the same start positions were applied.

5.2 Video Compression at Encoder

In order to maintain a constant channel data rate R_c, the available data rate for the source encoder was reduced according to the relation $R_e = rR_c$. This way, there was no change in the data rate due to Reed-Solomon coding. Thus, there was a tradeoff between video quality and error protection.

Generally, for acceptable video quality, the encoded PSNR should be greater than 30 dB [29]. An encoded video PSNR of 10-20 dB results in poor quality, while 30-40 dB is acceptable quality and 50-60 dB is very good quality (for broadcast type of applications) [29]. I chose to keep the encoded luma PSNR above approximately 32 dB. This value is a

little above the level for acceptable video quality of 30 dB, to give a little room for video degradation as a result of channel noise.

The first frames of the Foreman sequence, one with a luma PSNR of 34.06 dB, and the other with a PSNR of 31.74 dB, are illustrated in Figures 5.1 and 5.2, respectively. There is a slight degradation in video quality as the PSNR is reduced. However, the degradation is hardly noticeable making a reduction in quality for Reed-Solomon coding overhead a great choice. The sequence shown in Figure 5.2 was the lowest quality video sequence used in this thesis with Reed-Solomon coding for transmission.

For a given bit rate, the compression ratio achieved by the H.264/AVC video encoder is not dependent on the video sequence. As expected, the compression ratio increases with the reduction in video bit rate. This is illustrated in Figures 5.3 and 5.4. In this thesis, the compression achieved by the video encoder as well as the reduction in frame rate resulted in a compression ratio which ranged from 100 to 220. That is, the original video sequence of 14.5 Mbytes, was compressed to file sizes of 145 - 66 Kbytes for transmission. The compression ratio is a function of the video encoder used. H.264/AVC can achieve compression ratios much higher than previous video coding standards, such as H.263 and MPEG-2. Due to these higher compression ratios, fewer bits are used to represent the same video sequence. Loss of these encoded video bits will typically result in the loss of valuable information making error recovery techniques critical for H.264 video transmission.

For the bit rates used, the reduction in video quality (PSNR) is plotted in Figures 5.5 and 5.6. For a given bit rate, the Foreman sequence has a luma PSNR of approximately 1.5 dB less than the Carphone sequence. The plots are also not completely straight lines, especially for the Foreman sequence. This is because more points were taken for the Foreman sequence than the Carphone sequence, and changing the quantization parameters in the video codec does not result in a completely linear change in the PSNR. Two quantization parameters that can be changed in the video codec were the quantization parameter of the I-frames, and the quantization parameter of the P-frames. Since there was 1 I-frame for every 9 P-frames at 64 kbps, and 1 I-frame for every 14 P-frames at 128 kpbs, changing

the I quantization parameter would change the PSNR less than changing the quantization parameter for the P-frames.

5.3 Frame Length

Selecting a suitable frame length for the PDU frame size is dependent on the Reed-Solomon block length, N, for RTP packetized video. Figure 5.7 shows that the frame loss decreases as the frame length decreases. Thus, if the received sequence was dependent only on frame loss as opposed to packet loss, decreasing the frame size would be advantageous, keeping in mind the associated overhead. However, since a frame loss results in the loss of an entire RTP packet, the frame size does not effect the packet loss rate, as can be seen in Figure 5.8. The large jumps in the plot are due to a single packet loss difference. Since few RTP packets are transmitted for an entire video sequence, typically 100-200, a single RTP packet loss will result in a significant jump in the error rate. The chosen frame lengths were 256, 128, and 64 since these are convenient values for Reed-Solomon coding.

5.4 Uninterleaved Results

The results shown in this section are for uninterleaved Reed-Solomon coding. The frame loss and packet loss are shown for 100 transmission runs over 4 different noise files. The noise file characteristics used in the simulations are listed in Table 5.1. For each noise file, the Foreman sequence was encoded with three values of N: 256, 128, and 64. The Carphone sequence was encoded at $N = 256$ to verify the accuracy of the results obtained with the Foreman sequence.

As can be seen by Figures 5.9 to 5.16, Reed Solomon coding steadily reduces the frame/packet loss rate as the amount of redundancy increases. In Figures 5.9 and 5.11, Reed-Solomon coding affects the reduction in loss in the same way. At Reed-Solomon code rates close to 1, the frame loss rate is spread out. As the coding increases and the

coding rate decreases, the plots tend to merge closer together. This is due to the nature of the errors in the error files. The addition of a little Reed-Solomon redundancy results in the saving of a number of packets which are likely due to random errors. However, the bursty errored packets cannot be saved without a significant amount of coding, which is why the plots do not reach zero. Also, the frame loss rate is much lower for noise file 4 than it is in noise file 3. This is because the higher mobile speed (50kph) results in a more ergodic channel than the case of the walking user (3kph). Therefore [5], the error rates are usually higher for slowly moving users than for fast moving users. Figures 10 and 12 depict the corresponding RTP-packet loss rates which are dependent on the corresponding frame loss rate plots 9 and 11, respectively. The RTP-packet loss rates are not spread out as much for Reed-Solomon code rates close to 1 since the Reed-Solomon code block length, N, is set to the frame length, not to the RTP packet length. The RTP packets are broken up into fixed sized frames, and a single frame size results in the entire associated packet loss. Thus, frame size does not significantly alter packet loss rate for a given noise file.

Figures 5.13 and 5.15 are similar to Figures 5.9 and 5.11, however, they were obtained over noise files 5 and 6, respectively. These noise files are for channel bit rates of 128 kbps as opposed to 64 kbps. The frame loss rates and packet loss rates are very similar to those obtained at lower frame rates. Thus, channel bit rate does not significantly alter the frame loss rate. The packet loss rates obtained in Figures 5.14 and 5.16 are also very similar to those in Figures 5.10 and 5.12. This is again because the only difference here is the channel bit rates and the random start positions in the noise files resulting in only slightly different results.

Values of code rate r less than 0.6 were not evaluated since the encoded PSNR of the video stream becomes too low. Even if packet loss could be reduced further at rates lower than 0.6, the video's PSNR would be too low resulting in a low quality video stream.

The best Reed-Solomon block length is $N = 256$, however, the two other block lengths used result in similar improvements. However, a block length of $N = 256$ always performs a little better. A Reed-Solomon code in the range of 0.625-0.65 is the best code rate to use

for real-time error protection. This is because at this code rate, the packet loss rate can be reduced the greatest without significantly reducing the video's quality. For streaming and messaging applications with less strict delay constraints, a higher Reed-Solomon code rate, which will only degrade the video quality by approximately 1dB is suggested. Since delay can be tolerated for these appliations, the video quality should be kept as high as possible, and other error protection techniques should be applied, such as interleaving.

5.5 Interleaved Results

Interleaving can significantly reduce the packet loss rate of H.264/AVC video due to the bursty nature of W-CDMA network channel conditions. Figures 5.17 and 5.18 illustrate the packet loss rate for a total interleaver delay of 64ms at a channel bit rate of 128 kbps. This interleaver delay may be tolerable by some real-time conversational applications to further reduce packet loss from the uninterleaved case. For an interleaver delay of 64ms, the Reed-Solomon code block length of $N = 256$ performs best. For an interleaver delay of 256 ms, at a bit rate of 64 kbps, the packet loss rate falls down to 0, as shown in Figures 5.19 and 5.20. Again, a block length of $N = 256$ performs best for a broad range of Reed-Solomon code rates. However, as the Reed-Solomon coding is increased, all three block lengths give similar performance.

The plots obtained in Figures 5.21 to 5.24 illustrate the packet loss rate as a function of the number of interleaved blocks, m. As the number of interleaved blocks increases, the packet loss rate almost always drops to 0, except in the case when the block length $N = 64$ and the channel bit rate is 128 kbps. This is because, the channel errors at this bit rate are more bursty, and a larger Reed-Solomon block length, such as $N = 128$ or $N = 256$ is necessary.

It is important to note that in some cases a small number of interleaved blocks ($m = 2$ or $m = 4$) or a small block length ($N = 64$) can in fact increase the packet loss from the case where no interleaving is used. This is because interleaving will spread the errors to

more packets without the error correction capability of saving them.

5.6 PSNR of Decoded Video

The luminance (luma) PSNR of the decoded video sequence is plotted as a function of frame number in Figures 5.25 and 5.26. Six video sequences were transmitted over Noise File 3 at 64kbps with a frame rate of 10 fps, each of which had a different random starting position. In both figures, the original sequence's PSNR is plotted without any noise or Reed-Solomon coding. The original video sequence was then transmitted 6 times through the noise channel and the decoded video PSNR was obtained both in the case without Reed-Solomon coding (shown in Figure 5.25) and in the case with Reed-Solomon coding (shown in Figure 5.26).

The PSNR is shown to give an idea of how Reed-Solomon coding improves the video quality, however, the plots shown here are not the exact values of the luma PSNR in the cases where noise was added. The PSNR shown for the original video sequence, however, is the exact PSNR value. The PSNR is not a good indication of received quality in the figures shown, as the decoder loses synchronization with the encoded bitstream when calculating PSNR. This occurs since packet loss often results in the discarding of a complete video frame. Thus, in reality, the received video quality is much higher than appears from these PSNR plots. Further, noisy sequence 6 could not be decoded at all and is not shown in Figure 5.25. This is likely due to a packet loss in the header information of the sequence. However, when Reed Solomon coding was added, this sequence was recovered and could be decoded as shown in Figure 5.26.

Another factor to keep in mind when viewing the PSNR plots shown is that there is no indication of lost frames as a result of packet loss and video decoding. In simulations without Reed-Solomon coding, approximately 5-10 of each decoded video stream's frames were lost. However, when Reed-Solomon coding was used, much fewer frames were lost: approximately 1-2 frames of 2 out of 6 video sequences were lost.

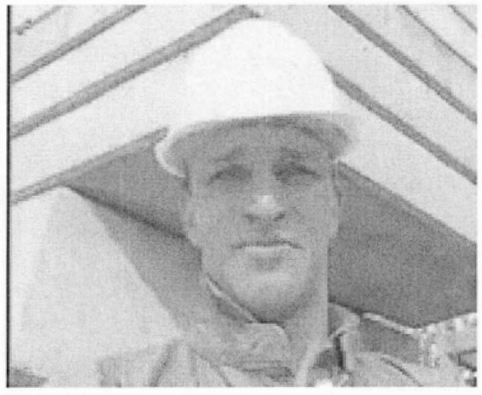

Figure 5.1. *Foreman Frame with Encoded PSNR of 34.06 dB [1]*

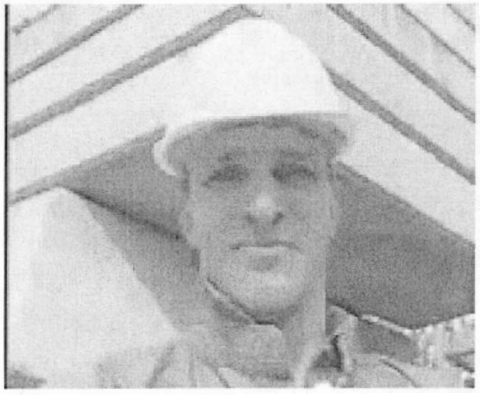

Figure 5.2. *Foreman Frame with Encoded PSNR of 31.74 dB [1]*

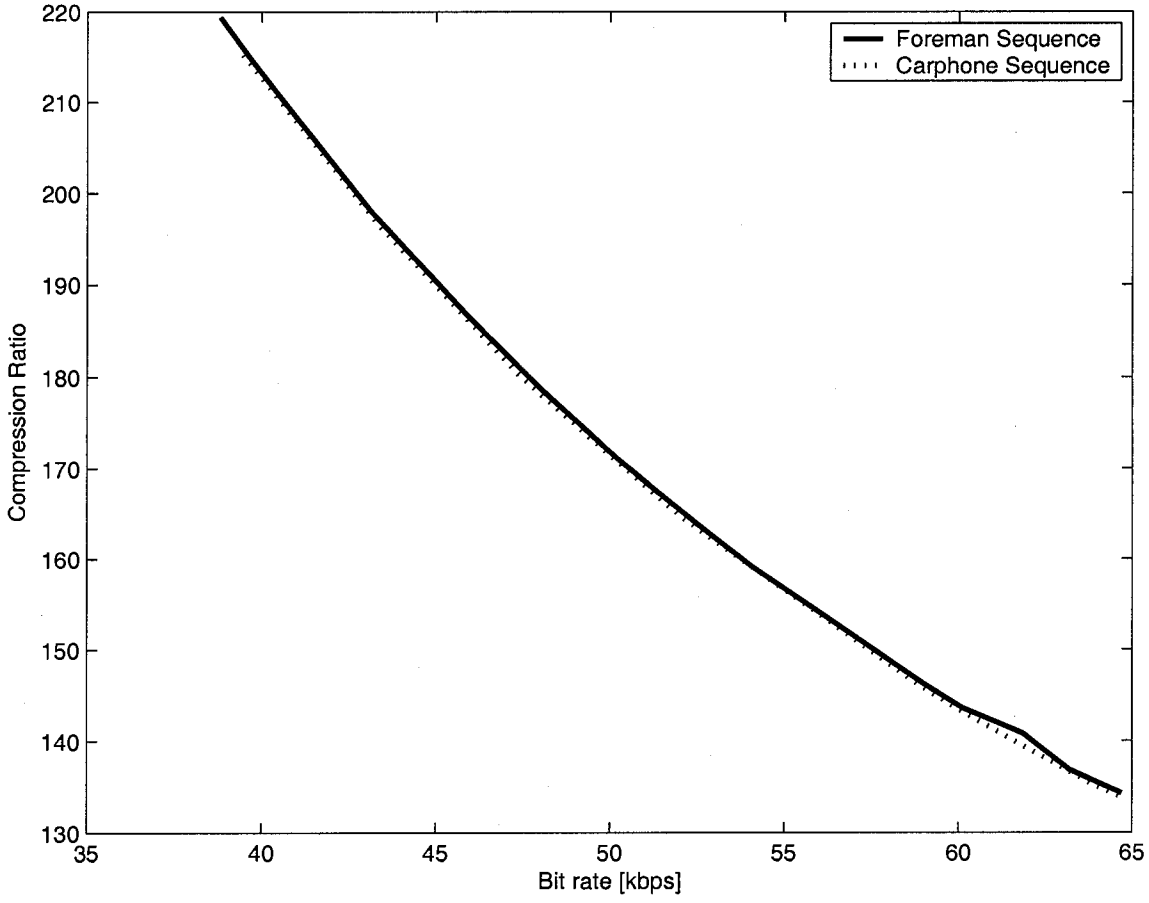

Figure 5.3. *Compression Ratio of Foreman and Carphone Sequences Used over 64kbps Noise Files*

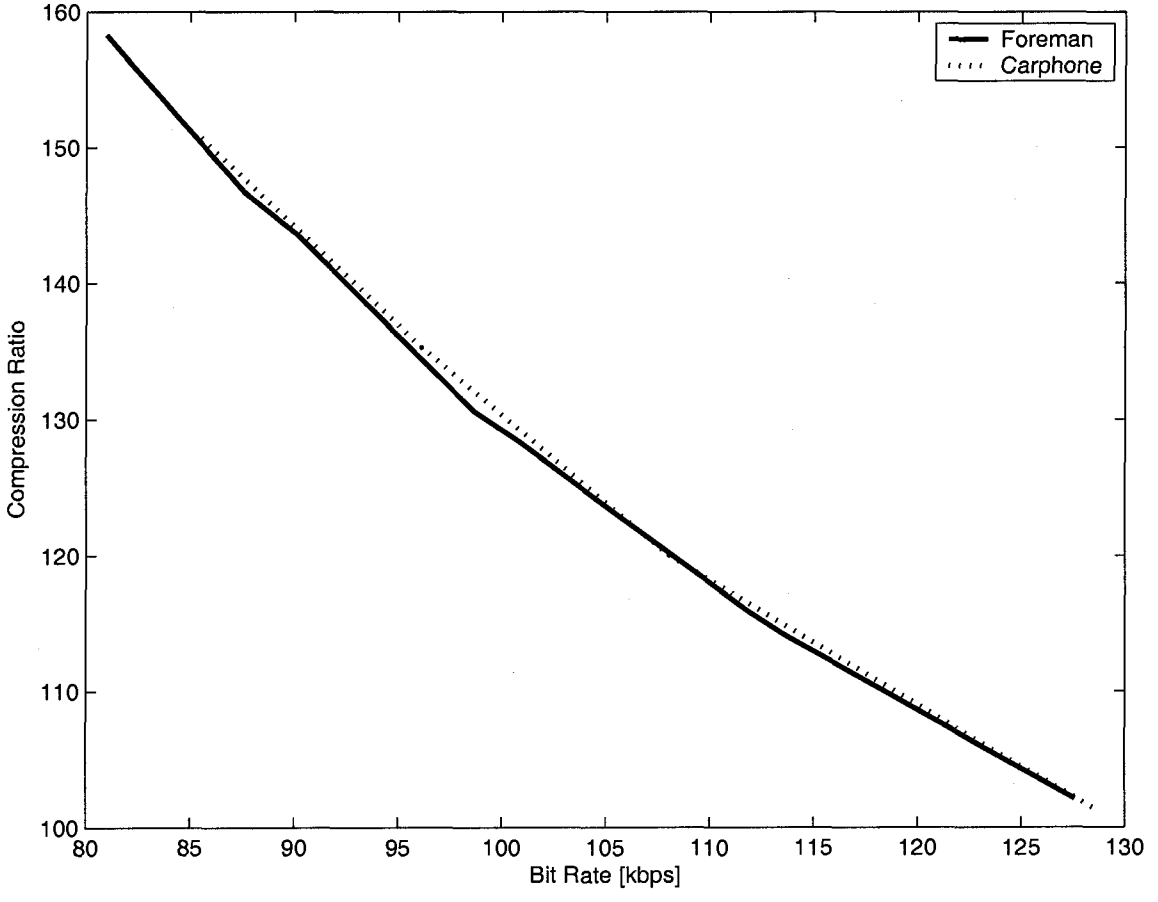

Figure 5.4. *Compression Ratio of Foreman and Carphone Sequences Used over 128kbps Noise Files*

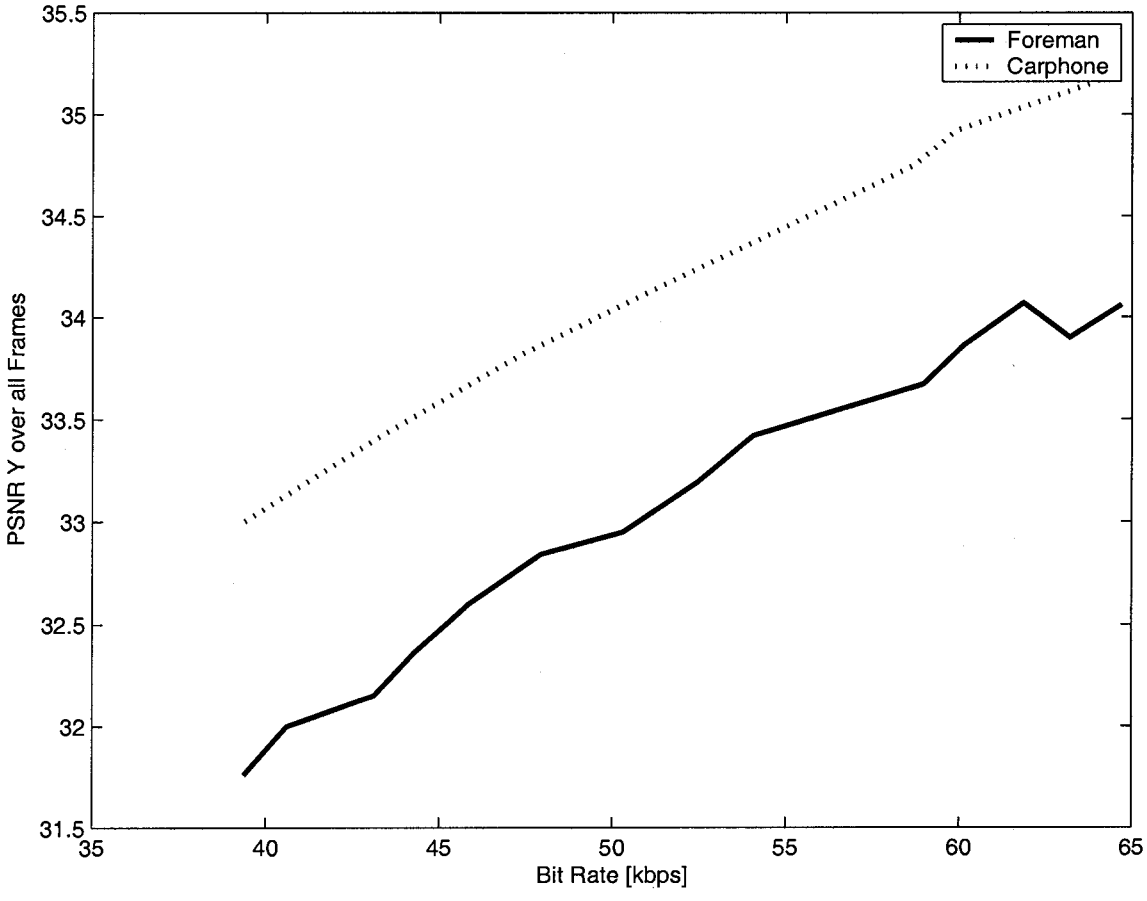

Figure 5.5. *Luma PSNR of Encoded Foreman and Carphone Sequences*

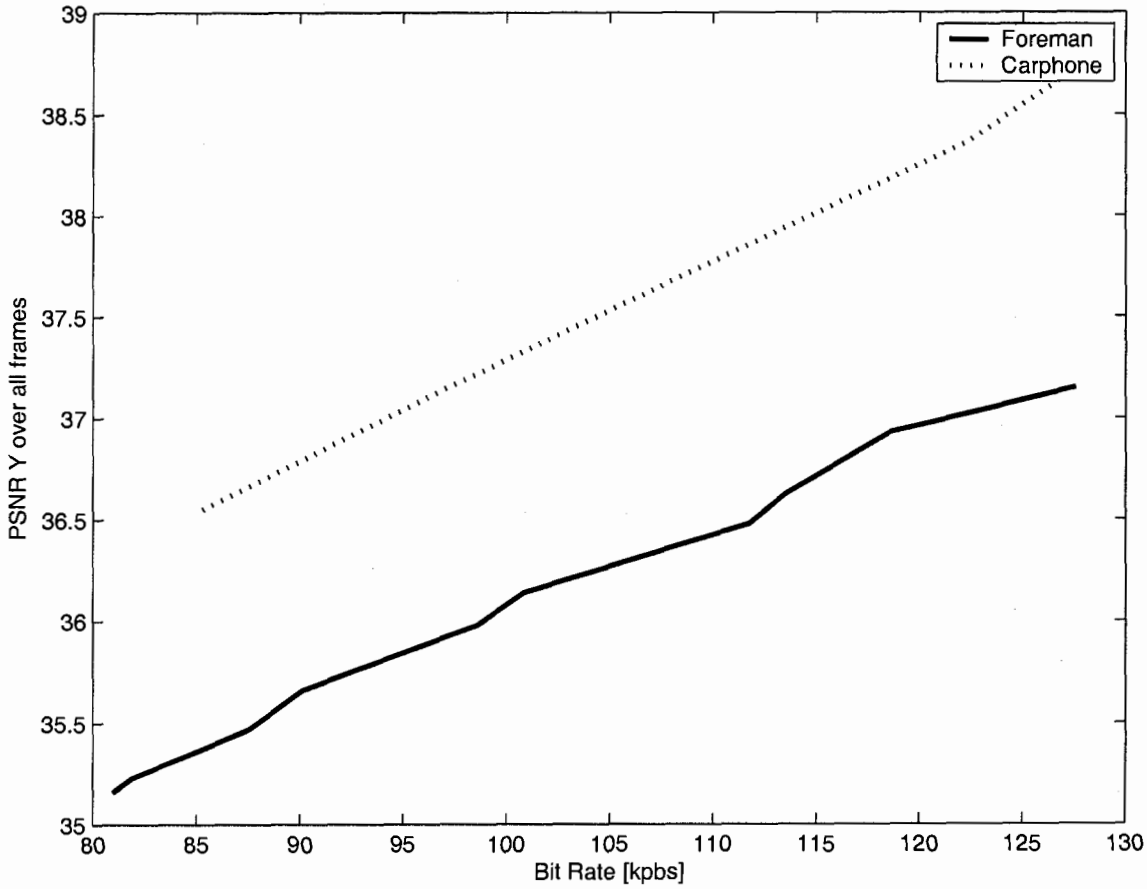

Figure 5.6. *Luma PSNR of Encoded Foreman and Carphone Sequences*

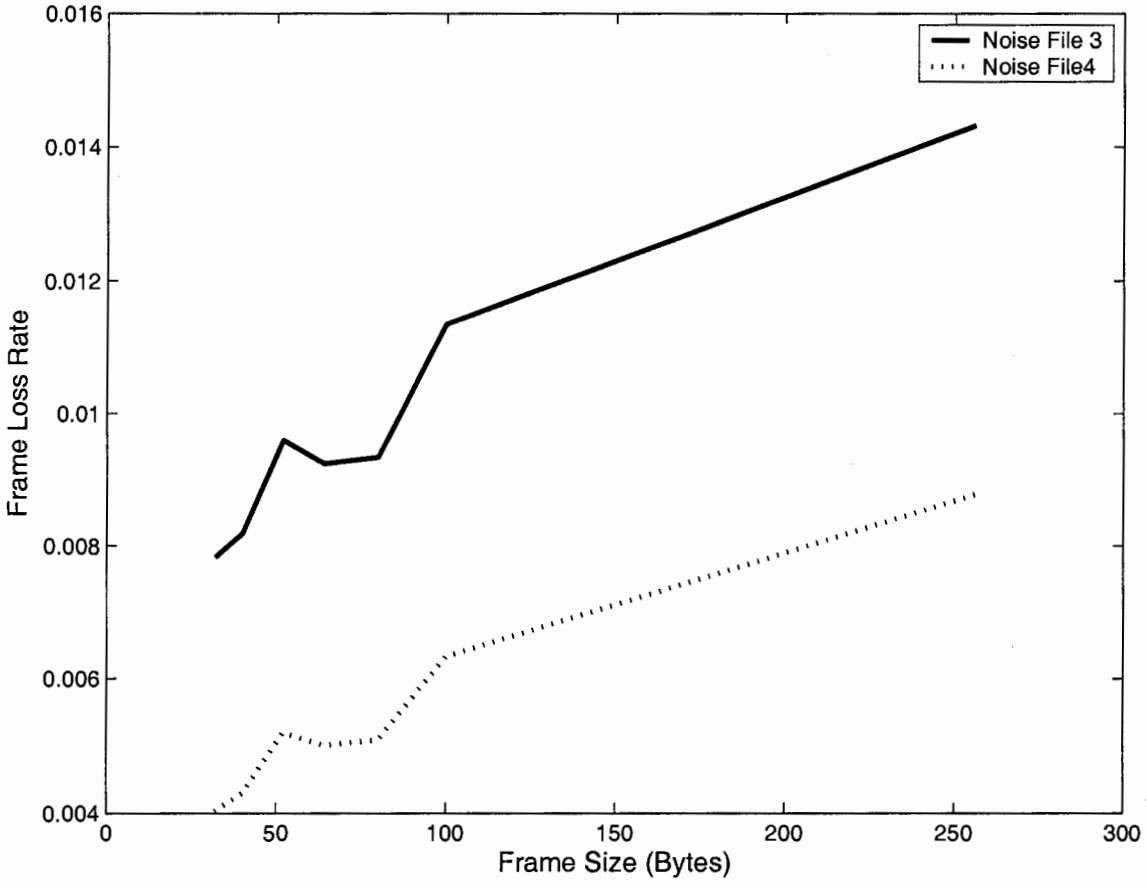

Figure 5.7. *Frame Loss versus Frame Length*

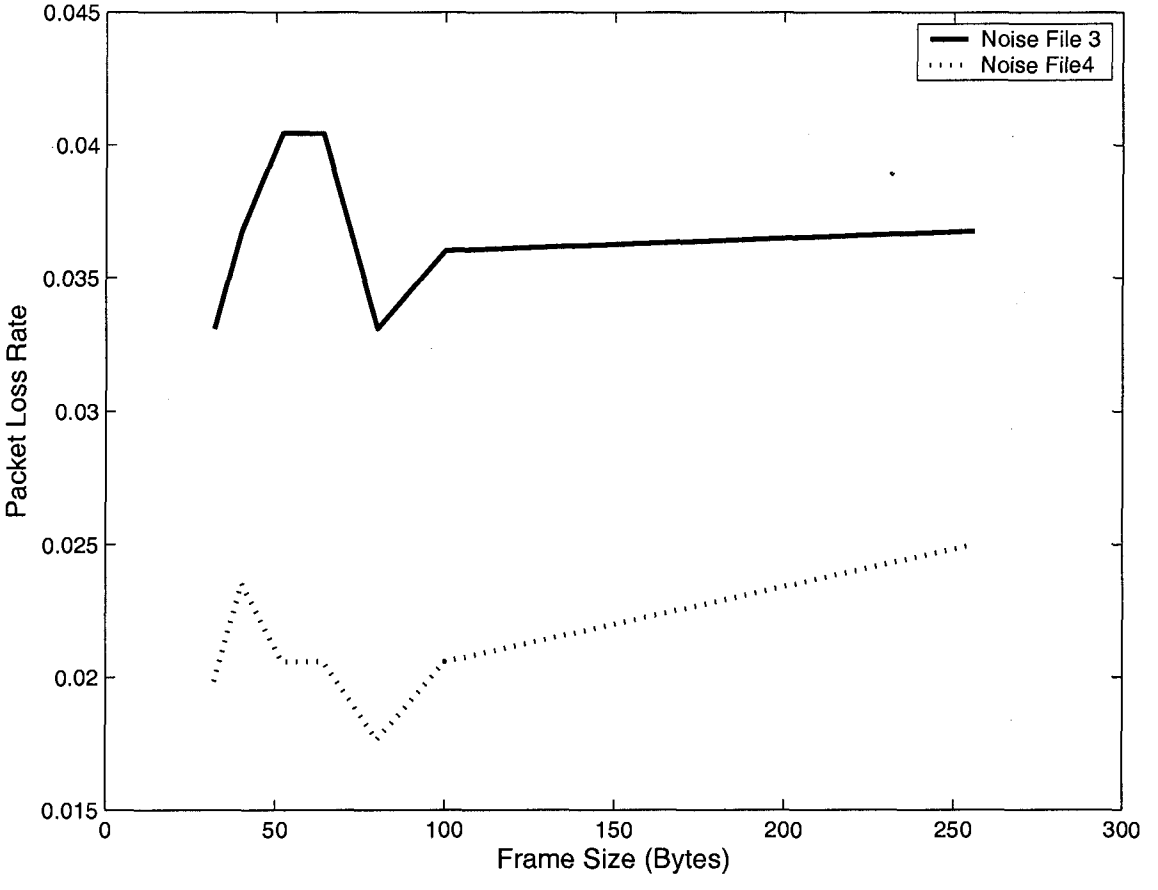

Figure 5.8. *Packet Loss versus Frame Length*

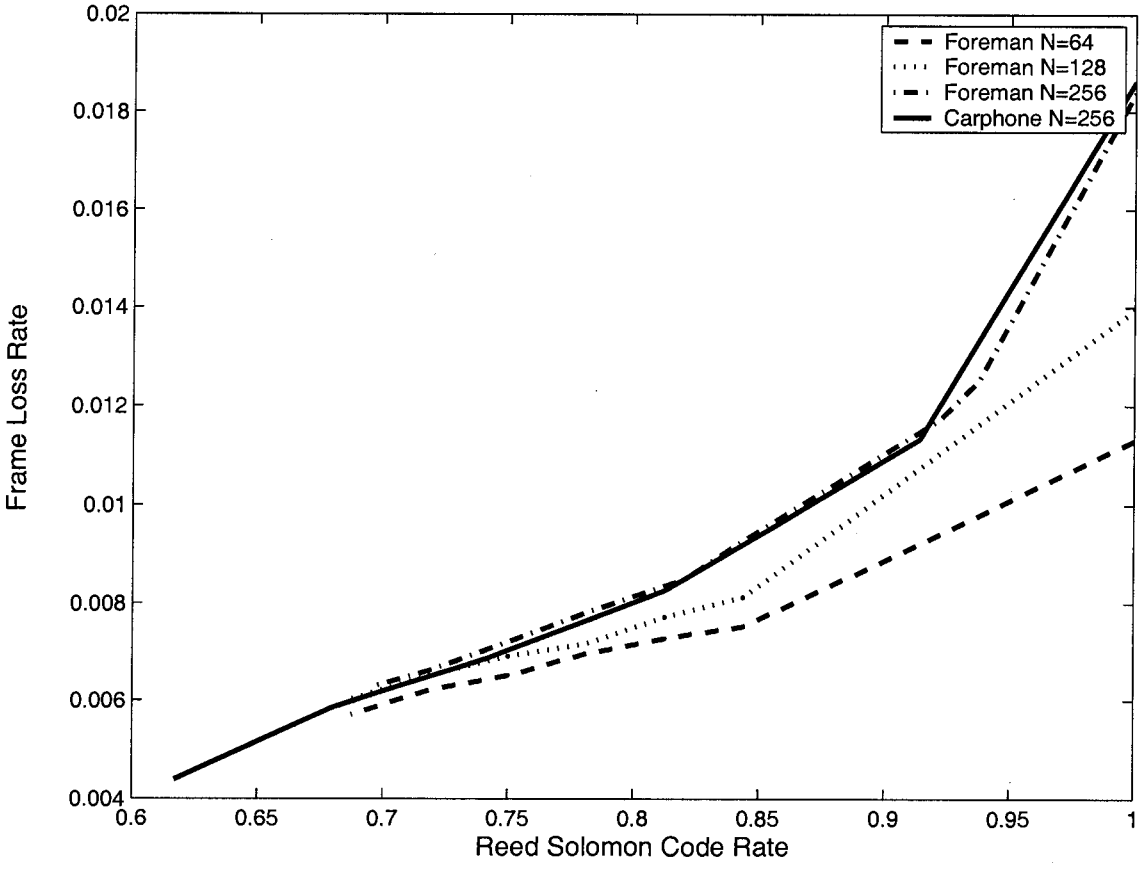

Figure 5.9. *Uninterleaved Average Frame Loss with Noise File 3*

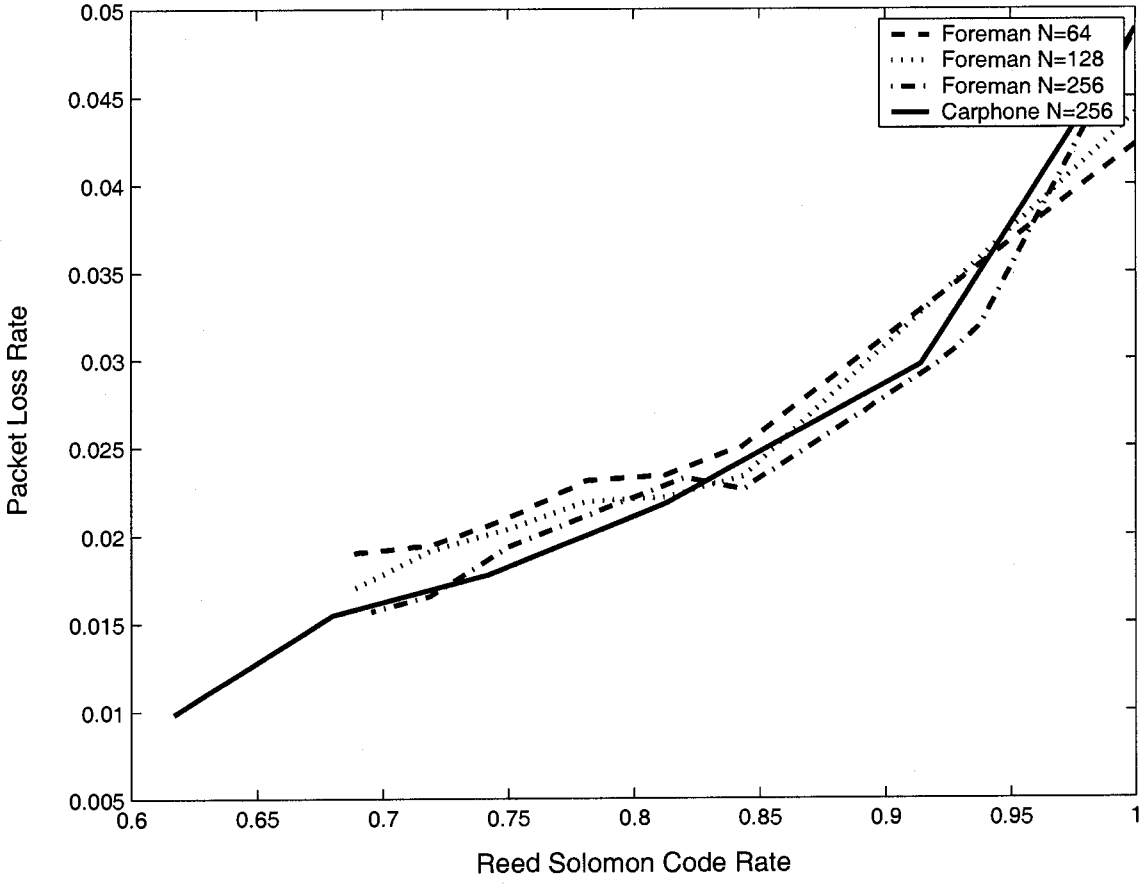

Figure 5.10. *Uninterleaved Average Packet Loss with Noise File 3*

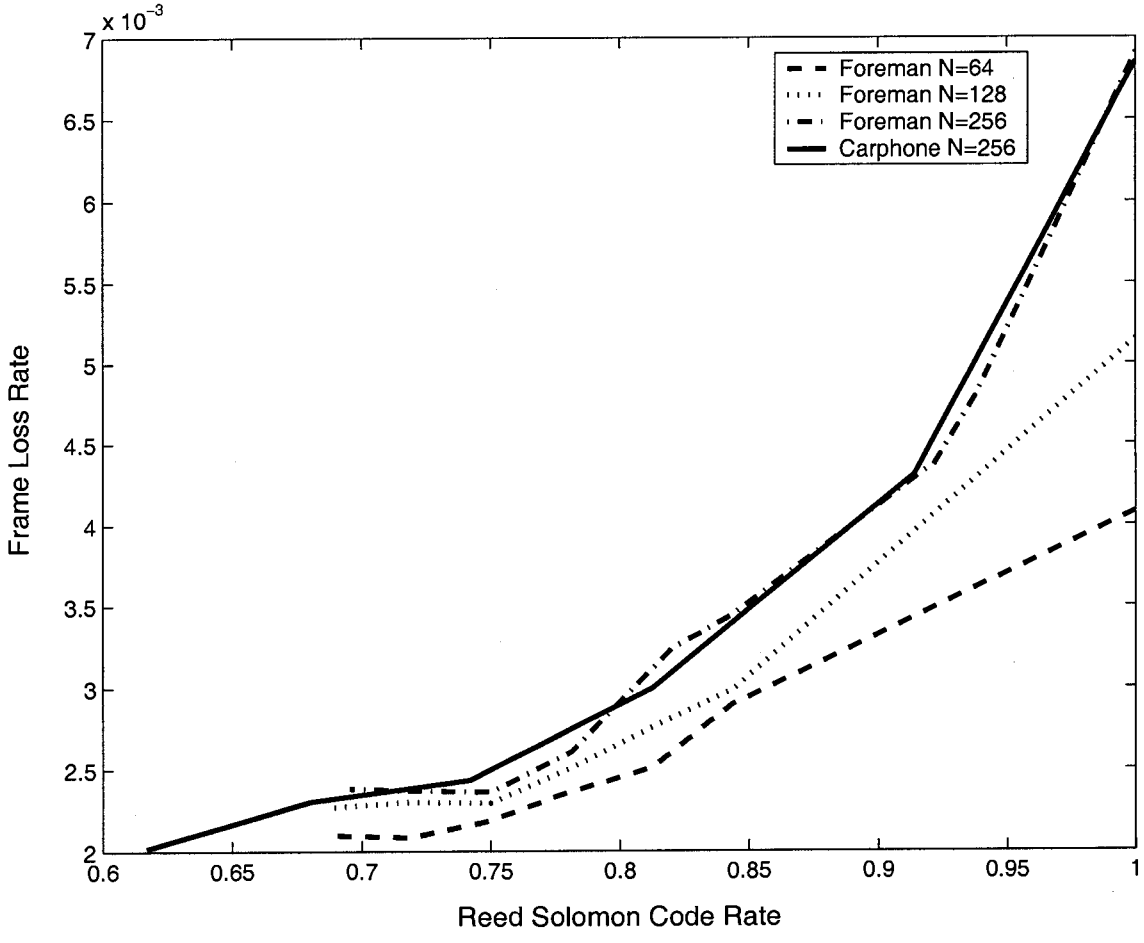

Figure 5.11. *Uninterleaved Average Frame Loss with Noise File 4*

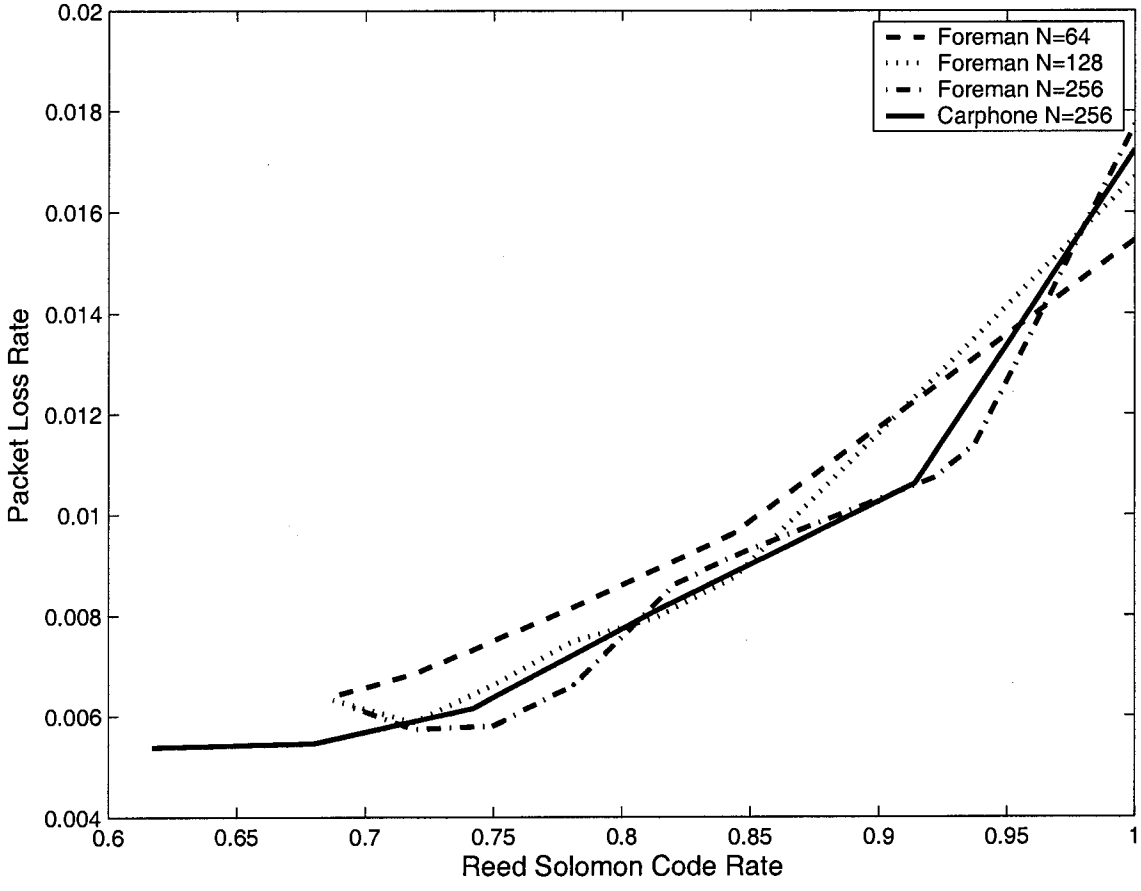

Figure 5.12. *Uninterleaved Average Packet Loss with Noise File 4*

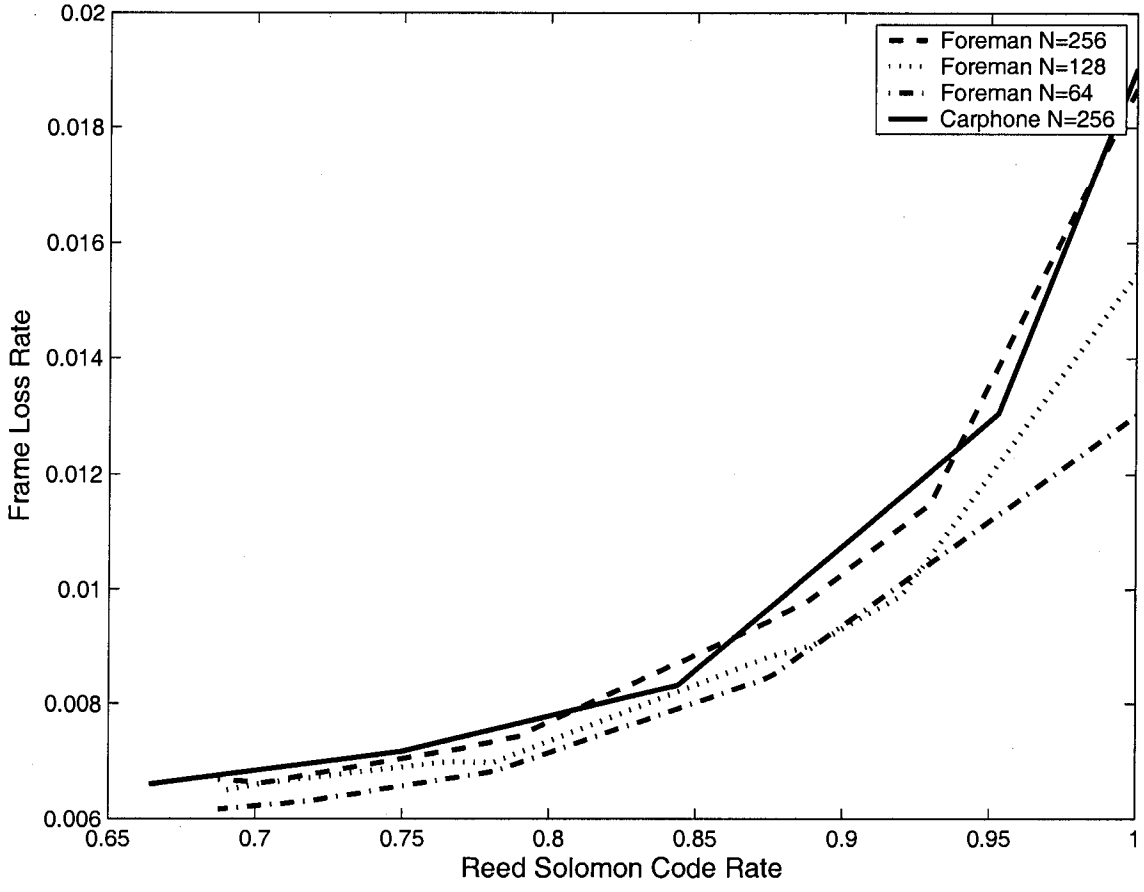

Figure 5.13. *Uninterleaved Average Frame Loss with Noise File 5*

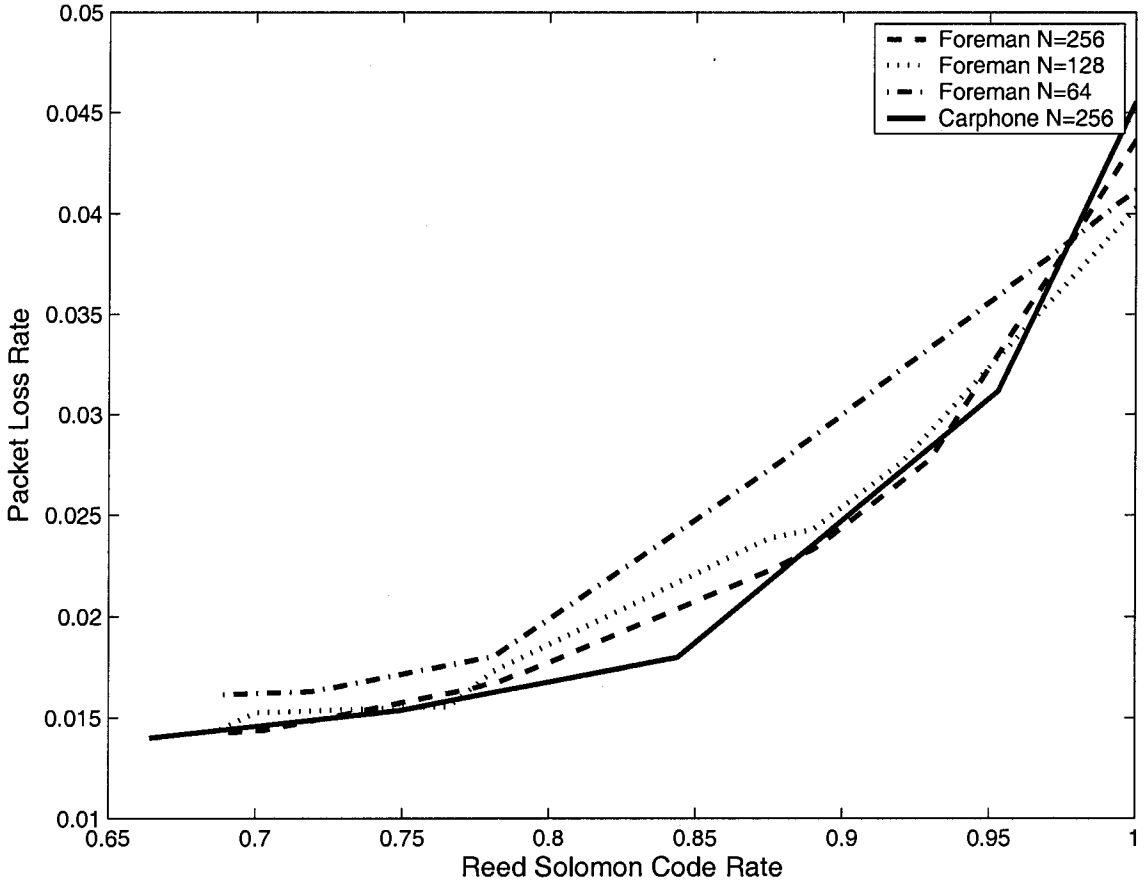

Figure 5.14. *Uninterleaved Average Packet Loss with Noise File 5*

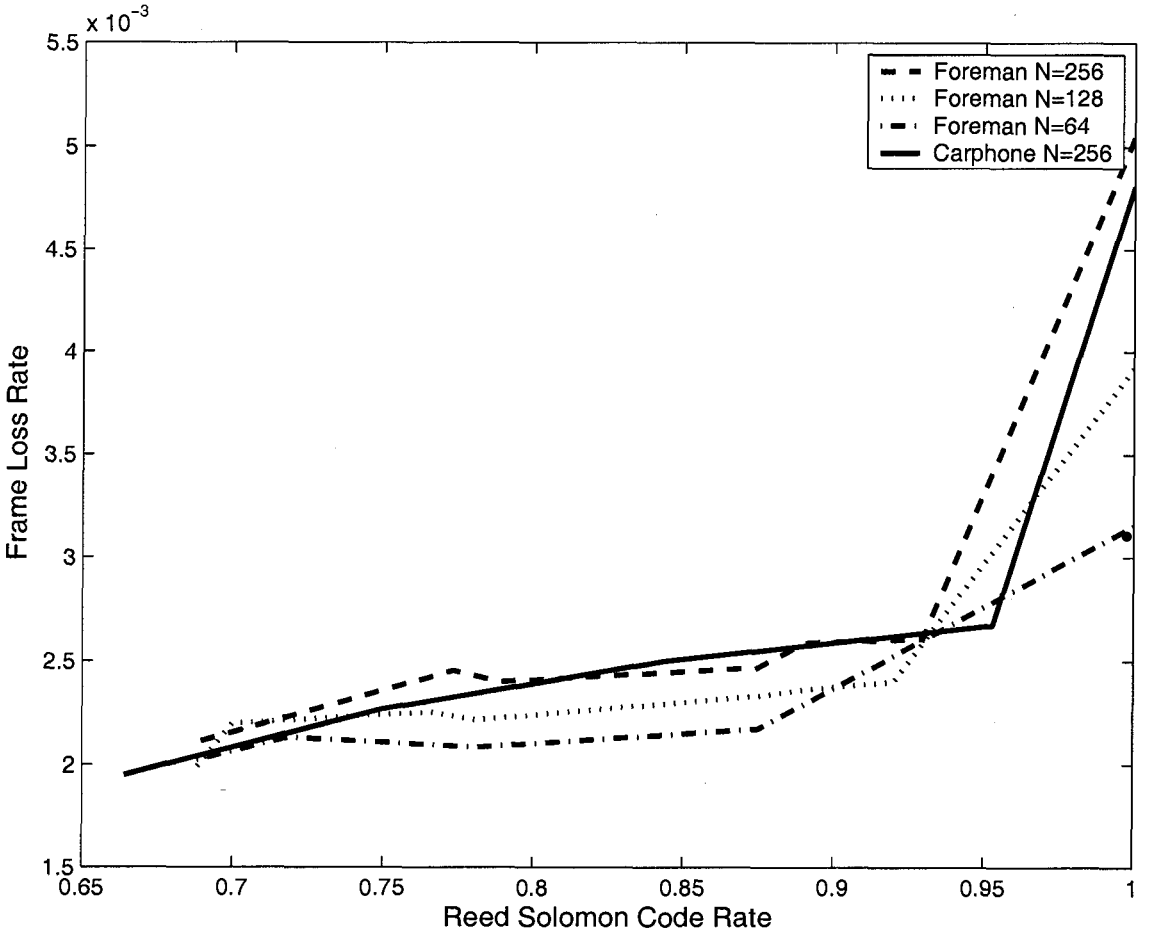

Figure 5.15. *Uninterleaved Average Frame Loss with Noise File 6*

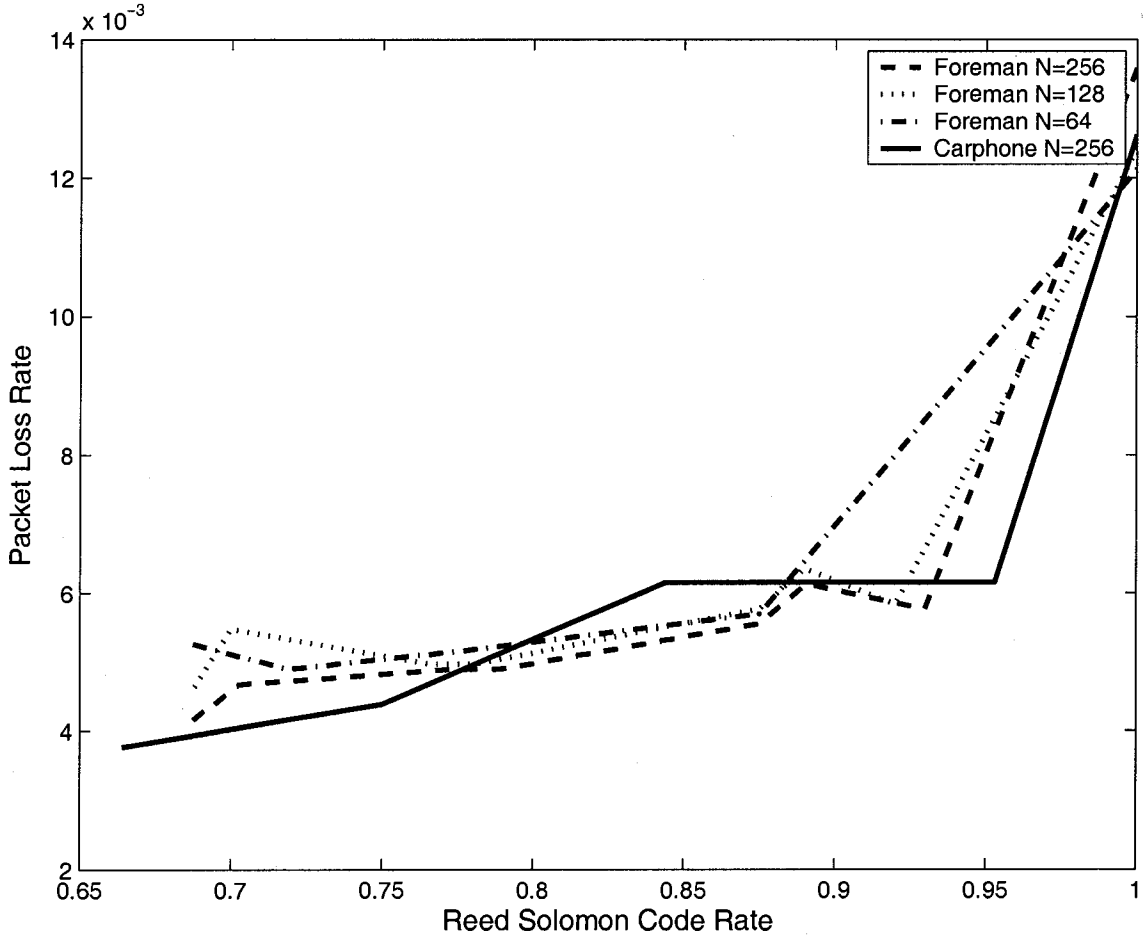

Figure 5.16. *Uninterleaved Average Packet Loss with Noise File 6*

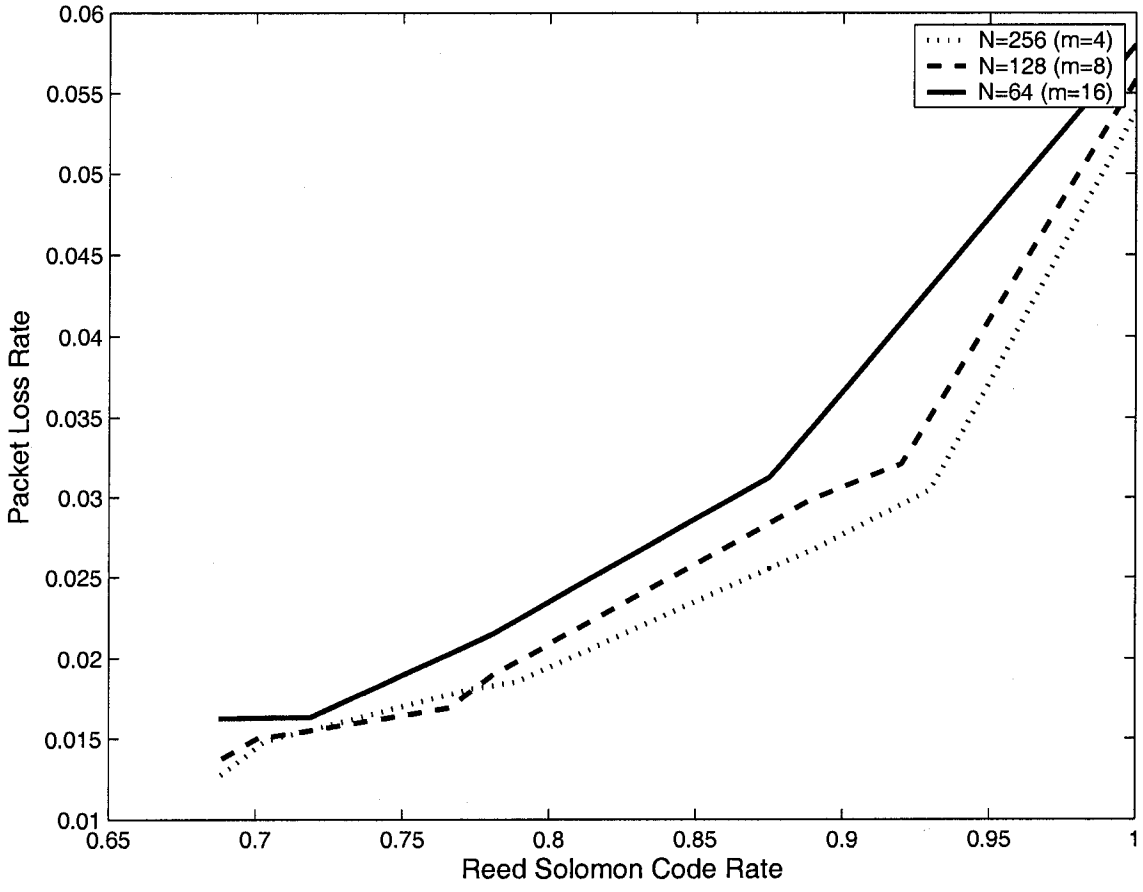

Figure 5.17. *Packet Loss with Interleaver Delay of 64ms with Noise File 5*

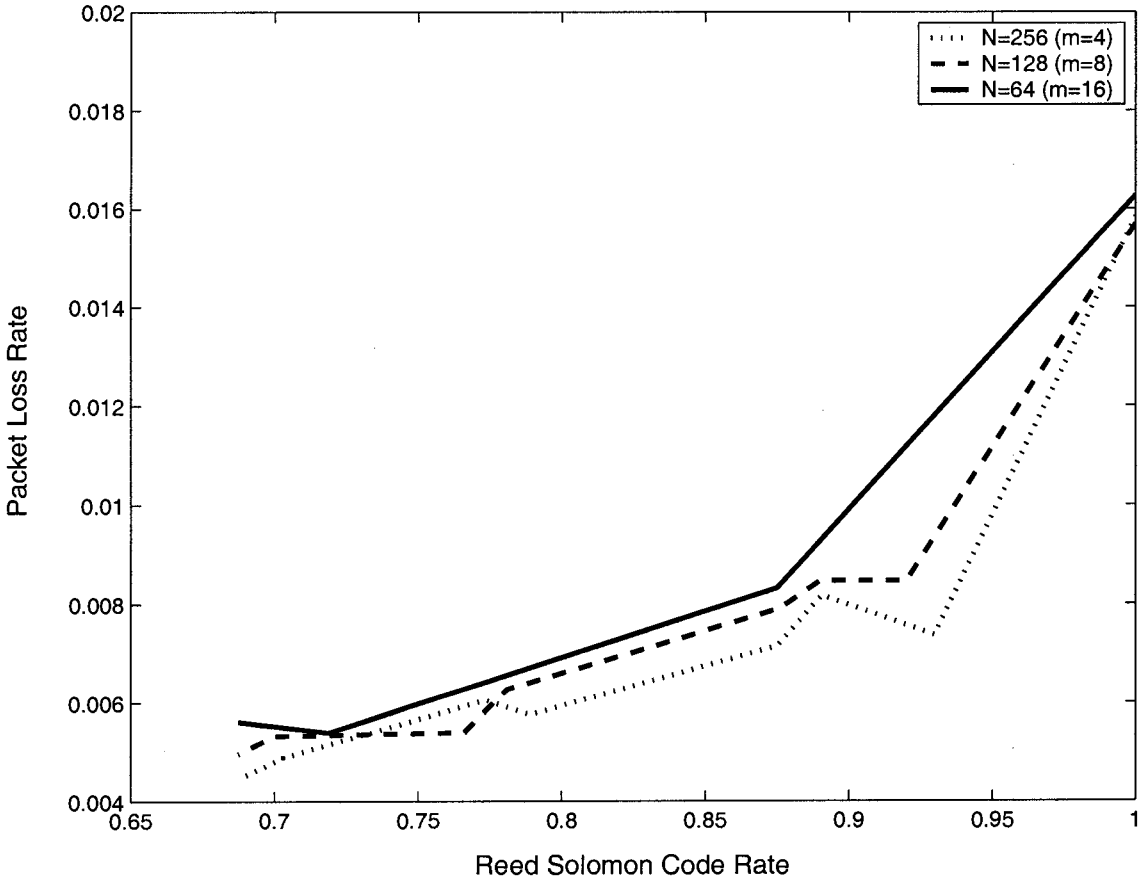

Figure 5.18. *Packet Loss with Interleaver Delay of 64ms with Noise File 6*

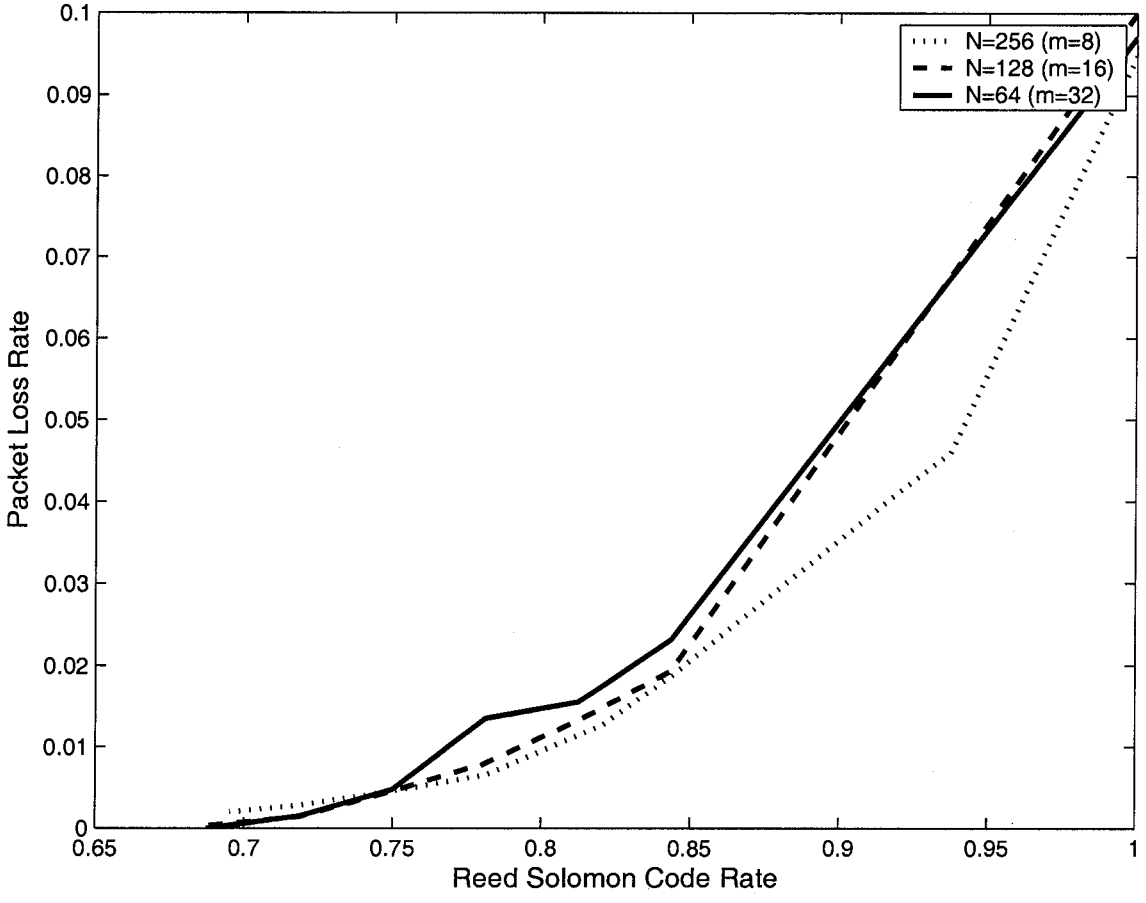

Figure 5.19. *Packet Loss with Interleaver Delay of 256ms with Noise File 3*

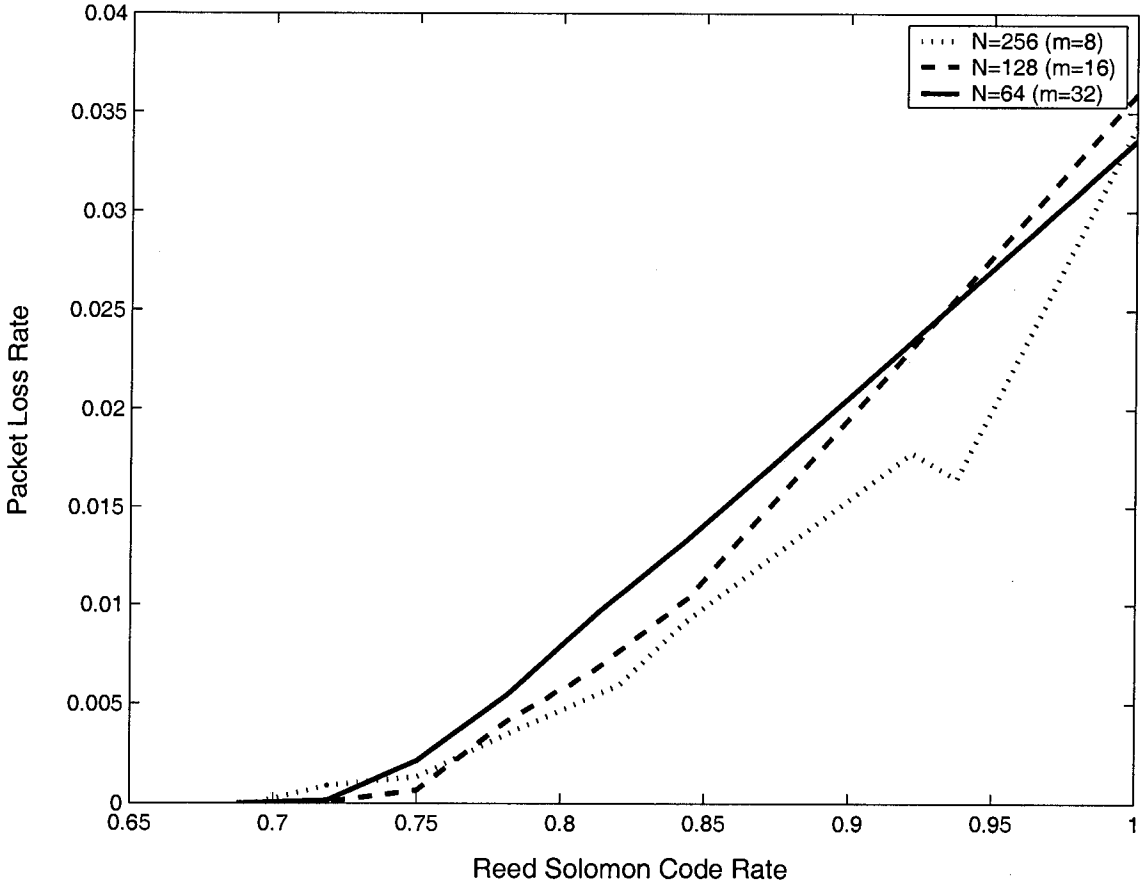

Figure 5.20. *Packet Loss with Interleaver Delay of 256ms with Noise File 4*

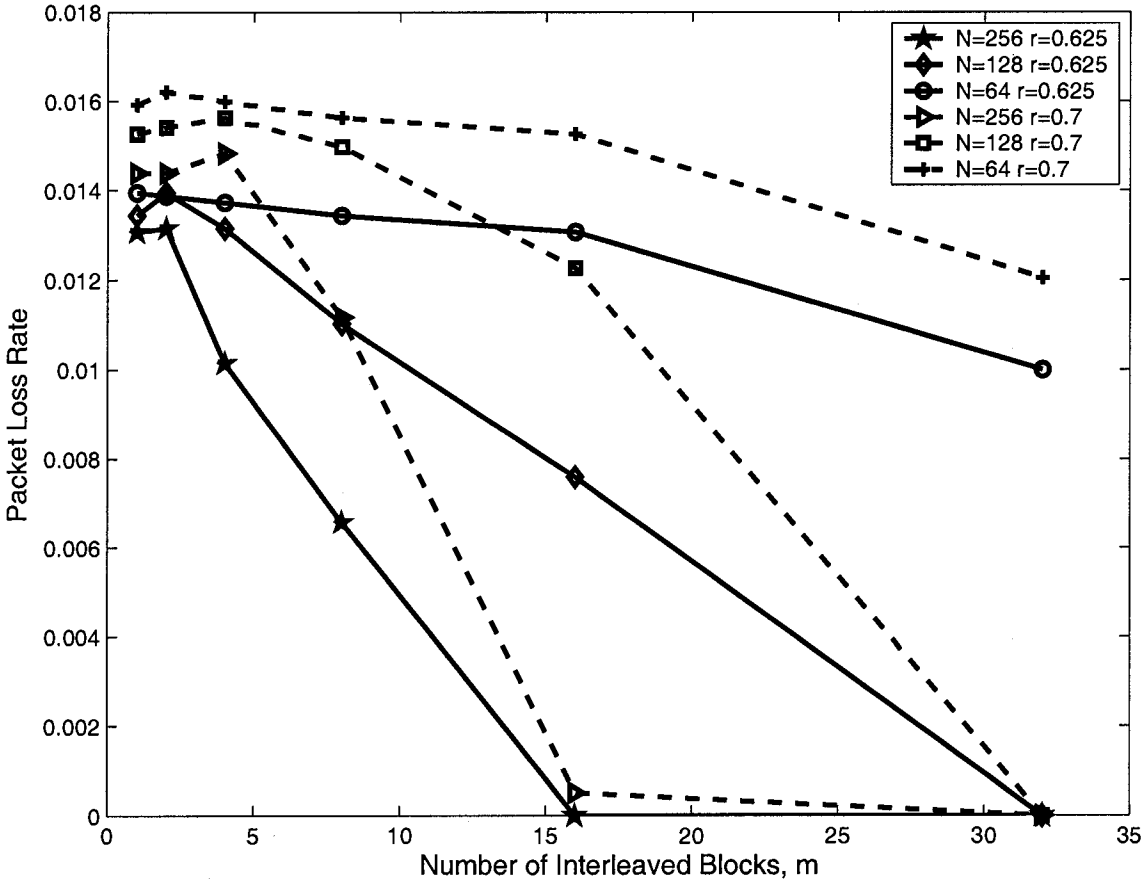

Figure 5.21. *Packet Loss with Noise File 5*

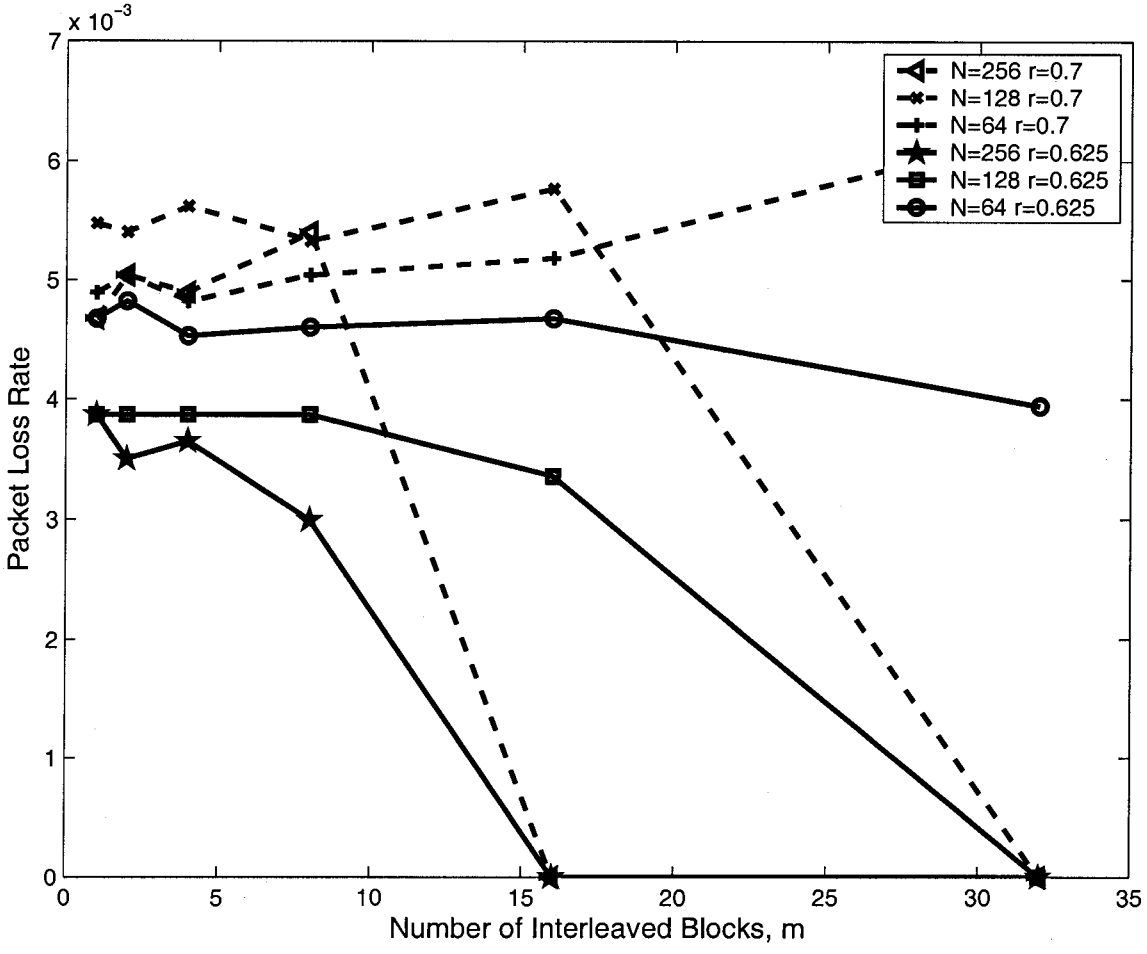

Figure 5.22. *Packet Loss with Noise File 6*

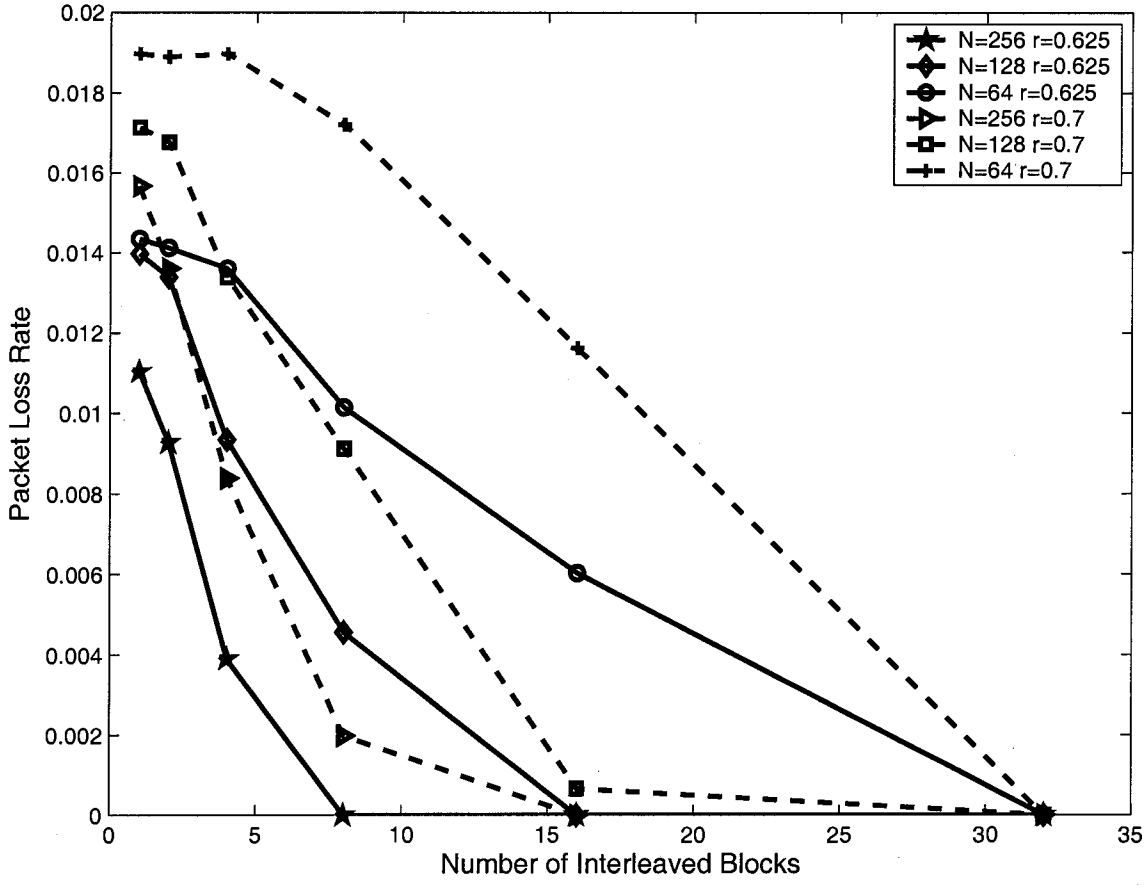

Figure 5.23. *Packet Loss with Noise File 3*

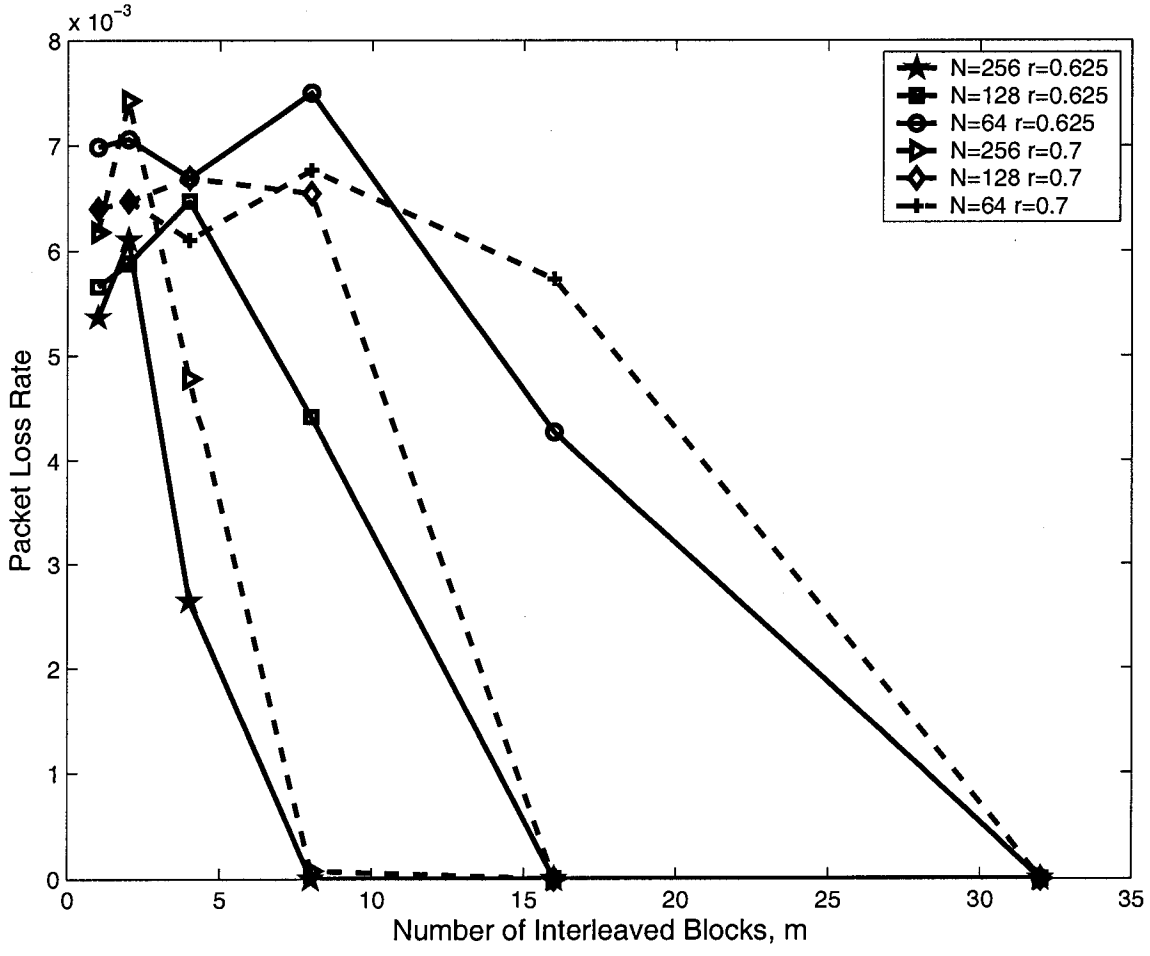

Figure 5.24. *Packet Loss with Noise File 4*

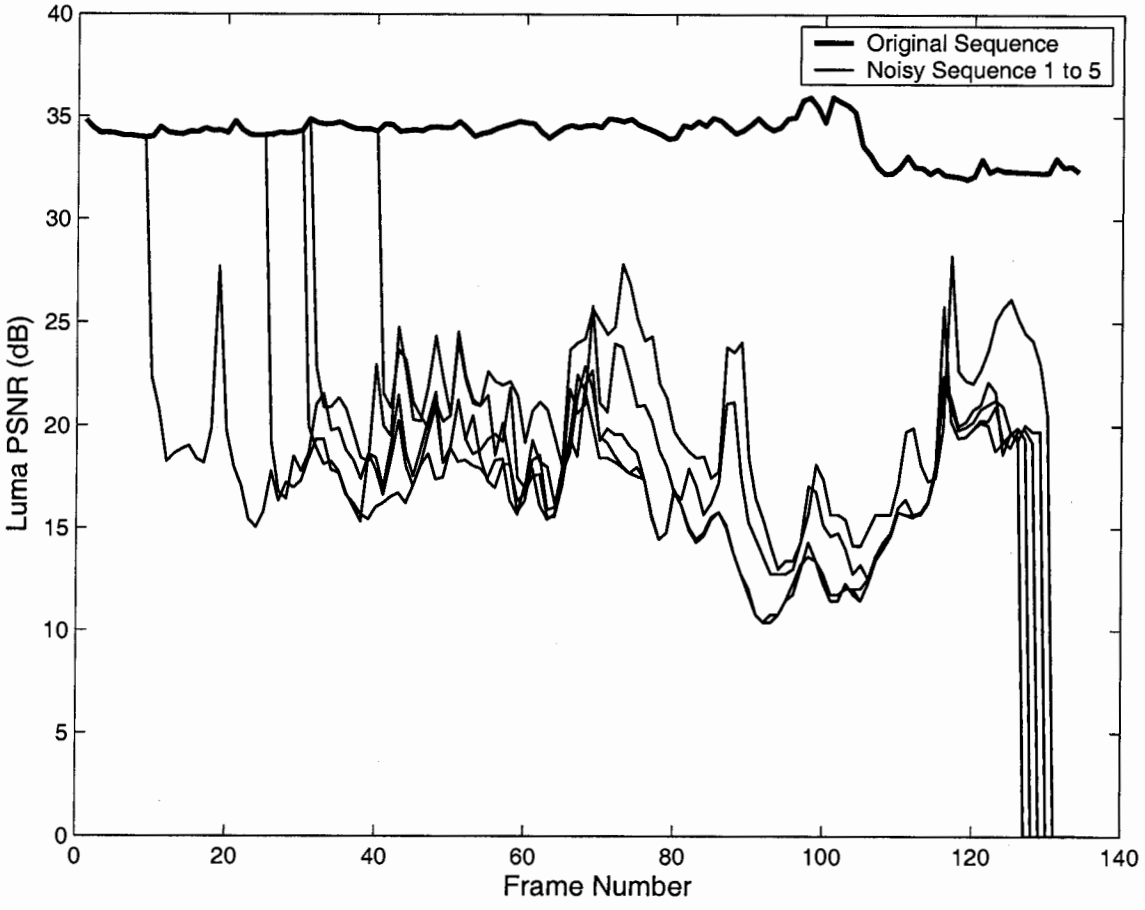

Figure 5.25. *Luma PSNR of Video over Noise File 3 without Error Protection*

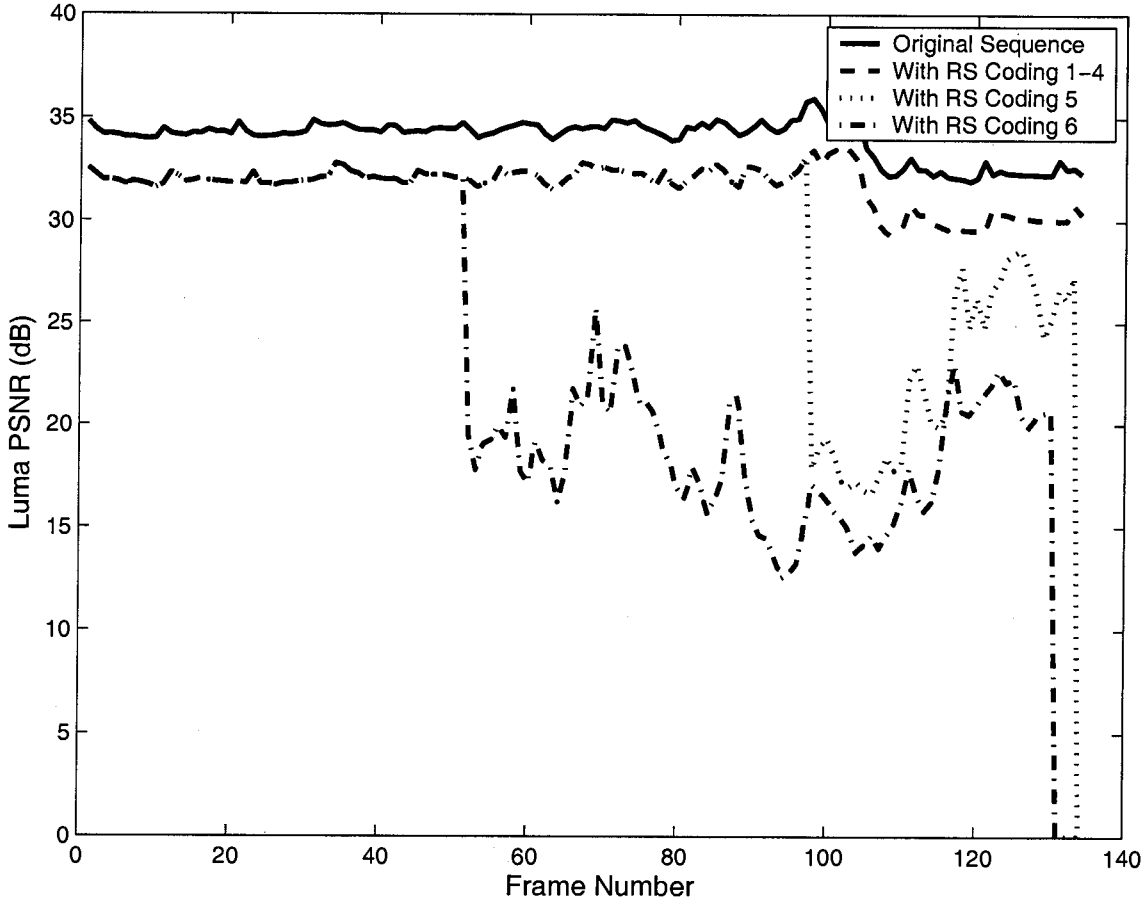

Figure 5.26. *Luma PSNR of Video over Noise File 3 with Reed-Solomon Coding*

Chapter 6

Conclusion

6.1 Summary

Based on common test conditions for 3G WCDMA networks, it has been shown that for the case without any feedback, packet loss can be significantly reduced by using forward error correction. By Reed-Solomon coding every link layer PDU and applying block interleaving, the RTP video packets can be made more robust to channel errors.

For conversational applications with strict delay requirements, a non interleaved Reed-Solomon coded video stream is suggested. However, at bit rates of 128kbps, if a 64ms of delay can be tolerated, a Reed-Solomon code of rate 0.65 or slightly lower with four interleaved blocks can further reduce the packet loss. If this delay cannot be tolerated, or lower bit rates are necessary, then interleaving is not suggested since small interleaver block sizes will often spread the errors to more packets without reducing packet loss, as was discussed in Section 5.5.

In general, a Reed-Solomon block length of $N = 256$ performs best in all cases. A Reed-Solomon block length of $N = 64$ is not recommended since it cannot perform as well as $N = 128$ and $N = 256$. In choosing a Reed-Solomon code rate, a tradeoff has to be made between picture quality (encoded PSNR) and error protection. In this thesis, the prefered choice of code rate ranged between $0.625 - 0.65$. With this code rate, the PSNR of the encoded video sequence is reduced by only approximately 2dB, which is not very noticeable, as was discussed in Section 5.2. Further, since the Reed-Solomon decoding

time can be assumed to be negligible, there is no added delay due to coding. A code rate of $0.625 - 0.65$ can be used to significantly reduce packet loss in most channel conditions and for various bit rates, such as $R_c = 64$ kbps and $R_c = 128$ kbps.

6.2 Future Work

Future work could include a comparison of retransmission based error control techniques (ARQ) with forward error correction techniques. The difference in end-to-end delay as well as the difference in packet loss, or received video PSNR could be evaluated. Further, other types of error correction coding could be applied and compared with Reed Solomon coding. This could include convolutional codes and turbo codes. Since interleaving significantly reduced the packet loss, other types of interleaving, such as convolutional interleavering, could be applied to forward error corrected video, and compared to the block interleaver structure.

This work can also be extended to circuit-switched conversational applications. For circuit-switched applications, the H.264/AVC video codec output would be in bitstream format, as opposed to RTP-packetized format. H.324M is the network protocol for circuit-switched conversational services proposed as an application for H.264/AVC. Finally, other mobile network channel conditions could be tested. This could include GSM, GPRS, and CDMA-2000 channel conditions. These would likely have different bit rates and overhead requirements for packetized video. However, the performance results should be similar for the same bit rates.

Bibliography

[1] Qcif sequences: Foreman video sequence. [Online]. Available: http://trace.eas.asu.edu/yuv/qcif.html

[2] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia*. John Wiley and Sons Inc., 2003.

[3] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.

[4] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Prentice-Hall, Inc., 2001.

[5] T. Stockhammer, M. M. Hannuksela, and T. Wiegand, "H.264/AVC in wireless environments," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 657–673, July 2003.

[6] K. Jeffay. (1999) Advanced distributed systems: Multimedia networking. [Online]. Available: http://www.cs.odu.edu/ cs778/jeffay

[7] C.-J. Tsai. (2004) Distributed multimedia systems. [Online]. Available: http://www.csie.nctu.edu.tw/ cjtsai/courses/dms/

[8] F. Kuo, W. Effelsberg, and J. Garcia-Luna-Aceves, *Multimedia Communications: Protocols and Applications*. Prentice-Hall, Inc., 2002.

[9] R. Steinmatz and K. Nahrstedt, *Multimedia Fundamentals: Media Coding and Content Processing*. Prentice-Hall, Inc., 2002.

[10] S. Wenger, "H.264/AVC over IP," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 645–656, July 2003.

[11] R. Schafer, T. Wiegand, and H. Schwarz, "The emerging H.264/AVC standard," European Broadcasting Union (EBU) Technical Review, January 2003.

[12] M. Karczewicz and R. Kurceren, "The SP and SI frames design for H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 637–643, July 2003.

[13] T. Stockhammer, M. M. Hannuksela, and S. Wenger, "H.26L/JVT coding network

abstraction layer and IP-based transport," *International Conference on Image Processing 2002*, September 2002.

[14] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: a transport protocol for real-time applications," RFC 1889, 1996.

[15] ——, "RTP: a transport protocol for real-time applications," RFC 3550, 2003.

[16] S. Wenger, T. Stockhammer, and M. Hannuksela, "RTP payload format for H.264 video," *Internet Draft, Work in Progress*, Draft-wenger-avt-rtp-h264-01.txt, March 2003.

[17] "3rd gpp; technical specification group core network; ip multimedia call control protocol based on sip and sdp," 3GPP Technical Specification 3GPP TS 24.229.

[18] C. D. Iskander, "Variable bit rate video transmission for code-division multiple-access systems in wideband fading channels," Ph.D. Thesis, University of British Columbia, July 2003.

[19] T. R. Huitika, "A study of packetization and concealment schemes for delivering H.263+ coded video over internet," M.A.Sc. Thesis, University of Victoria, 2002.

[20] J. Postel, "Transmission control protocol," RFC 793, September 1981.

[21] ——, "User datagram protocol," RFC 768, August 1980.

[22] G. Roth, R. Sjoberg, G. Liebl, T. Stockhammer, V. Varsa, and M. Karczewicz, "Common test conditions for RTP/IP over 3GPP/3GPP2," ITU-T SG16 Doc. VCEG-N80, 2001.

[23] T. Wiegand. (2002) Low-delay video transmission over lossy packet-switched networks. [Online]. Available: http://bs.hhi.de/ wiegand/ICG-Project-RDO-Lossy.html

[24] J. G. Proakis, *Digital Communications*, 4th ed. McGraw Hill, 2000.

[25] S. B. Wicker, *Error Control Systems for Digital Communication and Storage*. Prentice-Hall, Inc., 1995.

[26] (2004, January) H.264/avc software coordination, jm7.5c. [Online]. Available: http://bs.hhi.de/ suehring/tml/

[27] (2003, August) H.264/avc software coordination, jm7.3. [Online]. Available: http://bs.hhi.de/ suehring/tml/

[28] V. Varsa, M. Karczewicz, G. Roth, R. Sjberg, T. Stockhammer, and G. Liebl, "Common test conditions for rtp/ip over 3gpp/3gpp2," VCEG-N80, September 2001.

[29] (2003, June) [m4if technotes] common ways to evaluate encoded video quality? [Online]. Available: http://lists.mpegif.org/pipermail/mp4-tech/2003-June/002387.html