UNIVERSITY OF CALGARY

# A ROBUST VIDEO STREAMING PLATFORM OVER WIRELESS NETWORKS

by

Xiao Liu

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

CALGARY, ALBERTA, CANADA

July 2003

THE UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled, "A Robust Video Streaming Platform Over Wireless Networks" submitted by Xiao Liu in partial fulfillment of the requirements for the degree of Master of Science.

Supervisor, Dr. W. Badawy
Dept. of Electrical and Computer Engineering

Dr. Y. Wang
Dept. of Electrical and Computer Engineering

Anil Mehrotra

Dr. A.K. Mehrotra
Dept. of Chemical and Petroleum Engineering

Aug 21, 2003
Date

# ABSTRACT

Delivering video streaming over the varying network with acceptable quality is a challenging task. This is because of the huge size of video data and the complicated characteristics of wireless channels. In this thesis, we present a robust video streaming platform over wireless networks, which aims to provide reliable video transmission in limited bandwidth with acceptable delay.

For efficient video transmission, the Hierarchical Packetization Protocol (HPP) is designed and developed. It generates minimum bytes to describe the compressed video data and maintains an acceptable video quality at very low rate and provides high quality in rich bandwidth. We also propose a novel error control scheme, which is targeted to reduce the influence of bursty errors and bit errors. The scheme consists of three parts: hybrid adaptive unequal error protection (HAUEP), packets shuffling, and dynamic packet length adjustment (DPLA). HAUEP with adaptive parity codes can correct error in short time with controllable redundancy. The main idea of packets shuffling is not only to reduce overall error, but rather to tradeoff continuous error (the unacceptable error) for average error (the acceptable error). DPLA is a mechanism to adjust packet size rapidly according to different network situations. These three modules work together to maintain an acceptable video quality at very high rate of bursty errors or bit errors. Finally, promising results of experiments are presented as well.

ACKNOWLEDGEMENT

I am very grateful to my supervisor, Dr.Wael Badawy, for his intellectual guidance, continuing support and generous encouragement throughout my M.Sc. study. His ideas and suggestions have been invaluable to this thesis.

Many thanks to my colleagues Linsong Cai, Yankun Wei and Mehboob Alam for their insightful suggestions and assistance at every stage of my work.

Last in this list but first in my heart, I would like to thank my wife Hairong for the emotional support and encouragement. I am indebted to my family who stood behind me with prayers.

## TABLE OF CONTENTS

## LIST OF FIGURES

# LIST OF TABLES

# CHAPTER 1

# INTRODUCTION

Video streaming is a technology that delivers large digital video data in real time across computer networks. During the past decade, there has been a significant progress in video streaming over wired networks. With the increase in the next generation wireless networks and rapid growing demand for mobile visual communications, video streaming over wireless networks is becoming possible and has received more attention. This thesis presents an advanced reliable video streaming platform over wireless networks.

## 1.1 Research Background

Unlike text or voice, visual information, in the form of digital images and digital videos, has huge volume in data size. For example, the size of 352 x 288 color image in the Common Interchange Format (CIF) is 152,064 bytes. Transmitting one CIF image over a telephone line at a speed of 56 kbps takes 28 seconds. An NTSC TV resolution in RGB video (480 x 720 pixels) with 30 fps requires 124.416 Mbps bandwidth. Using 3G mobile networks [1] at a bit rate of 2 Mbps, it takes about 62.2 seconds to transmit a 1 second video. Digital image and video data with their original format are too voluminous for transmission over bandwidth limited mobile networks. But they contain a significant amount of redundancy, which makes it possible to be represented in much fewer bits through image and video compression. Spatial and temporal redundancy removal is commonly used in video compression.

The International Telecommunication Union (ITU) has released a number of international standards for video coding, such as H.261 [2] and H.263 [3,4]. In addition, the Joint Photographic Experts Group (JPEG) and the Moving Pictures Experts Group (MPEG) of the International Standards Organization (ISO) have also defined a set of standards for image and video compression, such as JPEG [5], MPEG-1 [6], MPEG-2 [7] and MPEG-4 [8]. These standards perform well for video transmission over noise-free channels. However, in wireless networks, there usually exists severe error corruption due to multipath and different time-varying characteristics of the wireless channels. Since all existing image and video compression standards employ variable-length coding, the compressed bit streams are very sensitive to channel noise. A single error may cause the loss of decoding synchronization and result in an error propagation beyond many data blocks.

In order to combat with channel impairment, error control techniques, Forward Error Correction (FEC) and Automatic Repeat Request (ARQ) [9], are usually employed in data link layer or transport layer to rebuild an error free platform for various applications. These techniques can be considered as network centric error control schemes when the end users do not need to understand the characteristics of the physical channels. Such approaches work well for traditional data communications. However, the characteristics of video communications are quite different from traditional data communications in several aspects. First, video communications usually have strict time constraint. The network centric error control scheme may lead to unaccepted transmission delays. Second, video communications may not need error free transmission, since certain types of degradation may be imperceptible to the human visual system. Hence, direct application of FEC or ARQ will lead to inefficient systems. Therefore, the error control strategy designed in application layer is considered more efficient.

Conclusively speaking, due to the limited bandwidth of wireless channels, video data need to be compressed by highly efficient compression schemes. On the other hand, due to the time varying error prone environments of wireless channels, controlled redundancy is necessary to be added in the compressed bit streams in order to ensure reliable transmissions. Therefore, there is a tradeoff between efficiency and reliability. It's also the key point of designing a video streaming system with high performance.

## 1.2 Research Objective

The objective of this research was to design a robust video streaming platform over wireless networks. The system aims to take advantage of the existed Hierarchical Adaptive Structured (HAS) Video Codec, which was invented by Dr.Badawy [10]. and design corresponding transmission module, and error protection module. The optimal tradeoff between efficiency and reliability for video transmission over wireless channels is the major target of this research.

## 1.3 Thesis Outline

This thesis is organized as follows.

Chapter 2 reviews the technique of video streaming over wireless network, including video compressing algorithms, characteristics of mobile networks and current approaches to control the influences caused by channel errors.

Chapter 3 presents the proposed robust video streaming platform over wireless channel. Four proposed schemes, including Hierarchical Packetization Protocol, Unequal Error Protection, Packet Shuffling and Dynamic Packet Length Adjustment, are discussed in detail.

Chapter 4 introduces the implemented prototype of the video streaming platform. We also present the detailed comparison and analysis of the proposed scheme with traditional schemes. A performance analysis illustrates that the proposed schemes significantly improve the performance of the video streaming system.

Chapter 5 concludes the thesis and outlines areas for future work.

# CHAPTER 2

# VIDEO STREAMING OVER WIRELESS NETWORK

## 2.1 Video Coding Technique

An image is a set of two dimensional (2-D) visual signal in the spatial domain. Video data can be viewed as a sequence of images. So they are three dimensional (3-D) signal, 2-D in spatial domain and 1-D in time domain. Digital video camera digitizes video signals into many samples. These samples are the smallest units and are called picture elements, or pixels. Each pixel value is usually represented by binary data for computer processing. Video data are too voluminous to be stored in digital archives or be transmitted over mobile networks. However, the number of bits to represent the actual information of a video sequence is substantially smaller because significant redundancy exists in most image and videos. There are three types of redundancy in a sequence of images [11]:

- spatial redundancy, which is due to the correlation between neighboring pixel values.
- temporal redundancy, which is due to the correlation between different frames in videos.
- spectual redundancy, which is due to the correlation between different color planes (e.g. in an RGB color image) or spectral bands .

Video compression techniques reduce the number of bits required to represent a video sequence by taking away these redundancies. When compressing a video sequence, one may consider the sequence a series of independent images, and compress each frame using a single image compression method, or one may use specialized video sequence compression schemes, taking advantage of similarities in nearby frames.

Two parameters, *compression ration* (CR) and *peak signal-to-noise ratio* (PSNR), are commonly used in image and video compression field to indicate the objective evaluation for different image and video coding algorithms [11]. They are also used in this thesis.

CR is defined as

$$CR = \frac{\text{number of bits in the original image}}{\text{number of bits in the compressed bitstream}} \tag{2.1}$$

PSNR is defined as

$$PSNR = 20\log_{10}(\frac{255}{RMSE}), \tag{2.2}$$

where RMSE denotes root-mean-squared error,

$$RMSE = \sqrt{\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}[f(i,j) - \hat{f}(i,j)]^2}\ . \tag{2.3}$$

Where $f$ denotes the original 8-bit N x N image, and $\hat{f}$ denotes the reconstructed image.

### 2.1.1 Intraframe compression

Intraframe Compression is the compression within individual frames, which is also known as spatial compression. Intraframe compression consists of three main steps [11]: Transform, quantizing and coding, as illustrated in Figure 2.1.

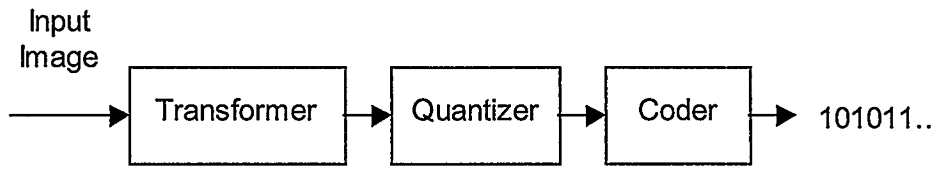



Figure 2.1. The three steps of digital image compression

The purpose of the transform is to reorganize the data, to make it possible for the encoder to do a better job. For statistical coders, the transform can typically be to give the data a representation featuring non-uniform probability distribution.

The quantizing step is used to remove or reject information that is regarded uninteresting. What is considered uninteresting depends on how the image is supposed to be used later. If the image is targeted at a human observer, which is the case for the video images covered by this thesis, the quantizing will typically remove details which are not distinguishable d by our visual system.

The coder produces the resulting bitstream using a general compression algorithm.

2.1.1.1 Discrete Cosine Transform

Discrete Cosine Transform (DCT) is a typical block-based transform. In DCT coding, first, an image is divided into many small blocks, then DCT is applied to each block. In this way, the original samples in pixel domain are transformed into frequency domain, where the frequency coefficients are statistically independent. The decorrelation results in the signal energy being redistributed among only a small portion of frequency coefficients. This transform does not cause any information loss.

The forward 2-D DCT of an $N$ x $N$ block of pixels is often defined as [11]:

$$S(u,v)=\frac{2}{N}C(u)C(v)\sum_{y=0}^{N-1}\sum_{x=0}^{N-1}s(x,y)\cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N} \tag{2.4}$$

Where, $u=0,1,...,N$, $v=0,1,...N$ and

$$C(u),C(v)=\begin{cases}1/\sqrt{2} & u,v=0\\ 1 & otherwise\end{cases}$$

And the corresponding inverse DCT (IDCT) is defined as follows:

$$s(x,y)=\frac{2}{N}C(u)C(v)\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}S(u,v)\cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N} \tag{2.5}$$

Where, *C(u)* and *C(v)* are as defined above.

DCT is widely used in image and video coding. It is the basic of the still image coding standard JPEG[5], and many video coding standards such as H.261 [2], H.263 [3], MPEG-1 [6], MPEG-2 [7], MPEG-4 [8].

2.1.1.2 Quantization

Quantization is the process of representing a set of continuous valued samples with a finite number of states. There are two types of quantization: Scalar Quantization and Vector Quantization. In scalar quantization, each input symbol is treated separately in producing the output, while in vector quantization the input symbols are clubbed together in groups called vectors, and processed to give the output. In video intraframe processing, scalar quantization techniques are commonly used.

A quantizer can be specified by its input partitions and output levels (also called reproduction points). If the input range is divided into levels of equal spacing, then the quantizer is termed as a Uniform Quantizer, and if not, it is termed as a Non-Uniform

Quantizer. A uniform quantizer can be easily specified by its lower bound and the step size. Figure 2. 2 shows a uniform quantizer. If the input falls between *nr* and *(n+1)r*, the quantizer outputs the symbol *n*.

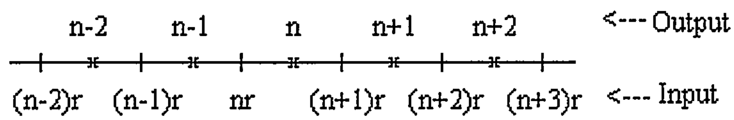



Figure 2. 2. A uniform quantizer

Inversely, a dequantizer is one which receives the output levels of a quantizer and converts them into normal data, by translating each level into a reproduction point in the actual range of data. If given the output levels or partitions of the encoder, the best decoder is one that puts the reproduction points x' on the centers of mass of the partitions. This is known as centroid condition. If given the reproduction points of the decoder, the best encoder is one that puts the partition boundaries exactly in the middle of the reproduction points, i.e. each x is translated to its nearest reproduction point. This is known as nearest neighbour condition. The quantization error (x - x') is used as a measure of the optimality of the quantizer and dequantizer.

2.1.1.3 Variable length coding

Before coding, the quantized block is converted to a sequence of numbers by collecting coefficients according to the zig-zag sequence in Figure 2.3.

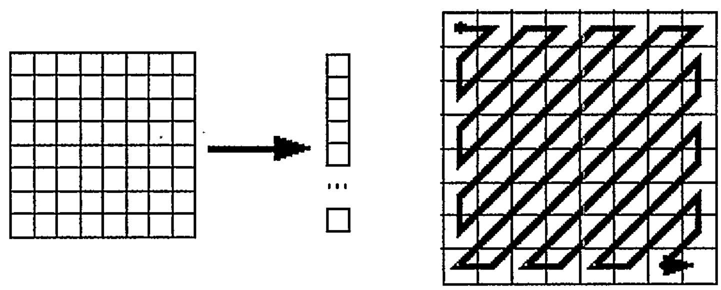


Figure 2.3. The zig-zag sequence

The zig-zag sequence orders the coefficients in approximately decreasing importance, collecting the more heavily quantized values towards the end. This ordering typically gives runs of zero values, which are runlength encoded. Non-zero values are coded using either Huffman or arithmetic coding. Huffman coding yields the optimal integer prefix codes given a source with a finite number of symbols and their possibilities. In prefix codes, no codeword is a prefix of another codeword. Such codes are uniquely decodable since a given binary string can only be interpreted in one way. Huffman codes are optimal in the sense that no other integer-length VLC can be found to yield a smaller average bit-rate.

2.1.1.4 Decompression

Decompressing a compressed stream to an image resembling the original is done using the pipeline in Figure 2.4. The process is the reverse of coding.

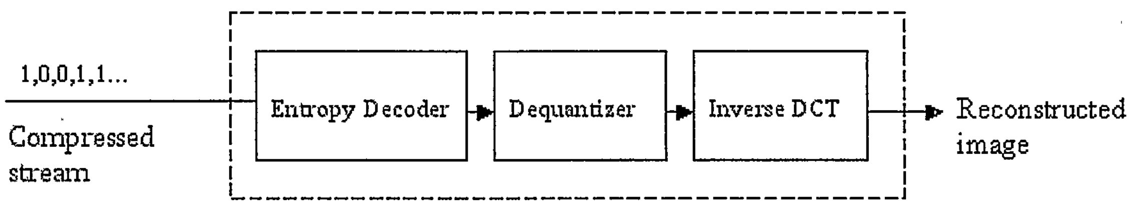



Figure 2.4. Pipeline for DCT-based decoding

### 2.1.2 Interframe compression

Interframe compression is applied to a sequence of video frames, rather than a single image. In general, relatively little changes from one video frame to the next. Interframe compression exploits the similarities between successive frames, known as temporal redundancy, to reduce the volume of data required to describe the sequence. There are several interframe compression techniques. Block based method and mesh based method are commonly used [12].

2.1.2.1 Block based method

Block-based motion estimation and compensation are among the most popular approaches. Block-based motion compensation has been adopted in the international standards for digital video compression, such as H.261, MPEG-1 and MPEG-2. Although these standards do not specify a particular motion estimation method, block-based motion estimation becomes a natural choice.

Block-motion model assumes that the image is composed of moving blocks and all the pixels in a certain block have only one motion vector. Block based motion compensation is completed in three stages. First, the frame to be approximated, the current frame, is divided into uniform non-overlapping blocks. Then each block in the current frame is compared to areas of similar size from the preceding or past frame in order to find an area that is similar. A block from the current frame for which a similar area is sought is known as a target block. The location of the similar or matching block in the past frame might be different from the location of the target block in the current frame. The relative difference in locations is known as the motion vector. If the target block and matching block are found at the same location in their respective frames then the motion vector that describes their difference is known as a zero vector. Finally, when coding each block of the predicted frame, the motion vector detailing the position (in the past frame) of the target block's match is encoded in place of the target block itself. Because fewer bits are required to code a motion vector than to code actual blocks, compression is achieved.

During decompression, the decoder uses the motion vectors to find the matching blocks in the past frame (which it has already received) and copies the matching blocks from the past frame into the appropriate positions in the approximation of the current frame, thus reconstructing the image. In the example used above, a perfect replica of the coded image can be reconstructed after decompression.

2.1.2.2 Mesh based method

Mesh-Based model offers a lot of functionalities that the Block-Based model does not offer. While the Mesh-Based modes are suitable for patially continuous representation of in-plane rotations,zooms,and other motions that can be represented by parametric mappings, the Block-Based models perform better around motion boundaries. 2D mesh [13-15] has gained widely application in video processing. Two types of 2D mesh structures, i.e., uniform and non-uniform mesh, are used in MPEG-4 standard, which deals with object-based multimedia compression and composition. Meanwhile, mesh technique allows comprehensive video processing, including compression, manipulation, searching, browsing, and object tracking, etc. Especially, hierarchical representation of 3D / 2D static and 2D dynamic meshes have attracted attention, because it 1) provides rendering at various levels of detail (quality scalability), 2) allows progressive and scalable transmission or storage of the object geometry and motion information, and 3) enables improved tracking performance (compared to a non-hierarchical representation) in the case of 2D dynamic meshes.

### 2.1.3 Video coding standards

Currently, H.261,H.263, MPEG-1, MPEG-2 and MPEG-4 are the most common video standards. Each of them targets a multimedia application.

2.1.3.1 ITU-T Recommendations H.261

This is a video coding standard published by the ITU (International Telecom Unit) in 1990. It was designed for data-rates which were multiples of 64Kbps. The standard covers the bandwidth from 64Kbps to 2Mbps.

H.261 supports two resolutions: Common Interchange Format (CIF) at 352 x 288 pixels, and Quarter CIF (QCIF) at 176 x144 pixels. Frames for the three components are partitioned in blocks of 8 x 8 pixels, each of which is transformed, quantized and Huffman-coded separately. A macroblock is defined as four neighboring luminance blocks, and one block from each of the chrominance components, making up a 16 x 16 sub-image.

Two types of frames are defined in H.261, intra coded frames and inter coded frames. Intra coded frames are coded as stand-alone frames, while inter coded frames use prediction errors with respect to the previous frame. The coded blocks of inter coded frames may include motion compensation, in which case a motion vector is associated with each macroblock. The motion vector allows specification of a displacement of up to 15 pixels in all directions. The sender may decide not to send blocks that haven't changed since the previous frame.

#### 2.1.3.2 ITU-T Recommendations H.263

H.263 works much like H.261, but there are several extensions, and some modifications. In addition to the two resolutions defined for H.261, H.263 allows the following: 16CIF at 1408 x1152, 4CIF at 704 x576, and sub-QCIF at 128 x96 pixels.

As extensions to H.261 includes "PB-frames mode", where two frames are coded as one unit. The latter frame is coded as an intra frame, while the former frame is coded in inter mode, possibly using bidirectional prediction between the previously seen frame, and the intra coded frame of the same unit.

Another extension is the use of unrestricted motion vectors, where motion vectors are allowed to point outside the frame. Edge pixels are used for prediction of the non-existing pixels. In H.263, motion vectors use half pixel prediction, instead of integer pixel prediction.

#### 2.1.3.3 MPEG-1 (Moving Picture coding Experts Group)

The MPEG-1 standards specify coding of video and audio streams, and how synchronization between them is supposed to be done. MPEG defines three different types of frames, as illustrated in Figure 2.5.

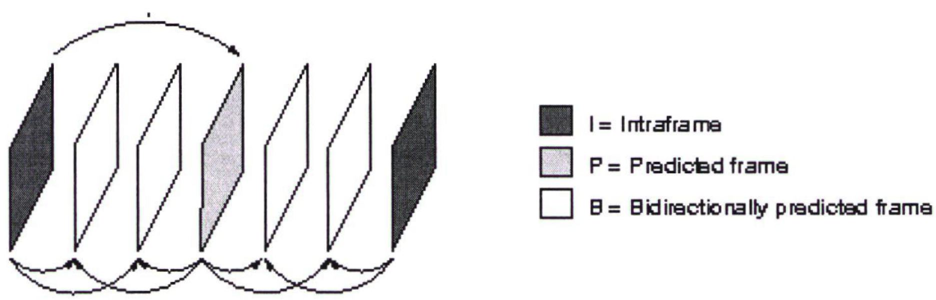



Figure 2.5. The relationship between frame types

Intraframes, or I-frames, defines the start of a group of frames. I-frames are coded as stand-alone images, using a method resembling the one described for JPEG .

A group of frames may contain predicted frames, called P-frames. These are predicted from the closest, previous I- or P-frame, with the help of motion compensation vectors. The motion vectors are associated with macroblocks of 16 x16 pixels.

Between the I- and P-frames, there may be zero or more bidirectionally interpolated frames, or B-frames. These are interpolated between the nearest I- or P-frames.

2.1.3.4 MPEG-4 standard

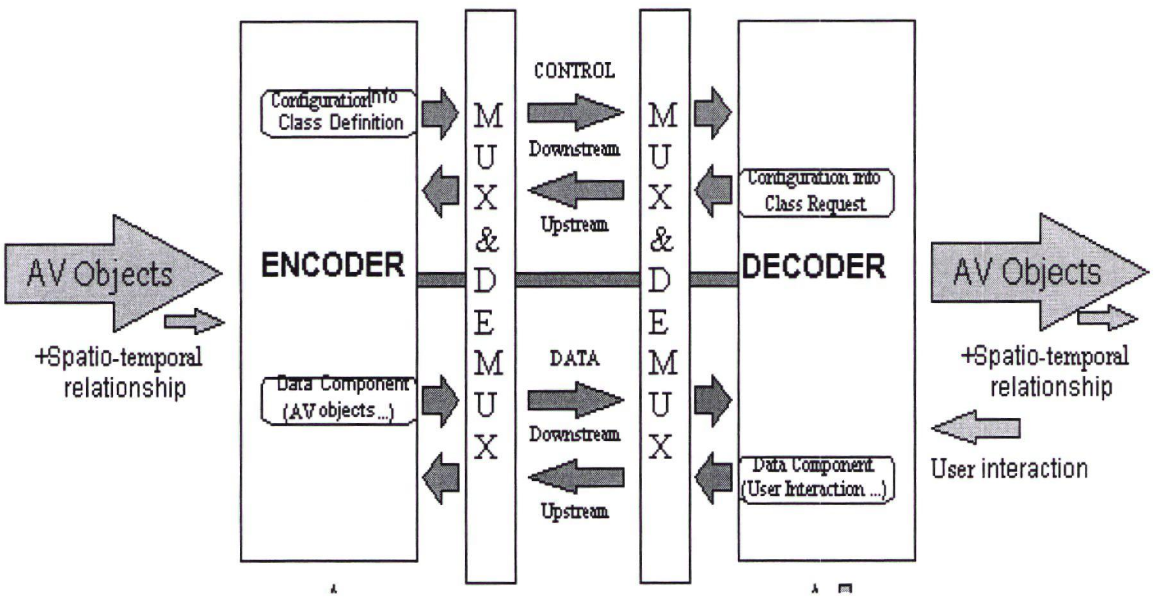



Figure 2.6. Overall architecture of a MPEG-4 system

The MPEG-4 standard provides multimedia coding schemes that are suitable for low bit rate video applications, i.e., data rates less 64Kbps. MPEG-4 is based on the segmentation of audiovisual scenes into "Audio/Visual Objects" (AVOs) which can be multiplexed for transmission over heterogeneous networks. Figure 2.6 shows the MPEG-4 framework. The encoder compresses the AV objects, may apply some error protection, multiplexes the multiple bitstreams and transmits it downstream to the decoder. At the decoder, several steps are taken: Demultiplexing, error correction, decompression. Composition and presentation to an end user. The end user may interact with the presentation, thus the encoder has to react on user input either locally, or may need to send information upstreeam to the encoder.

## 2.2 Communication and Mobile Network Background

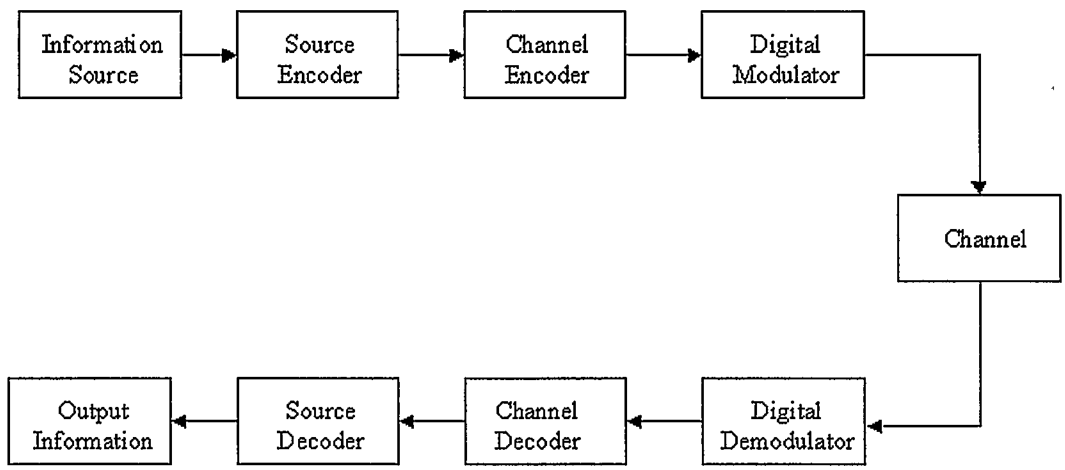



Figure 2.7. Digital communication system

A typical video digital communication system is shown in Figure 2.7 [20]. The function of source encoder is to convert the input video signal into a sequence of binary digits. As mentioned before, source encoding seeks an efficient representation of source signal that results in little or no redundancy. The purpose of the channel encoder is to introduce some controlled redundancy so that the receiver can use it to overcome the effects of noise and interference encountered in the transmission of the signal through the channel. Since nearly all of the communication channels used in practice are capable of transmitting electrical signals (waveforms), the primary purpose of digital modulator is to map digital information sequence into signal waveforms. Finally, the communication channel is the physical medium that is used to send the signal from the transmitter to the receiver. In this thesis, the wireless channel is focused.

### 2.2.1 Characteristics of Wireless Channels

In wireless communications, besides absorption, the propagation of electromagnetic waves is influenced by three basic mechanisms: reflection, diffraction, and scattering [21]. In conjunction with mobility of the transmitter and receiver, these mechanisms cause several phenomena, such as time-varying delay spread or spectral broadening, which can severely impair the transmission.

When a mobile moves over a large area or a long time intervals, the distance between the transmitter and receiver often varies significantly. Furthermore, the number and type of objects between the transmitter and receiver usually change and might cause shadowing. The resulting attenuation of radio frequency (RF) power is described by the path loss. Usually, path loss can be modeled as a function of distance ($n^{th}$ power law) and a random variation about the mean (log-normal distribution).

In addition to large-scale fading, there also exists small-scale fading, i.e., small changes in position can result in dramatic variations of RF energy. This small scale fading is caused by multi-path propagation. In a wireless communication system, a signal can travel from transmitter to receiver over multiple reflective paths. The signal will have different delay and attenuation in different paths. The superposition of these individual signal components can cause constructive and destructive interference alternating at a small scale (as small as half wavelength). For a moving receiver, the space-variant signal strength is also time-variant, where the speed of fluctuation is determined by the velocity of the mobile terminal. If the number of the multiple paths are large and each of them is equally significant, the envelop of the received signal can be modeled by Rayleigh distribution [22].

In summary, a mobile radio channel is both time-variant and space-variant at both large-scale and small scale. As a result, errors are not limited to single bit errors but tend to occur in bursts. In severe fading situations, an intermittent loss of connection may happen. Therefore, to design error control techniques for mobile communications is a major challenge.

### 2.2.2 Channel Models

Using channel models is very convenient for designing a communication system if the channel model can reflect the most important characteristics of the physical channels. There are three frequently used channel models in practice: the additive noise channel, the linear filter channel, and the linear time-variant filter channel [22,30].

The additive noise channel model can be expressed as

$$r(t) = \alpha s(t) + n(t), \tag{2.6}$$

where $\alpha$ is an attenuation factor. It means that the received signal $r(t)$ can be viewed as the summation of the attenuation of the sent signal $s(t)$ plus an additive random noise $n(t)$. If the $n(t)$ is the additive white Gaussian noise (AWGN) process, the channel model is called the AWGN channel.

The linear filter channel model can be expressed as

$$r(t) = s(t) \otimes c(t) + n(t), \tag{2.7}$$

where $c(t)$ is the impulse response of a linear filter and $\otimes$ denotes convolution. In this case, the channel is characterized as a linear filter with additive noise.

The linear time-variant filter channel can be modeled as

$$r(t) = s(t) \otimes c(\tau;t) + n(t), \tag{2.8}$$

where $c(\tau;t)$ is the response of the channel at time $t$ due to an impulse applied at time $t-\tau$. Mobile radio channels is a special case of

$$c(\tau;t) = \sum_{k=1}^{L} a_k(t)\delta(\tau - \tau_k). \tag{2.9}$$

Therefore,

$$r(t) = \sum_{k=1}^{L} a_k(t)s(\tau - \tau_k) + n(t) \tag{2.10}$$

In this case, in addition to additive noise, the received signal consists of $L$ multi-path components, where each component is attenuated by $a_k(t)$ and delayed by $\tau_k$. When the impulse response $c(\tau;t)$ is modeled as a zero-mean complex-valued Gaussian process, the envelope $|c(\tau;t)|$ at any instant $t$ is Rayleigh distributed. This is the well-known Rayleigh channel model.

The channel models introduced above are analogue channels or waveform channels. Sometimes, people only focus on studying the source coding and the channel coding. In this case, for simplicity, the modulator and the demodulator can be included as part of the channel as shown in Figure 2.8. Such channel model is called discrete channel model [16]. i.e., discrete-input and discrete-output.

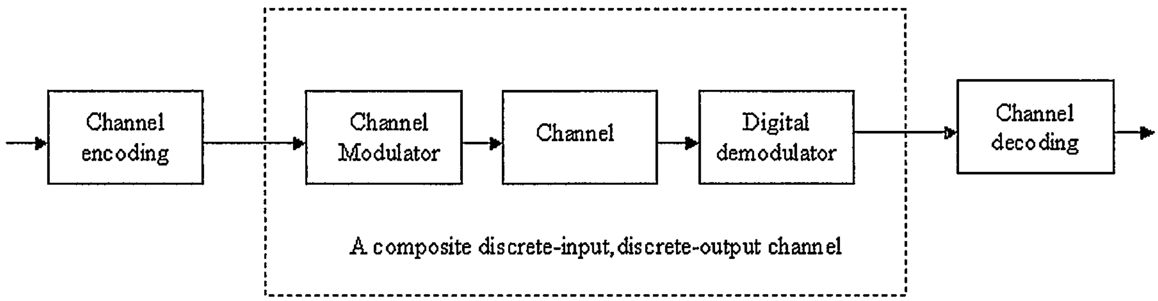



Figure 2.8. A composite discrete channel

There are two commonly used discrete channel models: the binary symmetric channel (BSC) model for memoryless channels and the finite-state Markov channel (FSMC) model for bursty channels. The BSC is a binary-input, binary-output, symmetric channel as shown in Figure 2.9, where $p$ is the average bit error rate. For the FSMC model, the simplest case is the two-state Markov channel, which is called Gilbert-Elliott channel (GEC). The GEC model, as shown in Figure 2.10, has two states: Good state and Bad state. In this case, each state will be considered as a BSC. The parameters of GEC include $T_{gb}$ and $T_{bg}$ which represent the transition probabilities from one state to the other. $e_G$ and $e_B$ are the BERs at Good state and Bad state, respectively. $P_G$ and $P_B$ are the probabilities staying at Good state and Bad state, respectively. The average BER of such channel can be written as $BER = P_G \times e_G + P_B \times e_B$. At the case of $e_G = 0$ and $e_B = 1$, the GEC channel model is termed as the simple GEC model. Many studies show that the GEC channel model is able to capture the bursty nature of the mobile channel errors.

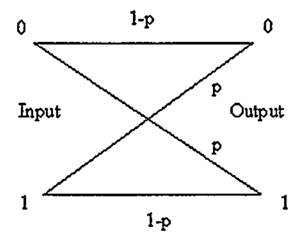



Figure 2.9. The BSC model

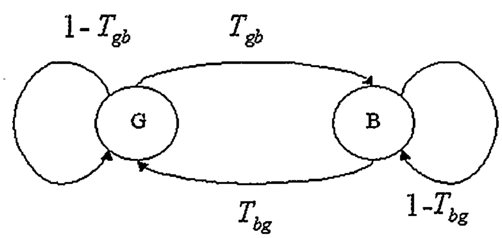



Figure 2.10. The GEC model

### 2.2.3 Wireless Network and QoS Parameters

Wireless networks are first introduced in the early 1980s. The first generation mobile communications provided analog voice communications to mobile users. The major first-generation standards include AMPS, TACS and NMT.

The second generation mobile system provided not only digital voice communications but also data services, mainly circuit-switched low to medium bit rate (9.6 kbps). The major second-generation standards include global system for mobile communications (GSM) and IS-95.

With the increasing demand of providing wireless Internet service, the third generation mobile communications, international mobile telecommunications in the year 2000 (IMT-2000), is introduced. The third generation system adds multimedia capabilities such as support for high bit rates and introduction of packet data and IP access. The bit rates of IMT-2000 are defined as 384 kbps for full area coverage and 2 Mbps for local area coverage.

As one of the most popular assess network, wireless local area network (WLAN) has grown rapidly since the introduction of 802.11b WLAN standards, which offer performance more nearly comparable to Ethernet. IEEE 802.11b can deliver raw data rates up to 11 Mbps over the 2.4 Ghz band. The majority of WLAN systems in the world today follow this standard.

The progress of wireless networks increases the bandwidth of wireless data transmission, which makes video streaming over wireless networks possible.

To evaluate the performance of wireless networks, normally three Quality of Service (QOS) parameters are used, including throughput, error rate and delay. The throughput of a network is the effective bit rate the network can provide. Because image and video signals have huge volume of data size, real-time applications typically requires a bandwidth about 0.4 to 1.4 Mbps. This throughput is often required only for unidirection because a lot of multimedia traffic is highly asymmetric.

There are two kinds or error rates: bit error rate (BER) and packet error rate (PER). Bit error is usually casued by channel noise and interference. While packet error is usually caused by network congestion and delay.

As for delay, real-time video application requires bounded end-to-end delay, i.e., every video packet must arrive at the destination in time to be decoded and displayed. For example, the ITU standards suggest a maximum total end-to-end delay of up to 150 ms for interactive video application. Delay variation or jitter is another important parameter, which should be considered in a video transmission system.

### 2.2.4 Traditional Error Control Techniques

To achieve reliable digital communications, error control techniques are applied during the transmission. The error control techniques can be classified into two categories: Forward Error Correction (FEC) and Automatic Repeat Request (ARQ) [35]. ARQ needs a feedback channel to send retransmission requests, while FEC does not have such requirement.

#### 2.2.4.1 Forward Error Correction (FEC)

FEC techniques can be divided into two classes: block coding and convolutional coding. Block coding needs to break the input data into blocks while convolutional coding is a continuous process without the need for assembling blocks. Convolutional code is also called trellis code. A (N,K,M) convolutional code denotes this convolutional code has N outputs, K inputs and M is the memory of the encoder. Therefore, at any given time, the N outputs depend not only on the K inputs but also on the previous M inputs. The channel coding rate is defined as

$$R = \frac{K}{N}. \tag{2.11}$$

Convolutional encoding is easy to implement both in hardware and software while the decoding process is more complicated. The most popular decoding approach is the Viterbi algorithm.

2.2.4.2 Automatic Repeat Request

Similar block coding, In ARQ, the input data is grouped into blocks. Each block is extended to a packet by adding a header, including a Sequence Number (SN), and appending an error detection code, such as cyclic redundancy check (CRC). The receiver uses the SN to determine whether there exists lost packets or not, and uses CRC to check whether the received packet is in error or not. If there are lost packets or corrupted packets, the receiver will request the retransmission by sending back Positive Acknowledgements (ACKs) or Negative Acknowledgements (NAKs) via the feedback channel. Usually, the retransmissions are repeated until data are received error-free or a time-out is exceeded.

There are three basic ARQ schemes: Stop And Wait (SW), Go Back N (GN) and Selective Repeat (SR). In the SW-ARQ scheme, the transmitter sends a packet and waits for an acknowledgement. The SW-ARQ is simple but not very efficient. In the GN-ARQ scheme, the transmitter sends packets continuously. In other words, it does not wait for an acknowledgment. If the transmitter receives an NAK, it will retransmit the indicated error packet and all subsequent packets. In SR-ARQ, the transmitter only resends the indicated error packets. It provides the highest throughput. ARQ schemes can be combined with FEC, termed as Hybrid ARQ, to further improve the performance.

One critical parameter in ARQ is the Round Trip Delay (RTD). If the number of retransmission attempts is A, the total delay until reception is D=A*RTD. Because A depends on the quality of the channel, the resulting delay and throughput vary over time and are not predictable. Therefore, for delay-constraint applications, the maximum number of retransmissions has to be limited.

## 2.3 Problem Statement

Comparing with video streaming over traditional networks, such as internet, wireless video streaming is a more challenging task because of the characteristics of wireless channels: (1) limited bandwidth; (2) time-varying error-prone environment. Moreover, in addition to severe bit errors in wireless channel, wireless video streaming may also encounter packet–loss problems. For example, in a typical wireless video streaming system, the compressed video data are stored in a server, which is usually located in a wired line network such as Internet. When a mobile user requests an access to the video data, the video server first sends the compressed video bitstream through the wireline network to the base station or access point located at the boundary between the wireline and the wireless networks. The base station then transmits the video bitstream via the wireless channels to the mobile unit. In this case, since the data are transmitted over both wired line and wireless networks, the transmission may be impaired by both packet-loss and bursty bit errors.

Many studis show that wireless channel errors are not limited to random signal bit errors; instead, the errors tend to occur in bursts because radio channels are both time-varying and space-varying at both large-scale and small-scale. On the other hand, packet-loss in the wireline network can be viewed as a special case of channel errors, i.e., bursty errors with known length and positions. It is in this spirit that we treat both packet-loss and bit errors as bursty loss or bursty errors. Therefore, the same error-control schemes designed for packet-loss channels may be also appropriated for wireless channels. In addition, the design of a wireless video streaming system should consider the following requirements:

- The bit-rate of the scalable video bitstream source should be able to change from 10 kbps to several hundred kbps in order to be adaptive to the variations of the wireless channels.
- Be robust to the bursty errors of wireless channels.

## 2.4 Related Work on Robust Video Streaming over wireless networks

With wireless network moving quickly to broadband, video streaming over wireless network has been widely studied [28,29,36-38]. As we just discussed, wireless network cannot provide guarantee on bandwidth, delay or loss. To combat these problems, there are two existing schemes. One is network-centric approaches, i.e., carefully designing the network equipment to provide QoS support. The other is end-system based approaches, which do not impose any requirements on the network. Those network-centric approaches are beyond the scope of our study. In this research, we focus on end-system based approaches.

End-system based approaches consist of two components: congestion control and error control. The target of congestion control is to reduce packet loss and delay by matching the video sending rate with the network bandwidth. Because the bandwidth of wireless network is time-varying, it is desired that the video signals be sent in a network-friendly way. Rate adaptive video encoding and rate shaping are two commonly used approaches.

Unlike congestion control, which is to prevent packet loss, the purpose of error control is to maximize the reconstructed video quality in presence of packet loss. The error control techniques can be divided into three categories: approaches at the encoder end, approaches at the decoder end, and approaches requiring interactions between the encoder and the decoder.

Approaches at the encoder end include error-resilient encoding and FEC-based robust encoding. Commonly used error-resilient approaches include inserting resynchronization marker, using reversible variable length codes (RLVC) and using intra block/frame refreshment. Inserting resynchronization marker uses the same approach as that in error-resilient image coding. RVLC are the codes which can be decoded in two direction. Using intra block/frame refreshment is to limit error propagation in time direction. FEC-based robust encoding includes joint source-channel coding and UEP-based layer video transmission.

One of the main error control approaches at the decoder end is error concealment. In addition to the spatial concealment and frequency concealment, another one for video signals is temporal concealment. For example, the motion vector of a lost MB can be reconstructed from the MVs of the neighbor MBs, and the referenced MB can be used for concealment. There are also other error concealment techniques such as the combinations of the three basic error concealment techniques.

When a feedback channel is available, encoder and decoder can have an interactive error control, i.e., the decoder can inform the encoder about which part of the transmitted information is lost or corrupted by errors, and the encoder can adjust its operation correspondingly to suppress or even eliminate the effect of such errors. The reference picture selection (RPS) and error tracking are two typical feedback-based error control schemes to limit error propagation in time direction. In RPS, if the encoder learns about damaged parts of a previously coded frame through a feedback channel, it can decide to code the next P-frame associated to not the most recent frame but an older reference frame, which is available in the decoder. In the error tracking approach, through the feedback information, the encoder can track the damage areas, and use some methods like intra-mode coding to stop the propagation.

# CHAPTER 3

# THE PROPOSED ROBUST VIDEO STREAMING PLATFORM OVER WIRELESS NETWORK

The proposed video streaming platform consists of two parts: video codec and the video transmission mechanism which could be further divided into: the packetization module and the error control module. In the error control unit, three novel schemes, Hybrid Adaptive Unequal Error Protection (HAUEP), Packet Shuffling (PS) and Fast Dynamic Packet Length Adjustment (DPLA), are proposed. Figure 3.1 illustrates these components in the proposed video streaming platform.

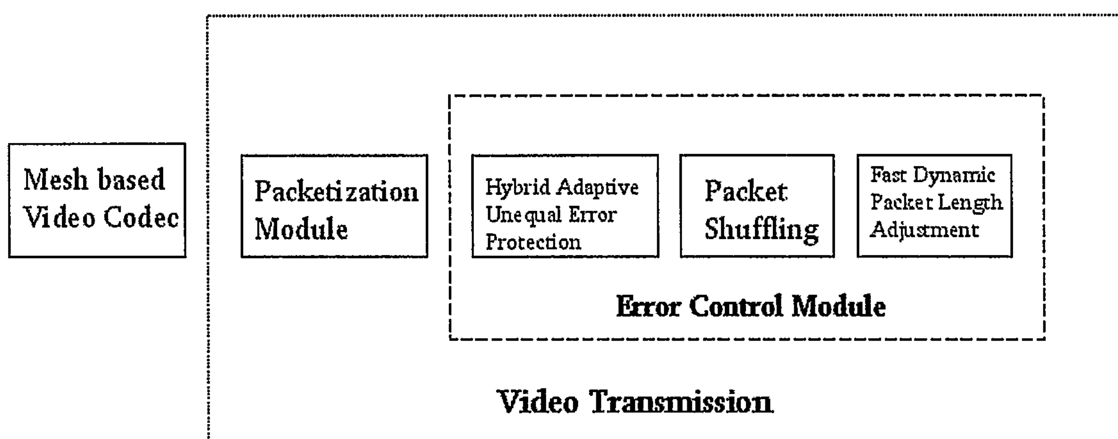



Figure 3.1. Components in the proposed video streaming platform

## 3.1 Mesh based video codec

Video codec is the engine of the video streaming system. In this research, we employed the Hierarchical Adaptive Structured (HAS) Video Codec [63,64]. The mesh model in this codec represents the motion more efficiently than traditional standard codecs, such as MPEG-1, MPEG-4 and H.263. In particular, it can provide better performance in the narrow-bandwidth network. Most video application vendors in current industry use their own video compression algorithms while they provide products with standard algorithms. It's an important add-on feature in most video products to cope with the competitors [29]. Hence, even HAS codec doesn't use the standard video compress algorithm, its outstanding performance still make this study valuable and applicable.

### 3.1.1 Structure of mesh based video codec

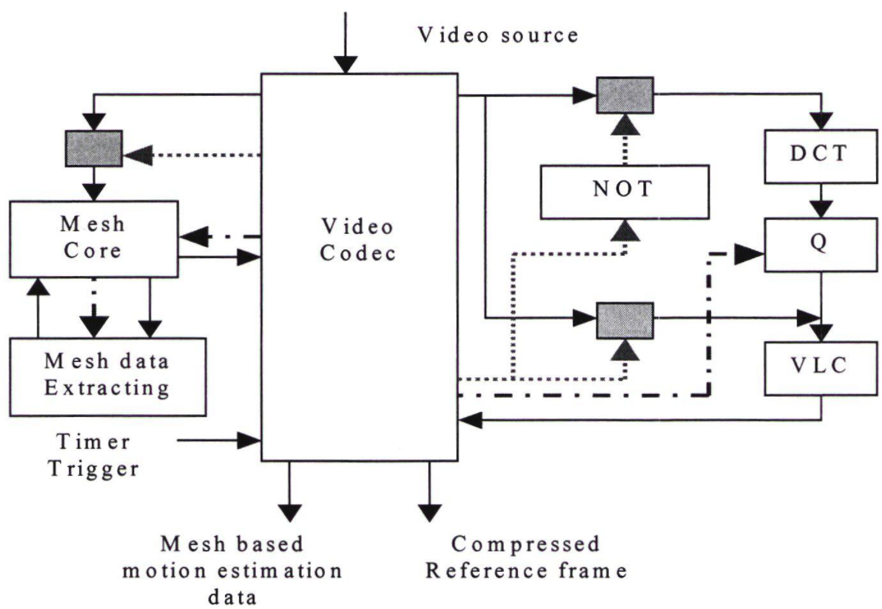



Figure 3.2. Architecture of HAS Video Codec [63,64]

This section briefly describes the architecture of the HAS video codec. As most classic codecs, HAS codec consists of two parts: interframe processing and intraframe processing. Intraframe processing, or known as reference frame processing, mainly works for redundancy reduction of a single frame. Mesh based interframe processing makes contribution to the reduction of temporal redundancy between consecutive frames. Figure 3.2 shows the architecture of HAS video codec.

### 3.1.2 Interframe processing in HAS codec

A reference frame is a still image in wide sense. Hence, most still image compression techniques are applicable in the intraframe processing. HAS codec employs 2D DCT, Quantization, Zigzag scanning, Run Length Encoding and Variable Length Coding in intraframe processing.

#### 3.1.2.1 2D Discrete Cosine Transform

Before 2D DCT coding, the reference frame is divided into N x N blocks. Each block contains 8 x 8 pixels. Then 2D DCT is applied to each block. In this way, the original samples in pixel domain are transformed into frequency domain, where the frequency coefficients are statistically independent. This decorrelation results in the signal energy being redistributed among only a small portion of frequency coefficients. The energy typically lies in low frequency coefficients.

#### 3.1.2.2 Regional Quantization

The quantizer maps many input values into a smaller, finite number of output levels. Compression is archived by quantization, which also introduces distortion. In HAS Codec, regional quantization is used. In mesh based interframe processing, different patch size implies the dynamic of corresponding region. Large patch means this region has less dynamics. In order to achieve further compression in very low bit-rate application, this specific region in reference frame can be more coarsely quantized.

3.1.2.3 Variable Length Coding

After block-based DCT and quantization processing, coefficients are zigzag scanned and run-length encoded. Then the data of N blocks are combined together to generate one reference frame data package, where N is adaptively adjusted according to network condition. After that, variable length coding is applied to approach the entropy of the source. There are two typical variable length coding schemes, i.e., Huffman coding and arithmetic coding. In HAS codec, Huffman coding is used to avoid the cost of complexity. Due to the nature of data streaming, even Huffman coding is bypassed in cases of very worse network condition, when codec adaptive mechanism chooses very small package size.

### 3.1.3 Interframe processing in HAS Codec

3.1.3.1 Hierarchical Adaptive Optimum Mesh (HAOM) algorithm

The HAOM is a technique that constructs a mesh topology using both coarse-to-fine and fine-to-coarse techniques. The mesh topology is generated through a split/merge procedure. This procedure begins with an initial mesh (uniform mesh) at a resolution *r* as shown in Figure 3.3a. According to a split/merge criterion, patches with significant motion components are further divided into four smaller sized patches as shown Figure 3.3b, while the remaining patches that have homogeneous motion parameters are merged together to form larger patches as shown Figure 3.3c. This split/merge procedure is carried out recursively until the split/merge criterion is satisfied or no further division or merging is possible. Figure 3.3d shows the resulting mesh topology after applying the split/merge procedure.

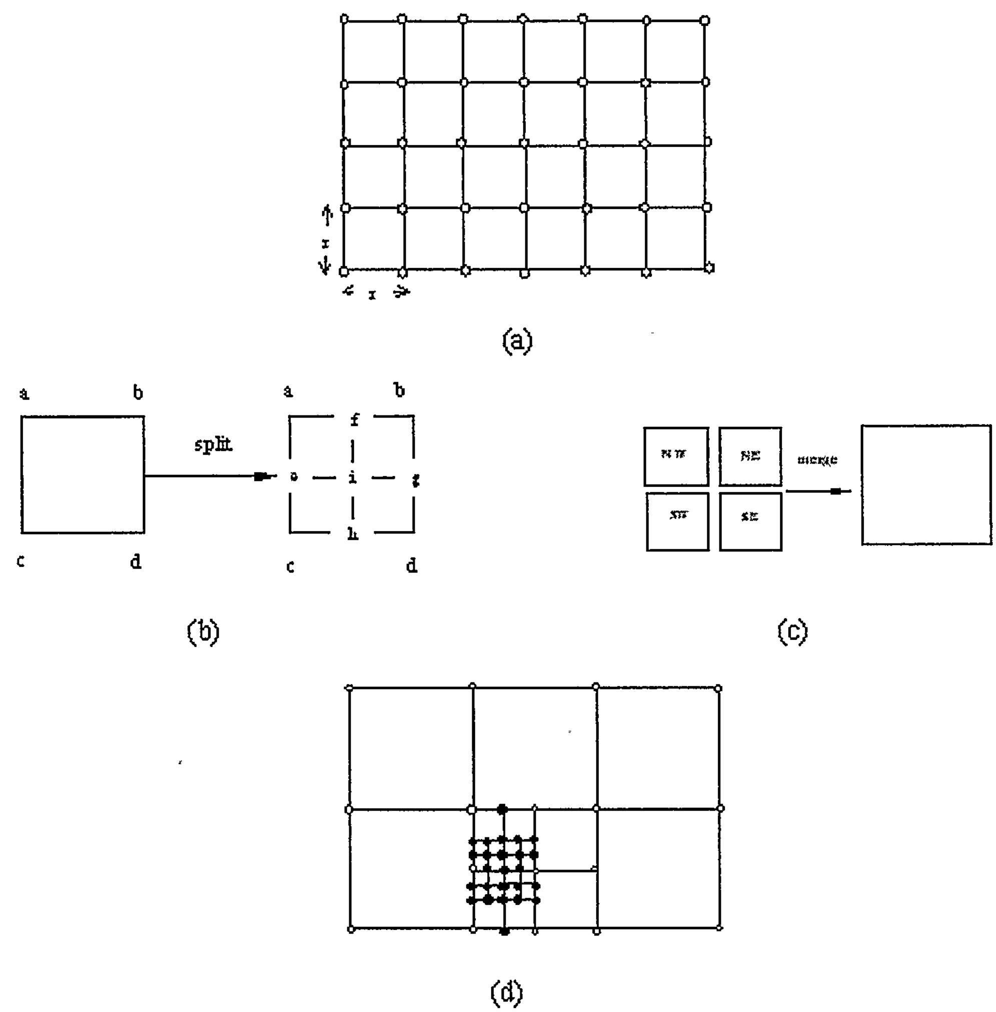


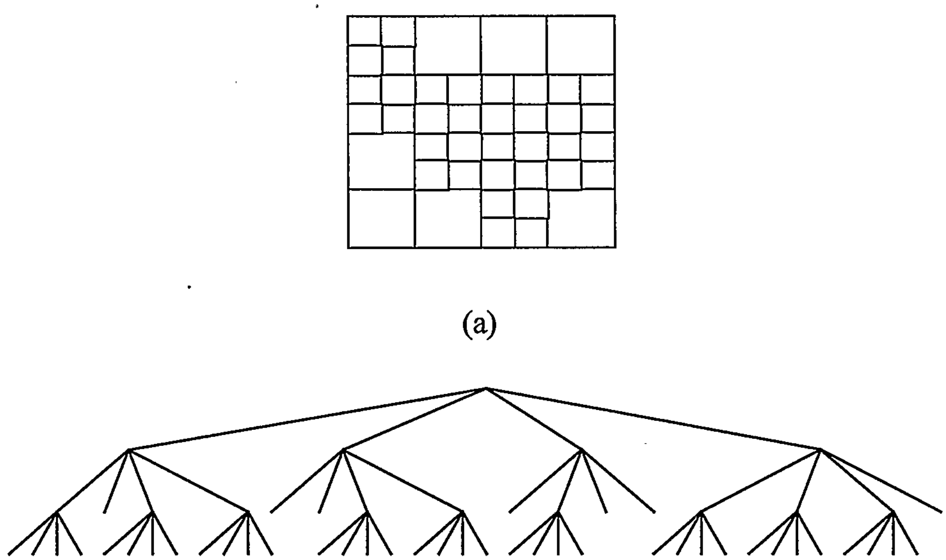

Figure 3.3. Construction of Mesh topology

(a)Initial uniform mesh topology (b) Split operation

(c ) Merge operation (d) Final mesh topology

Motion estimation on each node of HAOM is performed by using the 3 step search (3SS) block-matching technique, which is easy to implement and offers reasonable performance. Meanwhile, Sum of Absolute Differences (SAD) is used as the primary matching criterion for fast computation.

3.1.3.2 Date structure used in HAOM processing



(a)

(b)Figure 3.4. A mesh and correspondent Quadtree

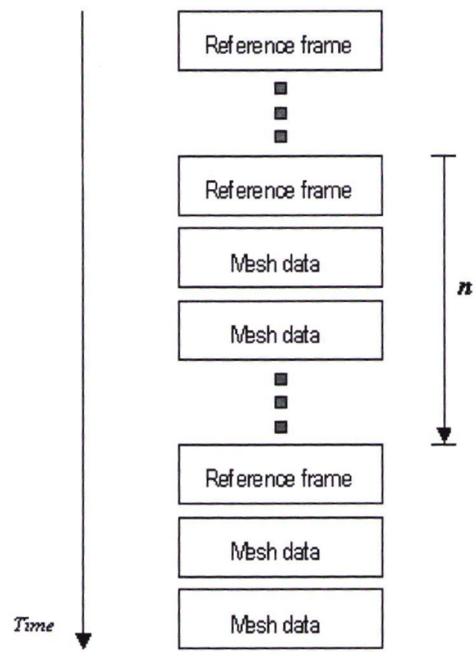
a) A mesh, b) Quadtree structure

To represent the relation between mesh elements efficiently, Quadtree data structure is applied here in HAOM video processing. It is a hierarchical data structure (see Figure 3.4) where a non-homogenous $2^k \times 2^k$ image $X$ is designated as the root of the tree, $k$ being a positive integer defining the pixel resolution of $X$. At the next level, it is decomposed into four equal non-homogenous quadrants (*patch*) which are further divided into sub-quadrants: the children of a particular node signifying the proper, disjoint, collectively exhaustive subsets of the parent, and the leaves representing the pixels.

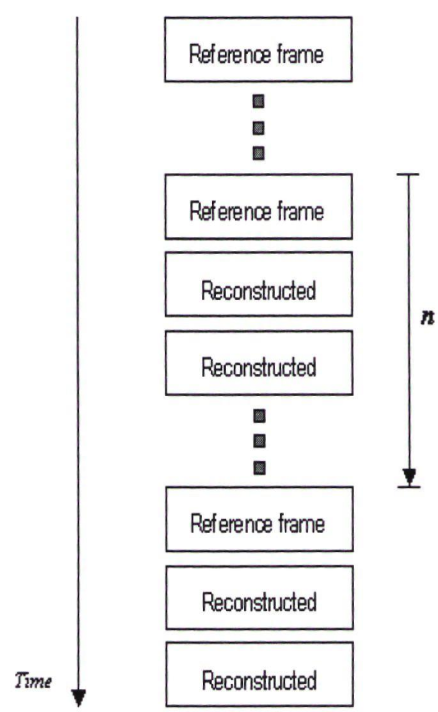
The quadtree could be easily implemented and manipulated by recursive subroutines. For operations, such as splitting, merging and updating, etc, one recursive process could complete the processing of a whole frame or a given region. *Nodes* in a quadtree represent *patches* in an image. To distinguish with the node in a mesh structure, which are stored in a double linked-list, we use the term *patch* to represent the *node* in a quadtree in this thesis.

3.1.3.3 Video sequence after HAOM processing

By employing HAOM algorithm, the logic video sequence will appear on network as shown in Figure 3.5a. Every *n* frames, a reference frame, which is followed by mesh structure and motion vectors, will be sent to the counterpart. Meanwhile, *n* (usually referred to as reference frame interval) is determined by acceptable quality of reconstructed video and bandwidth occupation. Figure 3.5b illustrates the reconstructed video sequence on client-side.




(a)




(b)

Figure 3.5. a) Logic sequence of video data through network

b) Reconstructed video sequence on client side

### 3.1.4 Benefits of using HAS Codec

An advantage of the mesh model over the block model is its ability to represent more general types of motion. At the same time, mesh structure constrains movements of adjacent image patches. Therefore, meshes are well suited to represent mildly deformable but spatially continuous motion fields.

Moreover, HAS Codec shows that it is desirable for low-bit-rate applications analogous to video conferencing or video surveillance, which is featured in less dynamics and simple & continuous scene.

Except the general characters of mesh based techniques, HAS Codec also shows its unique optimum nature because of following reasons:

- According to certain criteria for normal network condition, the mesh structure is always kept in an optimum status, with minimum nodes, by using adaptive algorithm;
- In mesh adaptation, splitting & merging techniques are performed simultaneously, where minimum computing overhead is achieved.

## 3.2 Proposed Hierarchical Packetization Protocol

HAS Codec generates the compressed reference frame data and mesh data in the buffer. To organize these data into the data stream and transmit them over the wireless networks, Hierarchical Packetization Protocol (HPP) is designed and developed.

### 3.2.1 Packetization protocol of reference frame

After processing of HAS Codec, the compressed reference frame is stored in the buffer block by block. Reference frames could seize 50%-90% content of compressed video data. Hence, it is quite important to effectively pack the reference frame with minimum bit-rate overhead.

During the packet transmission, due to the multi-routing and network congestion, wireless networks cannot guarantee all packets are received at the client side in order. Hence, it's necessary to mark the number in each packet. Furthermore, the total number of packets in one reference frame must be provided, because in the adaptive system, the packet length may change in different network conditions. The client needs this information to identify if all packets in one frame have been received. This design can effective avoid the use of synchronization packet and the failure due to lost synchronization packet.

| |
|---|
| **Content (1)** |
| **Total (2)** |
| **Current (2)** |
| **Check (1)** |
| **$B_0$** |
| **$B_1$** |
| **...** |
| **$B_n$** |

Figure 3.6. Reference frame bit-stream format of HAS codec

Figure 3.6 illustrate the HAS codec reference frame bit-stream format. We can see that only minimum extra bytes are attached to each packet. They are:

1. Content (1 byte) identifies packet content.

2. Total (2 bytes) represents total number of packets per reference frame.

3. Current (2 bytes) shows the index of current packet, i.e., this is the $n^{th}$ packet.

4. Check (1 bytes) is the check byte of an exclusive OR (XOR) operation, which is applied on every byte of the current packet except check byte itself. This offers a fast & simple error detection mechanism.

Then, the header is followed by consecutive micro-blocks, where each of them is ended with an EOB (End-Of-Block). In other words, blocks are separated by EOBs. When decoding, each block can be uniquely located by its packet index and block offset within a packet.

### 3.2.2 Packetization protocol of mesh data

After the interframe processing of HAS Codec, the mesh topology and motion vectors are stored in the buffer based on the form of quadtree. To convert the contents of the quadtree into the packages with sequence, two challenges need to be overcome:

1. How to design a compact and efficient algorithm of traversing the quadtree?

2. How to insert minimal information bytes into the package to mark the position of each patch and motion vectors for the reconstruction the quadtree at the receiver side?

The following sections explain how the proposed Packetization protocol solves these problems effectively.

### 3.2.2.1 Quadtree traversal algorithm and mesh data packetizing algorithm

In HAS codec, HAOM algorithms process video hierarchically and adaptively, and the generated mesh data are the necessary information to present a given mesh structure together with corresponding motion vectors of mesh nodes. As discussed in the previous section, each layer of the quadtree represents a certain level of motion compensation. The closer to the root, the coarser the compensation will be. Suppose that mesh data transmission stops at the $n^{th}$ layer in the middle, if all the following packets were corrupted, HAS codec can still work robustly with a reasonable loss of video quality by discarding corrupted packets. Therefore, breadth-first-traversal algorithm better matches the nature of HAOM.

Based on the analysis of quadtree, root-down and breadth-first-traversal method is adopted as the quadtree traversal algorithm. Starting from the root node, this algorithm pushes all the nodes in quadtree into a First-In-First-Out (FIFO) queue in the sequence of top-to-bottom and left-to-right. With this special FIFO queue, the traversal process is conducted layer by layer until all terminal nodes are reached. Figure 3.7 illustrate the route of proposed quadtree traversal approach.

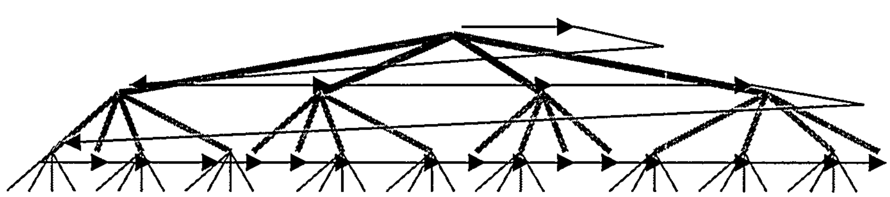


Figure 3.7. The traversal of quadtree

Then, the mesh data packetizing procedure can be described as follows:

1. Set the flag of begin_of_data in the first byte of the buffer.
2. Reserve one byte as the check byte.
3. Assign four bytes for virtual frame width and height.
4. Use two bytes to record the associated reference frame ID.
5. Set the flag of begin_of_tree.
6. Reserve four bytes for motion vectors of the root patch.
7. Set the root patch as current patch and the head patch in the queue.
8. While current patch is not empty
    a. Use the status information of 4 sub-patches of the current patch to generate an eigenbyte.
    b. If the eigenbyte does not equal to zero, that means a splitting operation was performed on current patch and 4 sub-patches exist.
        i. Add all non-leaf sub-patches into the queue, and update the queue end member.
        ii. Reserve 5 bytes for MVs, which are generated by one splitting operation.
    c. Set next candidate patch in the queue as current patch.
    d. If the current patch is in a new layer, put the flag of end_of_layer into the buffer.
9. Set the flags of end_of_mesh and end_of_data into the buffer.

The proposed approach is flexible and general enough to deal with the commonness of quadtree and different mesh structure. Actually, the algorithms can also be applied on part of a quadtree and corresponding part of a mesh. Therefore, it can fit into standards, such as MPEG-4, which employs mesh-processing techniques.

3.2.2.2 Format of Mesh packet

| Header | Eigenbytes | MVs | ... | Eigenbytes |
|---|---|---|---|---|

| Content1 | Layer1 | Total1 | Current1 | Reference ID2 | Check1 |
|---|---|---|---|---|---|

Figure 3.8. Mesh bit-stream format of HAS codec

The topology structure of mesh and motion vectors are combined into packets using the above algorithm. In the mesh packets, packet header describes the attributes of packets, such as the packet index and the reference frame ID that is associated with current mesh data. Following the header, eigenbytes and motion vectors are assembled as the main body of packets. Within a packet, eigenbytes and motion vectors are distinguishable by their high four bits of the byte. Meanwhile, eigenbytes use low four bits of the byte to represent the status of four sub-patches, i.e., whether there is a splitting operation performed or not. Among various eigenbytes, the flag of layer, which is usually at the end of a packet, is inserted after the eigenbyte of the last patch in this layer. Also, check byte and EOM (End-Of-Mesh) flag are assigned. Clients use them to verify packets. If any error were detected, the specific packet would be discarded.

### 3.2.3 Analysis of the proposed hierarchical Packetization protocol

The size of total mesh packets could be formulated as the following equation:

$$S_{\min} \le S \le S_{\max} \tag{3.1}$$

$$S_{\max} = S_h + \sum_{i=0}^{n} (4^i * 5 + 1) + S_e \tag{3.2}$$

$$S_{\min} = S_h + S_e + 6 \tag{3.3}$$

Where,

$S_h$ and $S_e$ are the size of the header part and end part;

$n$ stands for the total layer numbers;

$S_{\max}$ is the size of a full quadtree;

$S_{\min}$ represents a minimum tree only with root node.

It is shown that the total size of mesh packets is exponentially augmented while the layer number increases. It provides a flexible mechanism through which extremely low bit-rate is used for performing mesh-based motion compensation to avoid huge data loss, when available network bandwidth shrinks suddenly. On the other hand, if high bandwidth is available, HPP transmits more layers of motion vectors to improve the video quality outstandingly. Therefore, the hierarchical advantage of mesh processing can be utilized to the furthest. The coarse-to-fine & fine-to-coarse adaptation is realized by loop-control mechanism within HPP.

The mesh abstracting strategies applied here have exceptional advantages over other mesh presentation methods. On the client side, mesh reconstruction can stop at an arbitrary layer according to those received correct mesh data packets. This feature offers a robust & reliable decoding of mesh data packets. Even in the case of network congestion, limited quantity of mesh packets are received. Smooth video playback on the client side is still guaranteed with reasonable QoS decrease. Meanwhile, these strategies also improve codec performance in normal network condition, because packet corruption is inevitable in UDP-based real-time video streaming. Hence, compared with other method describing the mesh structure [26], HPP uses a more elegant and efficient algorithm to represent it.

### 3.3 Hybrid Adaptive Unequal Error Protection

After packetization, reference frame packets and mesh packets are ready to be sent. However, the high compression ratios introduce interdependency of data. Consequently, bit and packet errors due to transmission may render as significant artifacts after the compressed data is encoded. In order to provide reliable video communication, it is necessary to exert some forms of error control. Here we present a proposed hybrid error control solution, which consists of Hybrid Adaptive Unequal Protection Scheme, Adaptive Packet Shuffling scheme and Fast Packet Length Adjustment Scheme. Figure 3.9 illustrates the structure of this solution. In this section, we present the proposed Hybrid Adaptive Unequal Protection Scheme.

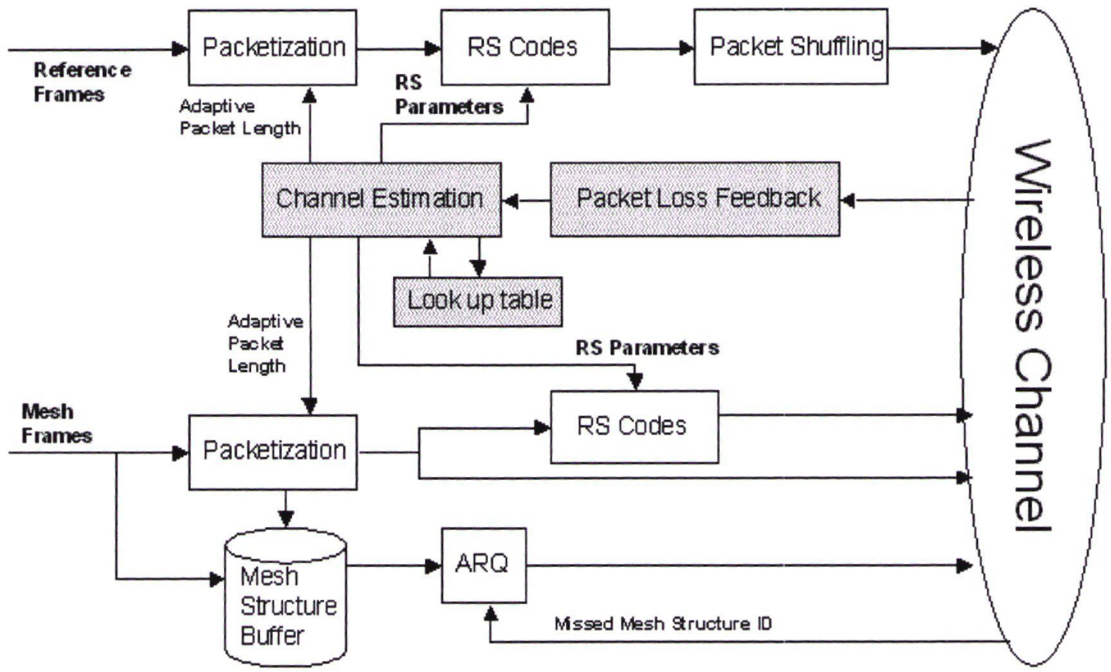



Figure 3.9. Architecture of the proposed error control scheme

### 3.3.1 Background and Motivation

Traditionally, two classes of communication protocols are used in practice to protect data over packet networks: synchronous and asynchronous. Asynchronous communication protocols, such as ARQ, are reliable but have unbounded delay. During the transmission, for each intact data packet received in the forward channel, the receiver sends back an acknowledgment. Thus, ARQ requires a two-way communication channel to be present. While this model works very well for data communication, it is not suitable for multimedia streams with hard latency constraints. The maximum delay of the ARQ mechanism is unbounded, and in multimedia applications it is usually preferable and, in the case of live streaming, necessary to interpolate late-arriving or missing information rather than insert a delay in the stream playback.

In synchronous protocols, the data are transmitted with a bounded delay. To provide for some measure of reliability, FEC coding is employed. FEC codes are applied to a group of source data packets. The FEC codes are designed to protect data against channel erasures by introducing parity packets. No feedback channel is required. If the number of erased packets is less than the decoding threshold for the FEC code, the original data can be recovered perfectly. However, FEC techniques cannot guarantee that the receiver receives all the packets without error.

Note that existing streaming media servers and clients are based on a partially synchronous version of the ARQ protocol [29]. These applications maintain a record of the approximate round-trip time for a packet and its acknowledgment, and use this information to determine at the server if a packet is likely to arrive at the destination before its deadline. In this way, the unbounded delay of ARQ protocols can be avoided.

However, even with this change, the ARQ-based protocols still require a small overall packet loss rate and low round-trip latency to achieve an acceptably small probability of stream transmission failure.

To our knowledge, all existing error protection schemes are not suitable for protecting the video data in our system. We need to design a new scheme to fit the characters of HAS codec.

### 3.3.2 Proposed Hybrid Error Protection Scheme

From the analysis of video packets in Section 3.2, we know that any distortion in reference frame will last until the next reference frame is received. By contrast, the loss of motion vector has less effect on the video quality. Hence we used different error protection scheme in these two type packets. Considering the real time system, too much retransmission may increase the system delay significantly. So ARQ is not suitable for protecting reference frame packets. In this scheme, we employed the Reed Solomon Codes [67] to protect the reference frame.

In mesh packets, mesh structure data represent the mesh topology. Normally for speed the process at the both sides, the same mesh structure will be repeatedly used for at least two frames. Hence the loss of mesh structure may result in missing several frames. Considering the mesh structure has only limit bytes and need be retransmitted just one time while encountering error [63]. We use ARQ to guarantee that the client must receive the mesh structure. As we discussed in Section 3.2.2.2, mesh structure is part of mesh packet. As showed in Figure 3.1, we designed another mechanism to extract mesh structure before Packetization. Mesh structure buffer keeps five recent structures and sends them to the wireless channel when it gets the resending request.

### 3.3.3 Reed Solomon (RS) Codes

RS codes are a systematic linear block code, which is one type of FEC [67]. It's a block code because the code is put together by splitting the original message into fixed length blocks. Each block is further subdivided into m-bit symbols. Each symbol is of a fixed width, usually 3 to 8 bits wide. An RS code is partially specified as an RS (n, k) with m-bit symbols. For instance the RS (204,188) uses 8-bit symbols. The *n* refers to the number of encoded symbols in a block, whilst *k* refers to the number of original message symbols. The difference *n-k* (usually called *2t*) is the number of parity symbols that have been appended to make the encoded block.

An RS decoder can correct up to *(n-k)/2* or *t* symbols, i.e. even if any *t* symbols can be corrupted in any way, and the original symbols can be recovered. Thus the video codec splits the message into blocks 188 symbols long. The 16 parity symbols (2t = 204-188 = 16) are then appended to produce the full 204 symbol long code. Up to 8 (t = 16/2) symbol errors can then be corrected. The power of Reed Solomon codes lies in being able to just as easily correct a corrupted symbol with a single bit error as it can a symbol with all its bits in error. This makes RS codes particularly suitable for correcting burst errors. Usually the encoded data are transmitted or stored as a sequence of bits. Often the RS encoded block is further encoded in a convolutional code to try and cope with both burst and random errors.

RS codes are based on finite fields, often called Galois fields. An RS code with 8 bit symbols will use a Galois field GF $(2^8)$, consisting of 256 symbols. Thus every possible 8 bit value is in the field. The order in which the symbols appear depends on the generator polynomial. This polynomial is used in a simple iterative algorithm to generate each element of the field. Different polynomials will generate different fields. For instance, the generator polynomial could be p(x) = $1 + x^2 + x^3 + x^4 + x^8$. This can be given the shorthand 285, from the binary value of the coefficients 100011101. From this the $n^{th}$ element of the field can be constructed by raising element 0 to the power n.

### 3.3.3.1 The Reed Solomon Encoder

The encoder is the easy part. Since the code is systematic, the whole of the block can be read into the encoder, and then output the other side without alteration. Once the $k^{th}$ data symbol has been read in, the parity symbol calculation is finished, and the parity symbols can be output to give the full $n$ symbols. The encoder acts to divide the polynomial represented by the $k$ message symbols $d(x)$ by the RS generator polynomial $g(x)$. This generator polynomial is not the same as the Galois Field generator polynomial, but is derived from it.

$$x(n-k).d(x)/g(x) = q(x) + r(x)/g(x) \tag{3.4}$$

The term $x(n-k)$ is a constant power of $x$, which is simply a shift upwards n-k places of all the polynomial coefficients in $d(x)$. The remainder after the division $r(x)$ becomes the parity. By concatenating the parity symbols on to the end of the $k$ message symbols, an $n$ coefficient polynomial is created which is exactly divisible by $g(x)$.

The encoder is a $2t$ tap shift register, where each register is m bits wide. The multiplier coefficients $g(0)$ to $g(2t-1)$ are coefficients of the RS generator polynomial. The coefficients are fixed, which can be used to simplify the multipliers if required. The only hard bit is working out the coefficients.

At the beginning of a block all the registers are set to zero. From then on, at each clock cycle the symbol in each register is added to the product of the feedback symbol and the fixed coefficient for that tap, and passed on to the next register. The symbol in the last register becomes the feedback value on the next cycle. When all *n* input symbols have been read in, the parity symbols are sitting in the register, and it just remains to shift them out one by one.

#### 3.3.3.2 The Reed Solomon Decoder

Decoding is a far harder task than encoding. The first step in decoding the received symbol is to determine the data syndrome. Here the input received symbols are divided by the generator polynomial. The result should be zero (the parity was placed there to ensure that code is exactly divisible by the generator polynomial). If there is a remainder, then there are errors. The remainder is called the syndrome.

The second step is to find the error polynomial lambda. This requires solving *2t* simultaneous equations, one for each syndrome. The *2t* syndromes form a simultaneous equation with t unknowns. The unknowns are the locations of the errors. In general there are many possible solutions to the set of equations, but we assume that the one with the least number of errors is the correct one. This assumption is the reason that more than *t* errors can actually cause the decoder to corrupt the received signal further (if allowed to). If more than *t* errors occur, then there will exist a possible solution to the equations with less than *t* errors.

The process of solving the simultaneous equations is usually split into two stages. First, an error location polynomial is found. This polynomial has roots which give the error locations. Then the roots of the error polynomial are found. Once the error polynomial lambda is known, its roots define where the errors are in the received symbol block. The most commonly used algorithm for this is the Chien search. This is a brute force and ignorance method, or more politely, an exhaustive search. All $2^m$ possible symbols are substituted into the error polynomial, one by one, and the polynomial evaluated. If the result comes to zero, we have a root.

At this stage, it could be known where the errors are, but not what they are. The next step is to use the syndromes and the error polynomial roots to derive the error values. This is usually done using the Forney algorithm. The algorithm is an efficient way of performing a matrix inversion. The algorithm works in two stages. First the error evaluator polynomial omega is calculated. This is done by convolving the syndromes with the error polynomial lambda. Omega is then calculated at each zero location, and divided by the derivative of lambda. Each calculation gives the error symbol at the corresponding location. If a bit is set in the error symbol, then the corresponding bit in the received symbol is in error, and must be inverted. All that remains is to correct the received symbols. The symbols are read again from an intermediate store, and at each error location the received symbols XOR'ed with the error symbol. Usually the parity symbols are stripped off.

### 3.3.4 Adaptive Unequal Error Protection

To work in different bandwidths, parameters of the RS Codes are dynamically adjusted along with the feedback from the wireless channel. In some bandwidth situation with high interference, the high redundancy bits will be inserted. In extreme low bit channel, we have to protect one reference frame per n frames (interval n = 2,3.x) to guarantee the packets throughput rate.

Above are traditional adjustment schemes. In this thesis, we proposed a novel adjustment method, which utilizes the advantage of mesh based codec. As discussed in section 3.1, the essence of HAOM mesh structure is using fine mesh to represent more movement parts and using coarse mesh to stand for still parts.

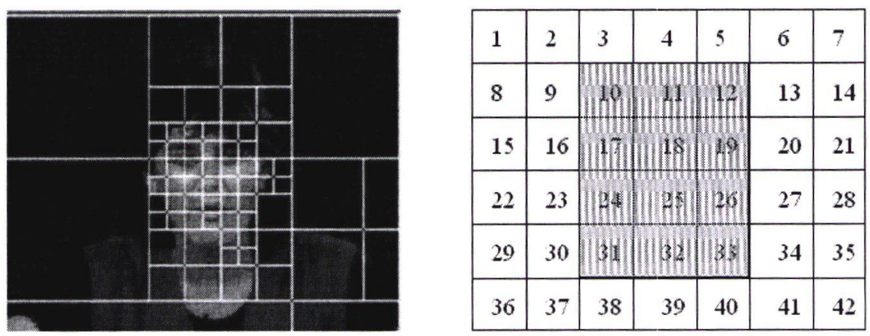



Figure 3.10 Adaptive FEC protection with selected blocks

In practice, because of the error caused by searching of block matching, the moved part normally generates more distorts. Hence it much needs reference frame to refresh / correct itself than the still part need. Based on this fact, we design a link between mesh structure and reference blocks. For example, in Figure 3.10, when the speaker moved her face, it caused more movement of blocks at that position. The mesh structure clearly captured this status. System recorded the mesh structure and marked corresponding blocks with the number of layers at this position. In the selective mode, blocks are selected according to the marked number. One threshold was used to control the number of selected blocks. In the extreme low bit rates network, the threshold could be set to very small to pick the most active blocks. Then the RS codes just need protect the selected blocks, which significantly reduce the redundancy bits.

## 3.4 Packets shuffling

As discussed in the last section, the reference frame is protected by FEC scheme. But FEC techniques cannot guarantee that the receiver receives all the packets without error. For example, if the number of error packet exceeds the half of parity packets, FEC cannot correct the errors. Furthermore, in the low bit rate wireless network, it's impossible to add too much redundancy to reduce the error. Under these circumstances, we need to supplement with other schemes to control the effect caused by burst errors.

### 3.4.1 Background and Motivation

According to the study of human perception, some loss of quality can be tolerated. This feature can be utilized in most video systems. It could be easy to understand that consecutive packets loss may give audiences a strong feeling of the distortion of the image.

This phenomenon motives us to design a scheme to lower the influence of consecutive error. To change continuous errors to average errors could be a workable solution. Another advantage of this scheme is that it could avoid FEC's working scale to be overun.

Figure 3.11 illustrate a sample of how the order of packets affects consecutive loss. For instance, we have a sequence of 15 packets numbered from 1 to 15. Two network burst errors caused the loss of packets 4 to 6 and 11 to 13.

If this sequence of packets is shuffled before transmission, the result is consecutive packets become far apart in the sequence, and then the consecutive loss can be reduced significantly.

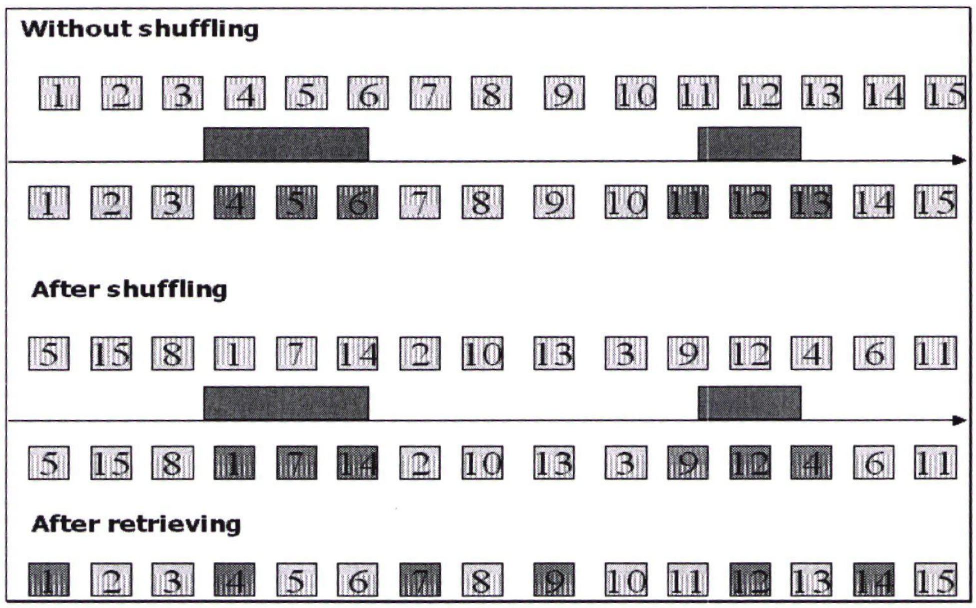



Figure 3.11. Example of how the order of packets affects consecutive loss

The above example shows even random shuffling could effectively average the continue errors. But a question is raised: What variation of the input sequence minimizes the continue error for a give network loss? In the following sections, a shuffling algorithm is proposed to solve this problem.

### 3.4.2 Problem Statement

The objective of this problem is to reduce the bursty error to a perceptually acceptable level (by spreading it out). There are following parameters to bound the problem. (1) The sender's buffer size *m*, which is determined by the sender's operating environment and its current status. (2) The upper bound on the size of a bursty loss in the communication channel *p*. (3) The user's maximum acceptable number of continue errors *k*. What we need to get is a permutation function *f* on S , where S = { 1, 2,…,m }, which decides the order in which a set of consecutive packets must be sent.

Obviously there is a typical trade off between buffer size and continue errors. The greater *m* is, the less number we can support but also the greater memory requirement and initial delay time.

Here we need notice one point. During the processing, each packet is consists of several blocks. This allows us to consider every packet to be equally important; thus, we can permute the frames in any way we would like to.

### 3.4.3 Packet Shuffling Scheme

In this section, we use a pure mathematical method to solve this problem. Now we are given positive integers *m* and *p*. Let $S_m$ denotes the set of all permutations on [m] = $\{1,2,3,\ldots,m\}$. For any permutation $x \in S_m$, the sets $W_i^x$ given by

$$W_i^x = \{x_{i,} x_{i+1}, x_{i+p-1}\}, 1 \le i \le m \tag{3.4}$$

are called the sliding windows of size *p (of x),* where the indices are calculated modulo *m*, then plus 1. When $1 \le i \le m-p+1$, $W_i^x = \{x_{i,} x_{i+1}, x_{i+p-1}\}$ , and when

$$m-p+1 < i \le m,\ W_i^x = \{x_{i,} x_{i+1}, x_m, x_1, \ldots, x_{i+p-m-1}\}. \tag{3.5}$$

For any pair of integers *k* and *l* such that $1 \le k < l \le m$, let [*k,l*] denote the set $\{k, k+1, \ldots l\}$. Let $\{c_i^x\}_1^m$ be the sequence of integers defined as follows.

$$c_i^x = \begin{cases} \max\{|[k,l]|\},[k,l] \subseteq W_i^x \\ if 1 \le i \le m-p+1 \\ \max\{|[k,m]|+|[1,l]|\},[k,m] \subseteq \{x_1,\ldots x_m\},[1,l] \subseteq \{x_1,\ldots x_{p+i-m-1}\} \\ if m-p+1 < i \le m \end{cases} \tag{3.6}$$

Let $C^x = \max\ \{c_i^x \mid 1 \le i \le m\}$. Then $k_0$ is defined to be

$$k_0 = \min\{C^x, x \in S_m\} \tag{3.7}$$

Our objective is to find $k_0$ as a function of $m$ and $p$. Moveover, we also wish to specify a permutation $x$ so that $C^x = k_0$.

Informally, when $1 \le i \le m-p+1$, $C_i^x$ is the maximum number of continuous integers in $W_i^x$. While if $m-p+1 < i \le m$, $C_i^x$ is the sum of two quantities $a$ and $b$, where $a$ is the length of the longest consecutive integer sequence in $\{x_i,\ldots x_m\}$ which ends in $m$, and $b$ is the length of the longest consecutive integer sequence in $\{x_i, x_{i+1}, x_{i+p-m-1}\}$ which starts at 1. The reason for this is that suppose we apply our permutation to two adjacent buffers of size $m$, we would like our permutation to also deal with the case where the network loss burst occurs across these two buffers.

The value of $k_0$ and permutation $x$ depends significantly on the relationship between $p$ and $m$. By using the following algorithm, ($m$,$p$) could produce the appropriate permutation which supports the best $k_0$ given $p$ and $m$.

The following table lists the packet shuffling algorithm.

**Packet Shuffling Algorithm**

Begin
If $p \le 0$ or $p \ge m$ then
Output the identity permutation
End if
If $p \le \frac{m}{2}$ then

$$M \leftarrow \{j, p \le j \le \frac{m}{2} \wedge \gcd(m, j) = 1\}$$

if $M \ne 0$ then
$p' \leftarrow \min\{j, j \in M\}$
for i=1 to $m$ do
$x(i) \leftarrow ((i-1)p' \bmod m) + 1$
end for
else
$p' = \frac{m}{2}$
for i=1 to $m$ do
$x(i) \leftarrow p' * (i \bmod 2) + \left\lceil \frac{i}{2} \right\rceil$
end for
end if
else
$q = m - p$
$r = \left\lfloor \frac{p}{q+1} \right\rfloor$
$t = \left\lfloor \frac{m}{r+2} \right\rfloor$
$t' \leftarrow m \bmod (r+2)$
if $t' = r + 1$ then
for i=1 to t+1 do
$a_i = 1 + (i-1) * (r+2)$
$b_i = (r+1) + (i-1) * (r+2)$
end for
$C \leftarrow \{1,2,...m\} - \{a_i\} - \{b_i\}$
for i=1 to t+1 do
$x(a_{t+2-i}) \leftarrow i$
end for

```
        for i= t+2 to m-(t+1) do
            x(c_{m-i-t}) ← i
        end for
        for i=m-t to m do
            x(b_{m-i+1}) ← i
        end for
    else
        for i=1 to t do
            a_i = t' + 1 + (i-1)*(r+2)
            b_i = i*(r+2)
        end for
        C ← {1,2,...m} - {a_i} - {b_i}
        for i=1 to t do
            x(a_{t+1-i}) ← i
        end for
        for i= t+1 to m-t do
            x(c_{m-i-t+1}) ← i
        end for
        for i=m-t+1 to m do
            x(b_{m-i+1}) ← i
        end for
    end if
end if
End
```

### 3.4.4 Analysis of packet shuffling scheme

Two parameters need to be considered to apply this scheme. One is buffer size, and the other is delay factor. For the buffer, both server and client sides need it. The size of the buffer $m$ is depended on the requirement of the maximum acceptable number of continue errors $k$.

Delay factors typically include startup delay and individual timing drifts. This scheme does not introduce new drifts and the only delay it induces is the startup delay, which is small enough for most practical purposes, except in the case of highly interactive applications.

## 3.5 Dynamic Packet Length Adjustment

### 3.5.1 Background and motivation

Proposed video streaming platform is based on the TCP/UDP protocol. For the transmission of time-sensitive video packet, UDP protocol is employed. TCP with acknowledge mechanism is used in transmitting the non real time information, such as the feedback from receiver to the sender. During the transmission, the physical layer of the intermediate network equipment detects bit errors in a packet by the use of checksums. Corrupted packets are useless for the end hosts because TCP/IP protocol has no mechanism to recover from individual bit errors. Thus, the physical layer sensibly drops a packet. In TCP link, the sender will be informed to retransmit the undelivered packet. In UDP link, the packet will not be resent.

In wireless environments, packets transmitted over the air may have bit errors introduced. The bit error rate of a typical wireless network maybe as high as 1 bit error in $10^{-5}$ bits, which is about 100 times greater than a wired network ($10^{-7}$). If we assume that the bit error rate(BER) is constant over time or over a bit stream, then the packet error rate (PER) grows exponentially when the sender increase the packet size [20]:

$$PER \leq 1-(1-BER)^{L+H} \tag{3.8}$$

where L+H is the length of packet in bits.

Of course, an invariant bit error model over time or bit stream may not necessarily reflect the real world. However, this model can reveal the relationship between PER and BER. For this study, we assume that bit error rate is constant over time, so the distance between two consecutive bit errors is exponentially distributed over the bit stream. This model is the prevalent one in the literature.

Obviously, when a packet size employed is too large, there is an increased possibility for encountering bit error. But we cannot use the smallest packet size to improve the packet error rate. Too small packet size is also not efficient because of the fixed overhead required per packet.

Most designers of the traditional video streaming system did not concern this fact. Even when some of them consider this problem, they typically choose a packet size that would work with the worst acceptable bit-error-rate. Unfortunately this packet size makes inefficient use of the channel when the bit-error-rate is much lower.

In this thesis, we propose the use of dynamically packet length adjustment (DPLA) to change packet size adaptively. This approach can effectively control the packet error rate (PER), which results in better performance in wireless network.

### 3.5.2 The Proposed Dynamic Packet Length Adjustment Approach

To design an efficient DPLA, there existed three problems to solve.

1. How to probe the bit error rate?
2. How to select the best packet length according to the estimated bit error rate?
3. Do the codec and packetizer adjust the packet length easily and correspondingly?

In following sections, we will describe the proposed approach based on the solutions of these three problems.

3.5.2.1 Feedback of packet loss numbers

In the system, we design a feedback channel between the sender and the receiver. The receiver informs the received packet numbers per second. At the sender side, system can get the packet loss rate by comparing the sent packet numbers with the received numbers.

The relationship between the historic data of packet loss numbers and the probability of channel bit error rate will be discussed in following section.

3.5.2.2 Estimation of the optimal packet length

Based on packet loss history, the packet size is chosen such that the conditional efficiency of the protocol is maximized. For deriving conditional efficiency expression, we assume that only packets containing errors are lost. For a given channel error rate p, the efficiency of a protocol that uses packets of size k is given by [3.9], as referred in [66]:

$$EFF = \left( \frac{k}{k + h} \right) \frac{1}{(1 - p)^{-(k+h)}} \tag{3.9}$$

where $k$ is the number of information bits, $h$ is the number of header bits in the packet and $p$ is the channel bit-error-rate.

The first term of the above expression represents the ratio of information bits to total bits in a packet, while the second term represents the average number of transmission attempts per packet.

The expected efficiency of the protocol given $R$, which is the number of the lost packets out of the last $M$ packet transmissions, can be expressed by averaging the above expression over all possible values of $p$ and using the conditional distribution of p given $R$ (assuming that $p$ is constant over the period of interest). The resulting expression is given by

$$EFF_R(k) = \int_p \frac{k(1-p)^{k+h}}{k+h} P(p \mid R) \tag{3.10}$$

The conditional probability of p given R, P(p|R), can be expressed as follows:

$$P[p \mid R] = \frac{P[p,R]}{P[R]} = \frac{P[R \mid p]P[p]}{P[R]} \tag{3.11}$$

We need know the prior distribution of $p$ to solve the above condition. In the absence of a prior, we assume a uniform prior; that is P[p] = 1.Then we get

$$P[R] = \int_p P[R \mid p]P[p] = \int_p P[R \mid p] \tag{3.12}$$

Given $p$, the probability that $R$ packets contain errors and therefore require retransmission is the probability that $R$ out of $M$ packets are in error. Since packet errors are independent from packet to packet, this probability can be expressed according to the binomial distribution with parameter E. E is the probability that a packet contains errors and is given by

$$E = 1-(1-p)^{\hat{k}+h} \tag{3.13}$$

where $\hat{k}$ is the packet size used in the previous $M$ transmissions.

Combining equations (3.12) to (3.13) we get the following expression for the expected efficiency of the protocol for a given value of R.

$$EFF_R(k) = \int_p \left[ \frac{k(1-p)^{k+h}}{(k+h)} \times \frac{\binom{M}{R} E^R (1-E)^{M-R}}{\int_p \binom{M}{R} E^R (1-E)^{M-R}} \right] \tag{3.14}$$

It is now possible to choose the value of $k$, the block size to be used in future transmissions, so that the efficiency of the protocol is maximized.

An optimal value for $k$ can now be found using numerical search algorithms. To reduce the search complexity, a restricted search using select values for $k$ can be performed. Suppose we just have 10 options to be selected, it's quite fast to find the best $k$ with better impact on the performance of the protocol.

3.5.2.3 Source coding and packetization with packet length adjustment

An important assumption in designing DPLA is the codec must have the ability of codec the different size of packet. In the HAS codec, as discussed before, the reference frame compressor can change the packet size by combining different number of blocks. For mesh packet, we can control the number of layers to adjust the size. In practice, the size of mesh packet has less options then reference frame.

The proposed dynamic packet length adjustment scheme gave answers to three questions we mentioned at the beginning. Experimental results also shows that this scheme works well in the video streaming platform.

# CHAPTER 4

# SYSTEM PROTOTYPE AND EXPERIMENTAL RESULTS

## 4.1 Prototyped Video Streaming Platform

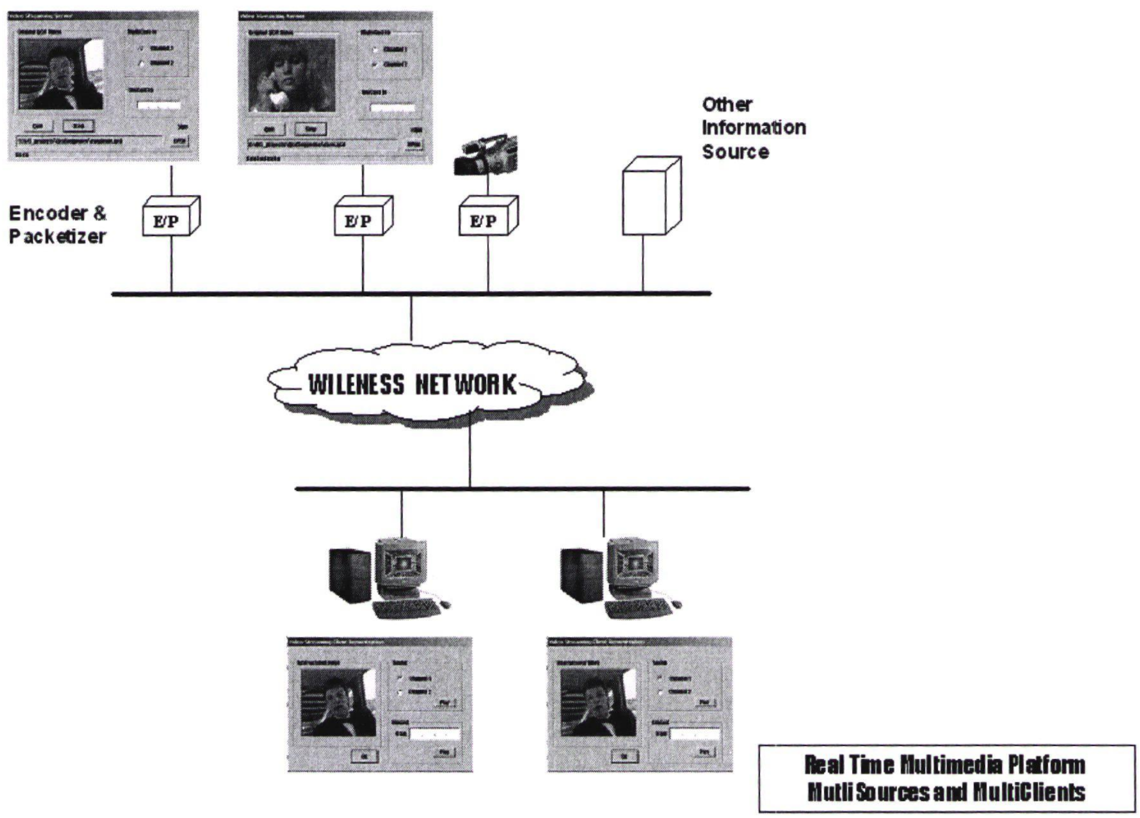



Figure 4.1. Proposed Video Streaming Platform Over Wireless Network

The system consists of several video servers and video clients as showed in Figure 4.1. Video server can be described more precisely as video source, video codec, packetizer and transmitter. In the prototype system, we employed the HAS codec. In the implementation of packetizer and transmitter, Hierarchical Packetization Protocol, Unequal Error Protection, Packet Shuffling and Dynamic Packet Length Adjustment are designed and applied. Video clients have similar modules as servers, but all modules have the reverse functions, including decoder, unpacketizer and receiver.

The system is called as "Platform", because it can support different video sources. In the implemented prototype system, live video from real time camera and stored video files (YUV video format) can be compressed and sent to the network. A pre-processing module is setup before the codec processing the video. Live video or video file are converted into the buffer frame by frame, whatever format or video interface they have. This mechanism makes the system expandable to unicast or multicast other types of media, such as pure images. As a video streaming platform, the system could be applied in many fields, such as distance learning, video surveillance and video conferencing system.

### 4.1.1 Software and Hardware Environment and software architecture

The system is prototyped using Visual C++ 6.0 and Winsock 2.0 on Windows 2000/XP platform. The PC used in the test is configured with P4 1.2G CPU and 256 RAM. Client software could be run on the machines with Pentium MMX 166MHz CPU or up, which has been tested successfully.

The software architecture is shown in Figure 4.2. Two transmission threads and one main thread collaborate to construct the streaming platform. The main thread captures user control information and executes video processing and packetization. In the meantime, transmission thread streams the processed packets and receives the feedback from the client side.

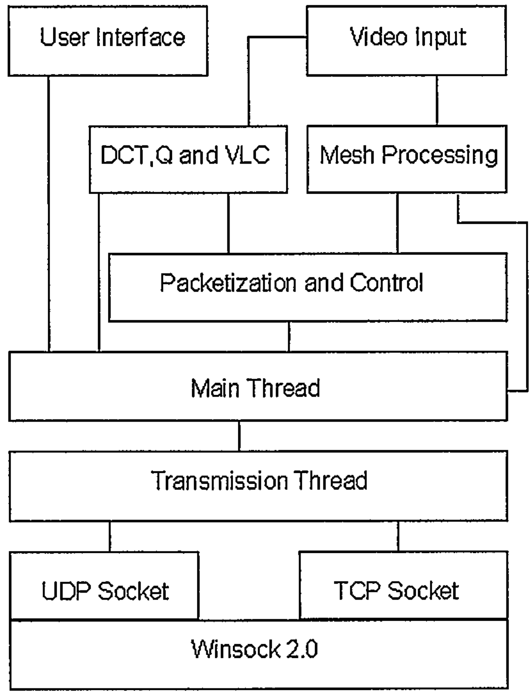



Figure 4.2 Software Architecture of Prototyped Video Streaming Platform

### 4.1.2 User Interface and System Functions

Figure 4.3 show the user interface of video server. User can select the streaming mode: unicast or multicast. Unicast refers to networking in which computers establish two-way, point-to-point connections. When streaming multimedia over a network, the advantage to unicast is that the client computer can communicate with the server supplying the multimedia stream. The disadvantage of unicast is that each client that connects to the server receives a separate stream, which rapidly uses up network bandwidth.

Multicast refers to the networking technique in which one server sends a single copy of the data over the network and many computers receive that data. Unlike a broadcast, routers can control where a multicast travels on the network. When streaming multimedia over the network, the advantage to multicasting is that only a single copy of the data is sent across the network, which preserves network bandwidth. The disadvantage to multicasting is that it is connectionless; clients have no control over the streams they receive. To use IP multicast on a network, the network routers must support the IP Multicast protocol.

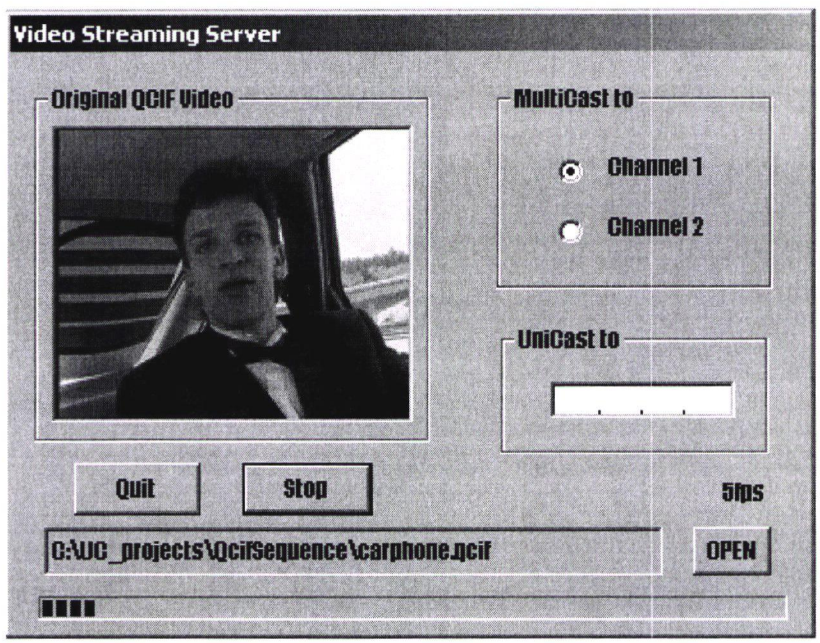



Figure 4.3. User Interface of Video Streaming Server

In the unicast mode, the server need to know the destination IP address and port number. Then all packets will be attached with the destination information and sent to the network. In the multicast mode, the destination IP address is one of special multicast address, which is in range 224.0.0.0 through 239.255.255.255. Proposed video streaming platform uses IP address from 225.1.1.1. This type of IP address stands for different multicast group, all clients joining this group can receive the video stream labeled this address. The first IP group is regarded as channel 1. By analogy, 225.1.1.2 represents channel 2, etc.

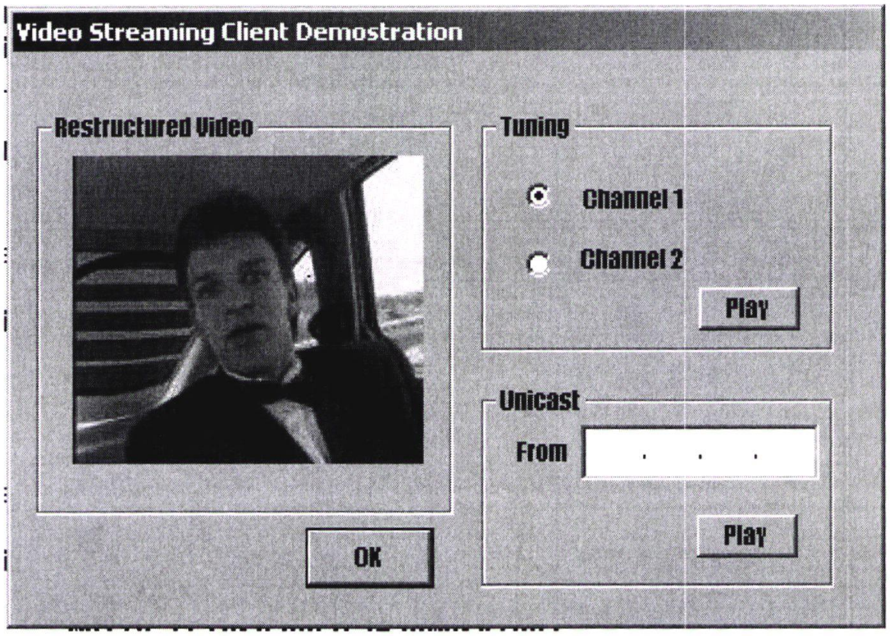



Figure 4.4 User Interface of Video Streaming Client

Figure 4.4 is the user interface of a video streaming client. As same as the server, clients can select the streaming mode: unicast receiving or multicast receiving. In unicast mode, the client needs to specify the server address to get the stream. In multicast mode, the client needs to point out which channel he wants to watch. On the other word, he needs to specify the multicast group that he joins to take the video stream.

## 4.2 Video quality and bandwidth analysis

Generally, video quality varies with available bandwidth. The connection between these two aspects can be used to judge the proposed video streaming system's efficiency and performance.

In the Figure 4.5, the test sequence Miss American is used as the video source. With this 10 fps QCIF (176*144, YUV) sequence, an average bandwidth of 117.6Kbps is needed, compared to raw QCIF's 3041.28Kbps. Effect of compression is notable. An average PSNR of 32.37dB (the minimum PSNR is 28.62dB) is achieved. Curves of PSNR and bandwidth are relatively constant, which is a superior feature of HAS codec.

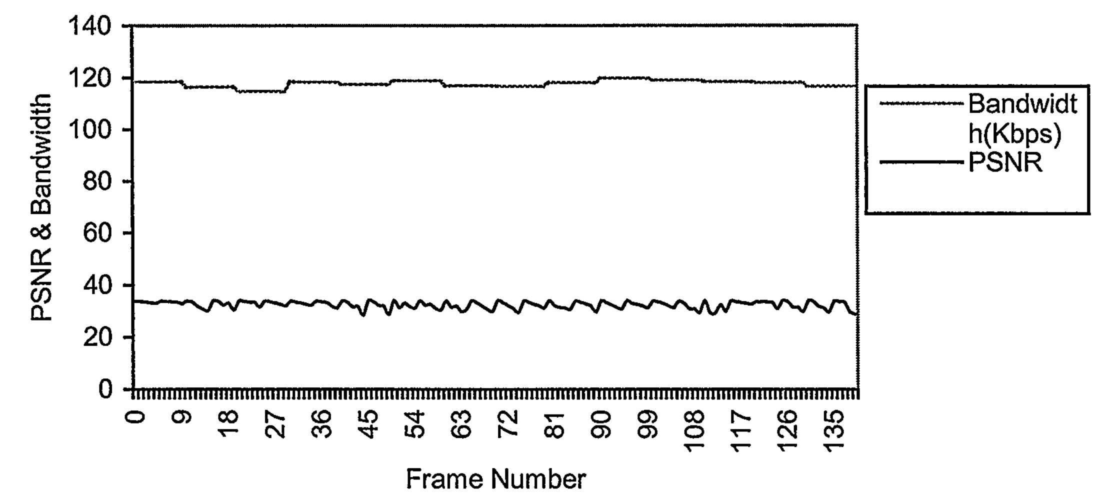
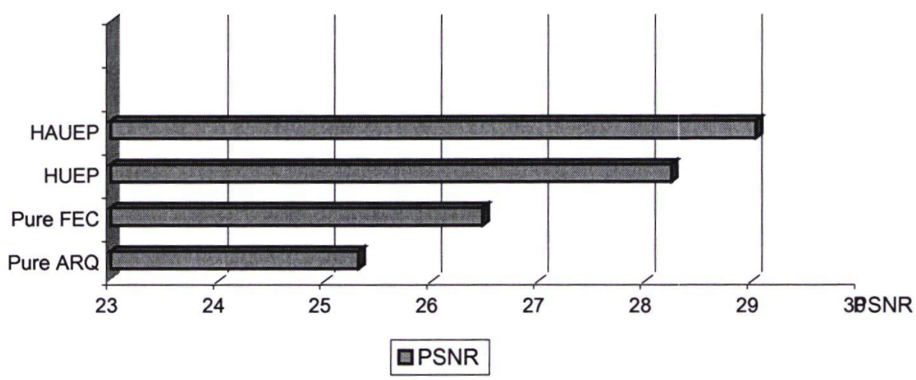


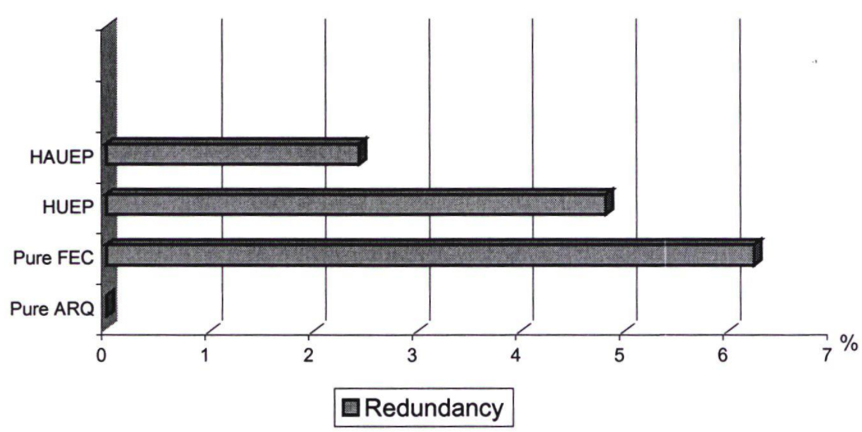

Figure 4.5. Video Quality and Bandwidth Analysis

(Test sequence: Miss American)

## 4.3 Performance analysis of HAUEP

To compare the proposed HAUEP with other general error protection methods, we designed this experiment. In the experiment, four typical wireless channels are simulated, including narrow bandwidth with low and high error rates, high bandwidth with low and high error rates. In each situation, we tested the video streaming with pure FEC protection, pure ARQ, HUEP and HAUEP. We used Miss American as the test sequence, which has 150 frames in YUV QCIF format. The average PSNR, redundancy and delay are recorded.



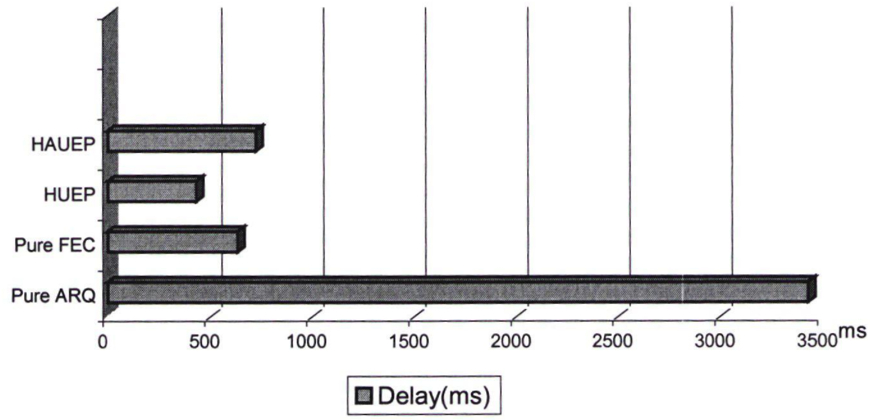
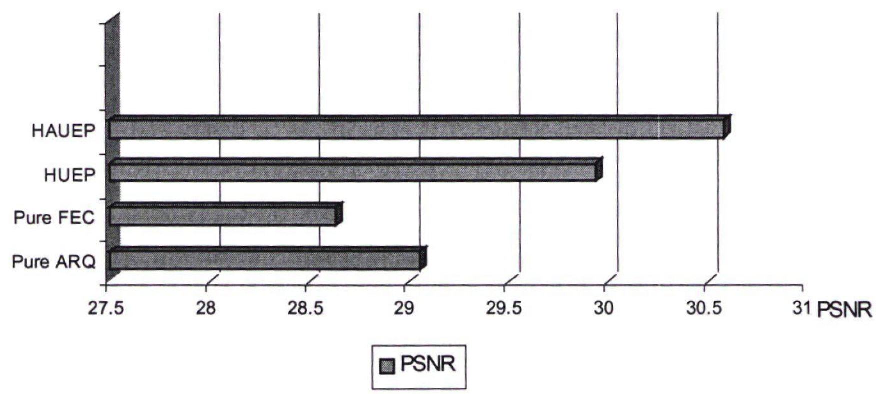
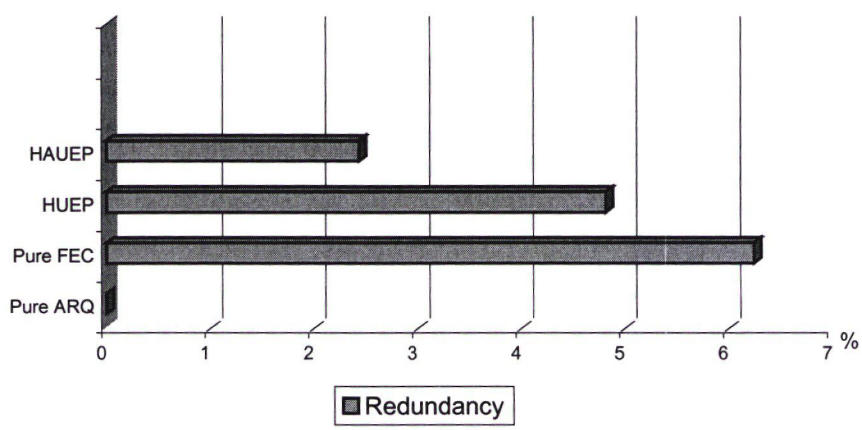



Figure 4.6. Average PSNR/Redundancy/Delay with different error protection scheme in Narrow bandwidth and Bursty Network (128k Bandwidth / PER=20%)



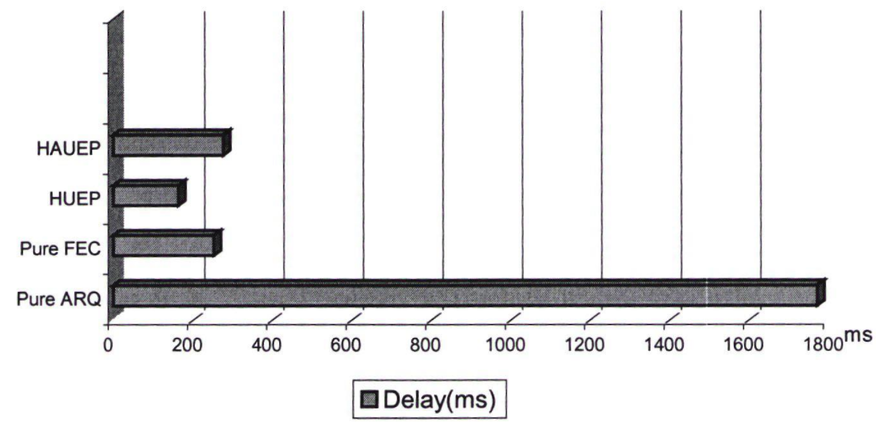
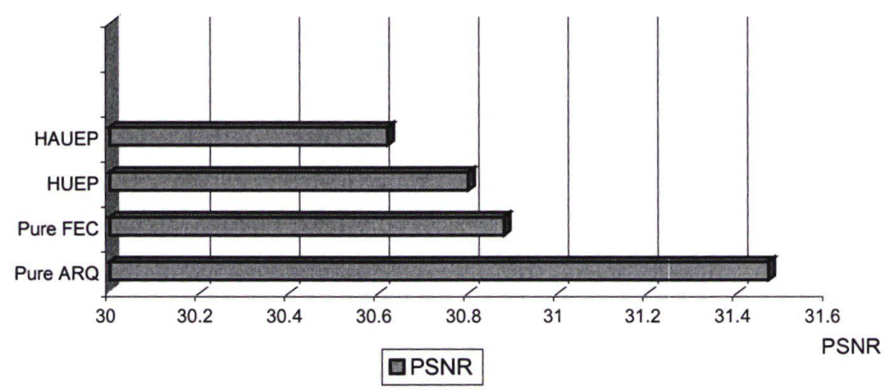
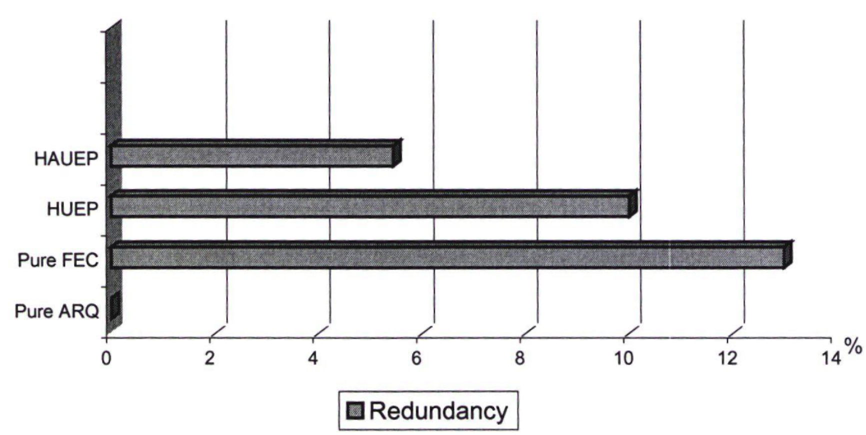



Figure 4.7. Average PSNR/Redundancy/Delay with different error protection scheme in Narrow bandwidth Network (128k Bandwidth / PER=0.5%)



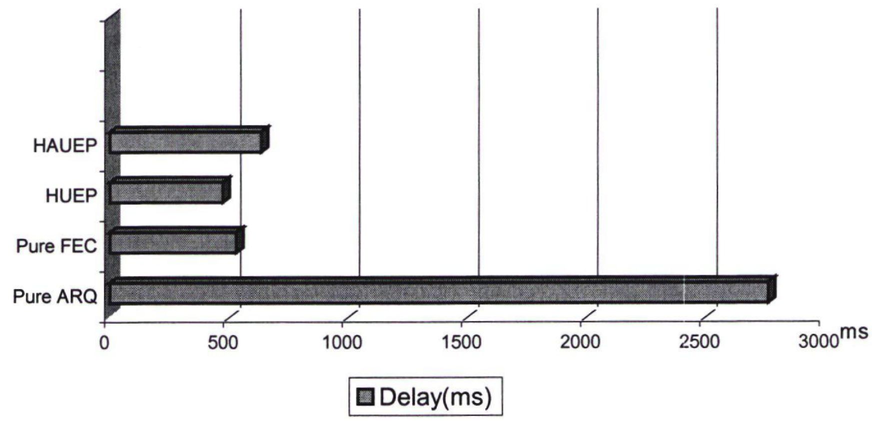
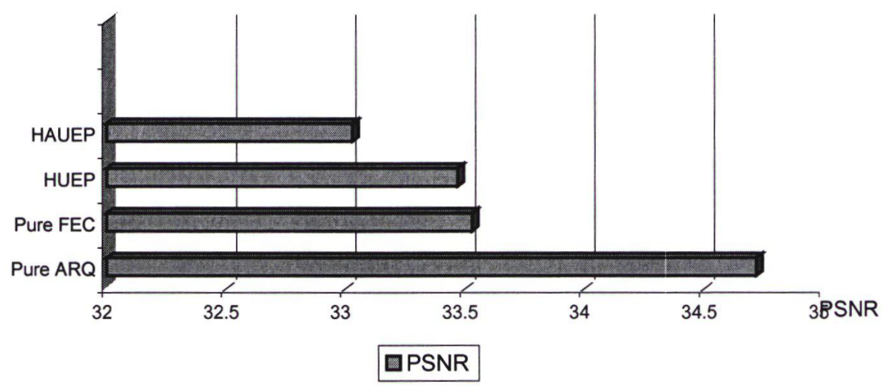
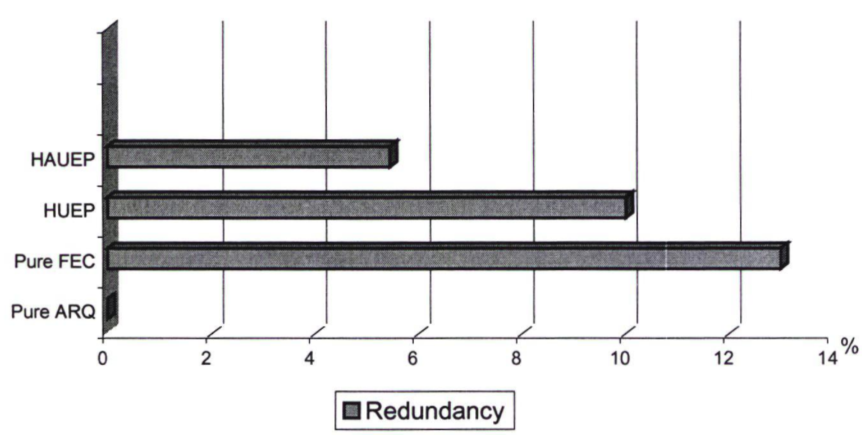



Figure 4.8. Average PSNR/Redundancy/Delay with different error protection scheme in Board bandwidth and Bursty Network 1.1M Bandwidth / PER=20%)



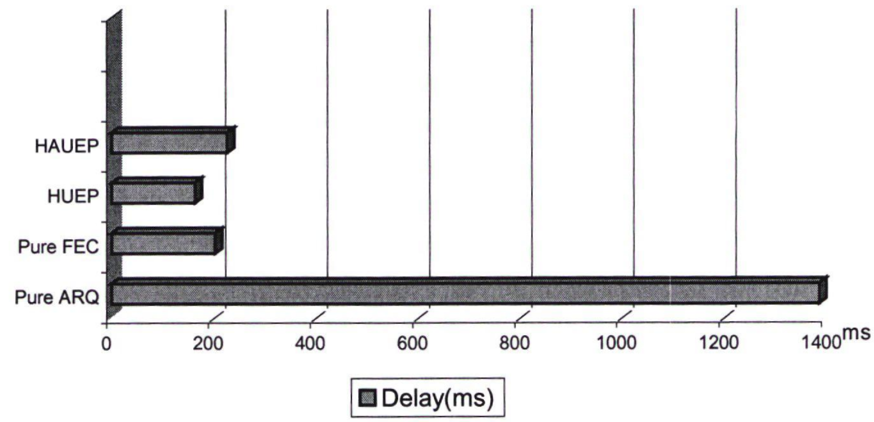



Figure 4.9. Average PSNR/Redundancy/Delay with different error protection scheme in Board bandwidth Network (1.1M Bandwidth / PER=0.5%)

From Figures 4.6 to 4.9, we can find that the pure FEC scheme works well in high bandwidth situation. But it performs very poorly in the low bit rate channel, because too much overhead occupies the limit network resource. Although the pure ARQ scheme can get the best PSNR in board bandwidth network, it increases delay significantly and the performance of pure ARQ in the bursty narrow bandwidth network is unacceptable. HUEP significantly reduces the redundancy. In the narrow bandwidth network, it still provides tolerable video quality in reasonable delay time. In the board bandwidth network, the performance is very close to one of the pure FEC, even it costs less redundancy. Because of the mesh selecting mechanism, HAUEP generates the least protection code. The throughput of this scheme is the highest in all test schemes. It gets the best PSNR in the low bit rate network. Moreover it also gets the adjacent result as other scheme with limit delay and least redundancy when the bandwidth is enough.

## 4.4 Performance analysis of Packet Shuffling

In this experiment, the network loss pattern is modeled by two-state Markov model as shown in section 2.2.3. The two states are good state and bad state. Since networks loose packets in bursty, once in the good state, the model remains there with probability $P_G$. Once it switches to the bad state with probability $1-P_G$ ,it remains there with probability $P_B$. The test frames are CIF sequence Flowergarden, which is the standard sequence in video testing. Each frame is divided into 108 packets. Each packet contains 22 blocks.

As bandwidth and network loss are the main parameters which effect continue loss factor, we measured the impact of these parameters in the experiment.

| | Mean of Consecutive Lost Factor Before Shuffling | Mean of Consecutive Lost Factor After Shuffling |
|---|---|---|
| $P_B$=0.2 | 1.36 | 1.04 |
| $P_B$= 0.5 | 1.82 | 1.53 |
| $P_B$=0.7 | 2.53 | 2.26 |
| 256k bandwidth | 2.79 | 2.21 |
| 1.1M bandwidth | 1.82 | 1.53 |

Table 4.1.Experiments on impact of network loss and available bandwidth

First three rows in Table 4.1 show the experiment on impact of network loss, the bandwidth is fixed at 1.1Mb/s, buffer size is 108 packages. The results demonstrate that the shuffling scheme reduces the continue loss factor at different network situations. The next two experiments show the shuffling scheme also perform better with different available bandwidth. Here the buffer size is 108 packages and the $P_B$ is set to 0.5.

## 4.5 Performance Analysis of DPLA

To compare the efficiency of the proposed adaptive packet length with the fixed packet length, we collected the efficiency value of 3 lengths with 7 different channel situations. The adaptive packet length is gotten from the look up table which is the approximate calculated value using EFF function.

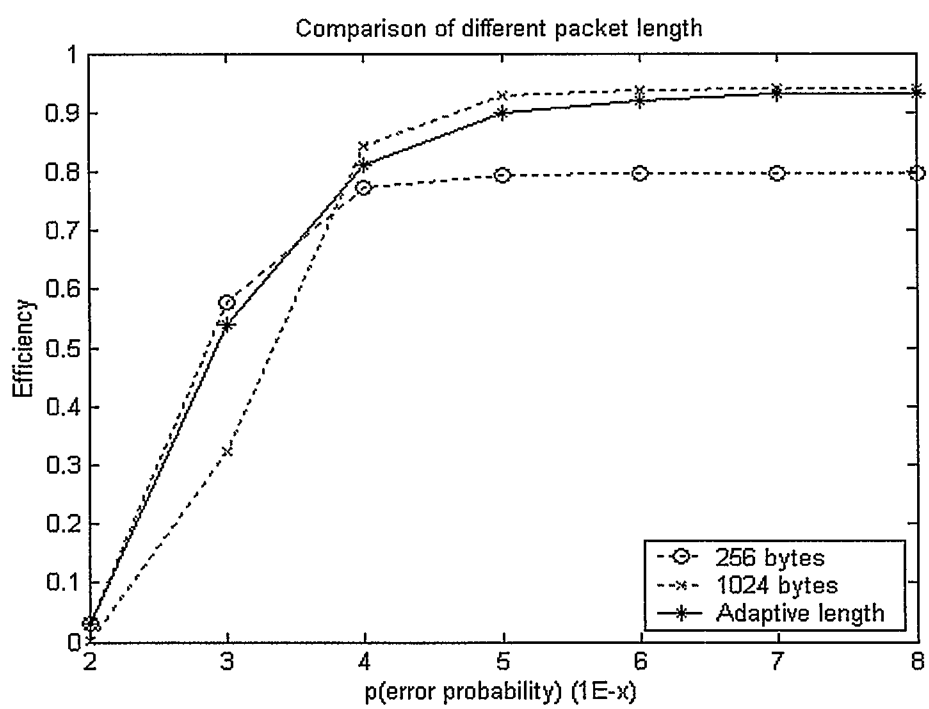



Figure 4.10. Efficiency comparison with different packet length

Figure 4.10 lists the relationship between the efficiency value and packet length in different network situation. We record the throughput value from error probability is $10^{-1}$ to $10^{-8}$. Experiments show that small packet size has better performance in the worse channel. Big packet size has better throughput in the situation with less error. Only the proposed adaptive packet length can get the near optimal throughout in any situation.

# CHAPTER 5

# CONCLUSION & RECOMMENDATION

## 5.1 Conclusion

This thesis studied the principles of video coding, the characteristics of communication networks, and generic error control techniques. Based on these studies, a robust video streaming platform over wireless networks is designed and prototyped. Specially, several schemes for video transmission are proposed, including Hierarchical Packetization Protocol, Adaptive Unequal Error Protection, Packet Shuffling and Dynamic Packet Length Adjustment Scheme.

To reduce the overhead of bit stream over the network, a special protocol is designed to generate minimum bytes to describe the mesh structure and represent motion compensation data. In the proposed Hierarchical Packetization Protocol, the essential information of quadtree, layer status and motion vectors are combined into bit streams. At the client side, mesh reconstruction can stop at an arbitrary layer according to those received correct mesh data packets. This feature offers a robust and reliable decoding of mesh data packets. Even in case of network congestion, limited quantity of mesh packets are received, smooth video playback on the client side is still guaranteed with reasonable QoS decrease. Furthermore, HPP uses more elegant and efficient algorithm to represent the mesh structure comparing with the tradition algorithms.

In the HAUEP scheme, the adaptive Reed Solomon coding scheme and ARQ is integrated to enable robust video transmission. Proposed hybrid error control scheme specially addresses the problem of mesh based video streaming over wireless channels. We also analyze the packet structure of video frames generated by mesh-based codec and arrange corresponding error protection method on different types of package. Moreover, an adaptive control mechanism is designed to effectively adjust the redundancy according to the situation of the wireless channel. Simulation shows this scheme is offering a better quality than other popular approaches

Packet Shuffling is a novel method to effectively average the continuous error to random error. The changing error pattern could effectively avoid that the error recovery capacity of FEC is exceeded. Another benefit is that the random error pattern can lower the video distortion from the view of human perception, compared with continuous error pattern. We design an algorithm to get the permutation to minimize the continuous error for a given network loss. Experiments show this algorithm successfully spread the continuous error. This scheme has significant value in the narrow bandwidth network with heavy bursty error.

A fast packet length adjustment scheme is also proposed. Based on the fact that Packet Length severely effects the packet truncation in wireless data transmission, we propose this scheme to find and apply the optimal packet length in the real time system. Normally the search algorithm of finding the optimal packet length with the given network bit error rate is very complex. In the proposed system, a fast lookup table is designed to avoid the complicated computation. In this lookup table, different error rate $p$ is mapped with correspond to packet length $k$. System can fast retrieve $k$ with given $p$. This approximate solution significantly reduces the calculation time and makes this scheme practically used in the real-time system.

## 5.2 Future work

Although the proposed system has been presented with promising results, the extensive modifications and further research still need to be carried out. Regarding further research work, following aspects are considered:

Currently all experiments were executed in the simulation environment. In the future, we need test the system in the realistic wireless networks. More complex channel situations, such as channel mismatch cases and channel lost, need to be further investigated.

Additionally, the proposed transmission module and error control module are flexible and could be expanded in most standard video codec, such as H.263 and MPEG-4. Further test need to be performed to compare these module's portability and performances in different video codec.

Moreover, presently the proposed error control scheme runs less efficiently on the PC platform, because it is designed for generic application. In this case, dedicated hardware device will have a much better performance. Due to the parallel and repetitive nature of video processing, the algorithms used in this system can be implemented with low-power VLSI design with much higher performance and lower cost. In the future, this system is supposed to be implemented with hardware.

**REFERENCE**


[1] E.D.et.al.,"UTMS/IMT-2000 based on wideband CDMA." IEEE Communication Magazing, pp.20-80, September 1998.

[2] ITU-T Recommendation H.261, "Video codec for audiovisual services at px64 kb/s," 1990.

[3] ITU-T Recommendation H.263, "Video coding for low bitrate communications," March 1990.

[4] ITU-T Draft Recommendation H.263 Version 2 (H.263+), "Video coding for low bitrate communications," September1997.

[5] W.Pennebaker, "JPEG technical specification, Revision 8," Working Document No. JTC1/SC2/WG10/JPEG-8-R8, August 1990.

[6] ISO/IEC IS 11172 (MPEG-1), "Coding of moving pictures and associated audio for digital storage media up to about 1.5 Mbits/s," Technical Report, Motion Picture Experts Group, 1993.

[7] ISO/IEC DIS 13818 (MPEG-2), "Generic coding of moving pictures and associated audio information." Technical Report, Motion Picture Experts Group, 1994.

[8] ISO/IEC JTC1/SC29/WG11 N4668 (MPEG-4), "Coding Of Moving Pictures And Audio: MPEG-4 Overview" Motion Picture Experts Group, 2002.

[9] R. E. Blahut, "Theory and Practice of Error Control Codes," Addison Wesley, MA, 1984.

[10] W. Badawy, "Hierarchical Adaptive Structure Mesh for Efficient Video Coding," The International Journal on Image and Video Processing, Vol 17, November 2001, pp. 171-178.

[11] A. M. Tekalp, "Digital video processing," Prentice-Hall, ISBN 0-13-190075-7, August 1995.

[12] P. Symes, "Video Compression demystified," McGraw-Hill, ISBN 0-07-136324-6, November 2000.

[13] P. Hsu, K. J. R. Liu and T. Chen, "A low bit-rate video codec based on two-dimensional mesh motion compensation with adaptive interpolation," IEEE Transactions on Circuits and Systems for Video Technology, Vol.11, No.1, pp. 111-117, January 2001.

[14] P. J. L. Beek, A. M. Tekalp, N. Zhuang, I. Celasun, and M. Xia, "Hierarchical 2D mesh representation, tracking and compression for object-based video," IEEE Transactions on Circuits and Systems for Video Technology, Vol.9, No.2, pp. 353-369, March 1999.

[15] Y. Altunbasak and A. M. Tekalp, "Closed-form connectivity-preserving solutions for motion compensation using 2-D meshes," IEEE Transactions on Image Processing, Vol. 6, No.9, pp. 1255-1269, September 1997.

[16] V. Parthasarathy, J.W.Modestino, and K.S. Vastola, " Design of a transport coding scheme for high-quality video over ATM networks," IEEE Trans. on Circuits and Systems for Video Technology, Vol.6, No.5, October 1996.

[17] K.Stuhlmuller, M.Lind, and B.Girod, "Scalable internet video streaming with unequal error protection," Packet Video Workshop, New York, April 1999.

[18] D.Wu et.al., "An end-to-end approach for optimal mode selection in Internet video communiation: theory and application," IEEE Journal on Selected Areas in Communications , Vol.18, No 6, pp.977-995, June 2000.

[19] M.Rabbani and P.W. Jones, "Digital image compression techniques," Vol. TT7. SPIE optical Engineering Press, 1991.

[20] J.G.Proakis, "Digital Communications," McGraw-Hill, Inc., 1995.

[21] V.K.Garg and J.E.Wilkes, "Wireless and Personal Communication Systems," Prentice Hall PTR , 1990.

[22] M.Schwartz, "Telecommunication Networks: Protocols, Modelling and Analysis," Addison-Wesley Publishing Company, 1987.

[23] Y.B.Lin and I. Chlamtac, "Wireless and Mobile Network Architectures," John Wiley & Sons, 2000.

[24] I.H.Witten, R.M. Neal, and J.G.Cleary, "Arithmetic coding for data compression," Communications of the ACM, Vol. 30, No. 6, pp. 520--540, June 1987.

[25] A.Said and W.A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," IEEE Trans. on CSVT, Vol. 6, June 1996.

[26] A.Skodras, C.Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," IEEE Signal Processing Magazine, Vol. 18, pp.36-58, September 2001.

[27] G.J.Sullivan and T.Wiegand, "Rate-distiortion optimizaition for video compression," IEEE Signal Processing Magazine, Vol. 15, No. 6, pp. 74-90, Nov. 1998.

[28] I.Moccagatta, S.Soudagar, J.Liang, and H.Chen, "Error-resillient coding in JPEG-2000 and MPEG-4," IEEE Journal on Selected Areas in Communications, Vol. 18, No. 6, pp. 899-914, June 2000.

[29] B.Girod and N.Farber, "Wireless video," in Compressed Video over Networks, Marcel Dekker, Inc. New York, 2000.

[30] H.S.Wang and N.Moayeri, "Finite-state Markov channel – a useful model for radio communication channels," IEEE Trans. on Vechiculur Technology, Vol.44, No.1, pp 163-171, February 1995.

[31] F.F.Kuo, W.Effelsberg, and J.J.Garcia-Luna-Aceves, "Multimedia Communications: Protocols and Applications," Prentice Hall PTR, 1998.

[32] D.Wu et.al., "On end-to-end architecture for transporting MPEG-4 video over the Internet," IEEE Trans. on Circuits and Systems for Video Technology, Vol.10, No.6, September 2000.

[33] S.Jiang, D.H.K.Tsang, and S.Gupta, "On architectures for broadband wireless systems," IEEE Communications Magazine, vol 35, No.10, pp.132-140, October 1997.

[34] R.V.Cox et.al., "On the applications of multimedia processing to communications," Proceedings of The IEEE, Vol. 86 ,No.5, pp. 755 -824 May 1998.

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

[36] Q.Zhang and S.A.Kassam, "Hybrid ARQ with selective combining for fading channels," IEEE Journal on selected areas in communications, Vol.17, No.5, May 1999.

[37] A.E.Mohr, E.A.Riskin, and R.E.Ladner, "Unequal loss protection: graceful degradation of image quality over packet erasure channels through forward error correction," IEEE JSAC special issue on Error-Resilient Image and Video Transmission, Vol.18, No.6, pp.819-828, June.2000.

[38] D.Wu and Y.Hou, "Transporting real-time video over the Internet: challenges and approaches," Proceedings of The IEEE, Vol.88, No.12, pp.1855 –1877, Dec.2000.

[39] Y.Yue, J.Zhou, Y.Wang, and C.W.Chen, "A novel two-pass VBR coding algorithm for fixed size storage applications," IEEE Trans. on Circuits and Systems for Video Technology, pp.345-356, March 2001.

[40] H.Song and C.C.J.Kuo, " Rate control for low-bit-rate video via variable encoding frame rates," IEEE Trans. on Circuits and Systems for Video Technology, pp.512-521. April 2001.

[41] Q. Zhang, W. Zhu, and Y. Zhang, "Network-adaptive Rate Control with TCP-Friendly Protocol for Multiple Video Objects," IEEE International Conference on Multimedia and Expo (ICME), New York. July 2000.

[42] J. Lu, "Signal Processing for Internet Video Streaming: A Review," Proceedings of SPIE Image and Video Communications and Processing, January 2000.

[43] Shan Du, Wael Badawy and K.V.I.S.Kaler " On Reducing the mesh overhead: a hierarchical adaptive merge and split structured mesh (HAMSM) approach" The 1st IEEE International Symposium on Signal Processing and Information Technology, pp.29-30, Cairo, Egypt, December 2001.

[44] A.M.Tekalp, P.V.Beek, C.Toklu, and B.Gunsel, "Two-dimension for interactive synthetic/natural digital video". Proc. Of the IEEE, Vol.86, No.6, pp.1029-1051, June 1998.

[45] D.Wu, Y. T. Hou, B. Li, W. Zhu, Y. Zhang, and H. J. Chao, "An end-to-end approach for optimal mode selection in Internet video communication: theory and application," IEEE Journal on Selected Areas in Communications, Vol. 18, No. 6, pp. 977-995, June 2000.

[46] S. McCanne, M. Vetterli, and V. Jacobson, "Low-complexity video coding for receiver-driven layered multicast," IEEE J. Selected Area Comm., Vol. 15, No. 6, pp. 983–1001, August 1997.

[47] P. Chou and K. Ramchandran, "Clustering source / channel rate allocations for receiver-driven multicast with error control under a limited number of streams," IEEE Intl. Conf. Multimedia Expo., July 2000.

[48] R. Rejaie, M. Handley, and D. Estrin. "An end-to-end rate based congestion control mechanism for real time streams in the internet". In Proceedings of INFOCOMM 99, 1999.

[49] G. Cheung, and A. Zakhor, "Bit Allocation for Joint Source/Channel Coding of Scalable Video," IEEE Transactions on Image Processing, pp.340-356, March 2000.

[50] G. Feng and T. S. P. Yum, "Traffic Analysis for Multiparty Videoconferencing in Virtual Path Based ATMNetworks," Int. J. Commun. Systems, Vol. 12, pp. 79–96, Feb. 2000.

[51] C. Zhou and Y. Yang, "Wide-sense nonblocking multicast in a class of regular optical WDM networks," IEEE Trans. Commun., Vol. 50, pp.126–134, January 2002.

[52] X. Jia, X.-D. Hu, L. Ruan, and J. Sun, "Multicast routing, load balancing and wavelength assignment on tree of rings," IEEE Commun. Lett., Vol.6, pp. 79–81, Feburary 2002.

[53] D. Ooms, B. Sales, W. Livens, A. Acharya, F. Griffoul, F. Ansari "Framework for IP Multicast in MPLS," IETF, IEEE 802.17 Resilient Packet Ring Working Group (2002), 2002.

[54] W. Li, "Overview of fine granular scalability in MPEG-4 video standard," IEEE Trans. Circuits. Syst. Video Technol., Vol. 11, pp. 385–398, March 2001.

[55] C.Weiß, T. Stockhammer, and J. Hagenauer, "The far end error decoder with application to image transmission," in Proceeding Globecom 2001, Vol.2, pp.1405–1409, November 2001.

[56] T. Stockhammer and C.Weiß, "Channel and complexity scalable image transmission," in Proceeding. ICIP 2001, Vol. 1, pp. 102 -105, October 2001.

[57] G. Caire and D. Tuninetti, "The throughput of hybrid-ARQ protocols for the Gaussian collision channel," IEEE Trans. Inform. Theory, Vol. 47, pp. 1971–1988, July 2001.

[58] R. Zhang, S. L. Regunthan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," IEEE J. Select. Areas Commun., Vol. 18, No. 6, pp. 966–976, June 2000.

[59] G. Cote, S. Shirani, and F. Kossentini, "Optimal mode selection and synchronization for robust video communications over error-prone networks," IEEE J. Select. Areas Commun., Vol. 18, pp. 952–965, June 2000.

[60] Q. Zhang and S. A. Kassam, "Hybrid ARQ with selective combining for fading channels," IEEE J. on Selected Areas in Comm., Vol. 17, No. 5, pp. 867-880, 1999.

[61] P. K. McKinley and S. Gaurav, "Experimental evaluation of forward error correction on multicast audio streams in wireless LANs," in Proceedings of ACMMultimedia 2000, pp. 416–418, November 2000.

[62] G. Cheung, A. Zakhor, "Bit allocation for joint source/channel coding of scalable video," IEEE Transactions on Image Processing, Vol.9 No.3, pp. 340 –356, March 2000.

[63] W. Badawy and M. Bayoumi, "Algorithm-Based Low Power VLSI Architecture For 2d-Mesh Video Object Motion Tracking," The IEEE Transaction on Circuits and Systems for Video Technology, Vol 12, No. 4, April 2002, pp. 227-237.

[64] W. Badawy and M. Bayoumi, "On Minimizing Hierarchical Mesh Coding Overhead: (HASM) Hierarchical Adaptive Structured Mesh Approach," IEEE International Conference on Acoustics, Speech, and Signal Processing, Istanbul, Turkey, June 5-9, 2000, pp. 1923-1926.

[65] C. Hsu, A. Ortega and M. Khansari "Rate Control for Robust Video Transmission over Burst-Error Wireless Channels," IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 17, NO. 5, MAY 1999.

[66] S. Ci, H. Sharif, and A. Young. "Frame size prediction for indoor wireless network," IEE Electronics Letters, 2001.

[67] J. S. Plank, “A Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-like Systems,” Software-Practice & Experience, pp. 995-1012, September, 1997.

[68] H. M. Radha, M. Schaar, Y. Chen,“The MPEG-4 Fine-Grained Scalable Video Coding Method for Multimedia Streaming Over IP,” IEEE Tran. Multimedia, Vol. 3, No. 1, March 2001.

[69] L. Cai, X. Liu and W. Badawy, “Hierarchical adaptive control scheme for video streaming over Internet,” International Conference on Multimedia and Expo ICME02, Vol.1, pp. 9 -12, August 2002.

[70] L. Cai, X. Liu and W. Badawy, ”Agent-based hierarchical video presentation over Internet,” International Conference on Cognitive Informatics ICCI02, pp.182–185, 2002.

[71] X. Liu and W. Badawy, “A Novel Error Control Scheme For Video Streaming Over Ieee802.11 Network” Accepted by Canadian Conference on Electrical and Computer Engineering, 2003 Montréal, Canada.

[72] X. Liu and W. Badawy “An adaptive error control method for real time wireless video streaming” Accepted by International Conference on Information Technology:Research and Education , August 10-13, 2003, New Jersey, USA.

[73] X. Liu and W. Badawy “ Hybrid Adaptive Error Protection in Mesh based Video Streaming:” Accepted by ITCOM 2003 Multimedia Networks and Management Systems, Orlando, Florida USA.