MULTI-LEVEL ADAPTIVE VIDEO STREAMING OVER WIRELESS CHANNELS

A THESIS IN ELECTRICAL ENGINEERING

Presented to the faculty of the American University of Sharjah
School of Engineering
in partial fulfillment of
the requirements for the degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

by
HUSAMELDIN MUKHTAR
B.S. 2004

Sharjah, UAE
May 2010

MULTI-LEVEL ADAPTIVE VIDEO STREAMING OVER WIRELESS CHANNELS

Husameldin Mukhtar, Candidate for Master of Science in Electrical Engineering

American University of Sharjah, 2010

ABSTRACT

Delivery of digital video over wireless networks offers added functionalities and advantages over traditional wired video systems. However, reliable video streaming over wireless channels is fraught with several challenges. Video streaming has strict requirements on bandwidth, delay, and loss rate. In addition, wireless channels are dynamic and error-prone by nature. In this thesis, multi-level adaptive approaches are proposed to mitigate these challenges. The objective of these approaches is to ensure continuous playback with acceptable video quality. First, bitstream switching is combined with adaptive playback to accommodate variations in the channel condition and to avoid interrupted video playback. Second, scalable coding, adaptive modulation, and adaptive channel coding are integrated to achieve efficient video streaming. A probabilistic approach is used to adequately scale video frames to ensure successful delivery within a buffer-state-dependent budget time. Adaptive modulation and channel coding help reduce the amount of required scaling, hence, improving the quality of the received video.

Moreover, in this thesis, new temporal quality metrics are introduced for the evaluation of video quality. These quality metrics are the skip length (SL) and inter-starvation distance (ISD). A quality assessment system which takes into account these

two metrics is implemented. Experimental results show that the proposed system is capable of estimating the peak signal to noise ratio (PSNR) of the received video frames without the need of the reference video. Hence, it provides a better alternative to the conventional PSNR calculation approach.

Contents

List of Figures

List of Tables

Abbreviations

ACK	Acknowledgment
ARQ	Automatic Repeat Request
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CBR	Constant Bit Rate
CR	Code Rate
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
FEC	Forward Error Correction
GBN	Go-back-N
GoB	Group of Blocks
GoP	Group of Pictures
HP	High Priority
HQAM	Hierarchical Quadrature Amplitude Modulation
ISD	Inter-Starvation Distance
JSCC	Joint Source Channel Coding
LP	Low Priority
MAD	Mean Absolute Difference
MDC	Multiple Description Coding
MGS	Medium Grain Scalability
MOS	Mean Opinion Score

MSB	Most Significant Bit
MTU	Maximum Transmission Unit
MV	Motion Vector
NACK	Negative Acknowledgment
PSNR	Peak Signal to Noise Ratio
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
RTT	Round Trip Time
SL	Skip Length
SNR	Signal to Noise Ratio
SR	Selective Repeat
SW	Stop-and-Wait
VBR	Variable Bit Rate

Acknowledgments

I would like to thank my thesis advisors, Dr. Mohamed Hassan and Dr. Taha Landolsi for their guidance and continuous support. They provided me with valuable technical knowledge and personal advice. I am also grateful for their motivation and constructive criticism which pushed me to do my best. I would also like to thank Dr. Tamer Shanableh for his help and contribution in the video quality assessment work.

I am thankful to all my professors especially from the Electrical Engineering Department for their great teaching which helped me understand complicated engineering problems. In addition, I thank the College of Engineering for offering me the graduate assistantship.

Finally, I would like to thank my family for their endless support. I gratefully acknowledge the love and devotion of my parents who have always inspired me.

Chapter 1

Introduction

Delivery of multimedia contents over wireless channels is becoming increasingly popular. Recent advances in wireless access networks provide a promising solution for the delivery of multimedia services to end-user premises. In contrast to wired networks, wireless networks not only offer a larger geographical coverage at lower deployment cost, but also support mobility. Nevertheless, challenges still exist for extending video streaming in the wireless dimension [1].

Wireless channels are dynamic and error-prone by nature. However, video streaming requires high bandwidth. It also has strict requirements on the end-to-end delay and delay jitter especially for live and interactive video. Variations in the transmission delay greatly impact the quality of the received video. It is a common practice in most video streaming applications to prefetch few video frames in the decoder buffer before the start of the playback to smooth out the variations in the transmission delay. In general, it is required to match the arrival rate of video frames at the decoder buffer with the playback rate of the video player to avoid buffer starvation. Moreover, most standard video encoding techniques are sensitive to packet loss where error might propagate for successive inter-dependent frames causing significant degradation in the decoded video quality.

Different approaches have been proposed in the literature that constitute a solution space for the above challenges. Examples of these approaches are video compression, source rate control, bitstream switching, error control, adaptive modulation, power allocation, transcoding, and adaptive playback [2, 3, 4, 5, 6, 7, 8].

In this work, we integrate several methods in a multi-level adaptive approach

to achieve efficient video streaming. The objective of our multi-level adaptive methods is to ensure uninterrupted playback with acceptable video quality at the client side. Furthermore, we introduce a no-reference video quality assessment system that takes into account new temporal quality metrics, skip length (SL) and inter-starvation distance (ISD) . The proposed system allows the estimation of the received video quality at the client side without the need of the original video. This information can be communicated back to the sender to act accordingly and maintain a required quality of service (QoS) level.

The following two sections provide background information about the methods (solution space) considered in our study and the new metrics used in our video quality evaluation.

1.1 Solution Space

1.1.1 Video Compression

Raw digital video contains an immense amount of data. It is comprised of a time-ordered sequence of still images (frames). These images are required to be displayed at a certain rate so that objects' motion in a video sequence is perceived as continuous and natural by the human eye. Therefore, digital video transmission is considered one of the most bandwidth demanding data communication applications. Despite the recent advances in communication networks, channel bandwidth is still considered a scarce resource [2]. Hence, source coding and compression are essential in practical digital video communication. Video compression is comprised of three main stages. These stages are motion compensation, transform coding, and quantization and binary encoding.

Motion Compensation

Motion compensation is a compression technique which exploits video temporal redundancy. In a video sequence, adjacent frames are very similar. Consequently, significant compression can be achieved by only encoding the differences between video frames. Most video codecs (e.g. MPEG-2, MPEG-4) implement motion compensation. A video sequence is encoded into 3 main frame types, namely, I, P, and B frames. I frames are 'intra-coded', independently of other frames using still image compression techniques (e.g. JPEG). On the other hand, P and B frames are inter-coded based on previous or future encoded frames. P frames are 'predictively' coded based on previous I or P frames, while B frames are 'bi-directionally predicted' based on both previous and future I or P frames. B frames can also be used as a source of prediction as in the H.264 standard [9]. B frames achieve the highest compression level compared to other frame types. Nevertheless, I frames, which achieve low compression ratios, are introduced at regular intervals to help recover from transmission

errors and limit error propagation among successive inter-coded frames. The number of frames between two consecutive I frames is known as the group of pictures (GoP) .

Transform Coding

Transform coding involves transforming video frames from the spatial domain into a more compact representation. Examples of transform coding are Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) . DCT is a transformation technique which is widely used in video compression. It provides transformation coefficients which represent video frames in the frequency domain. Unlike the data in the pixel domain, these coefficients are separable and with unequal importance. Knowing the fact that video frames (images) are low-frequency data by nature, compression can be achieved by considering the most important DCT coefficients which are the low-mid frequencies coefficients. Therefore, DCT is capable of reducing the spatial redundancy within a video frame by averaging out similar areas of color. On the other hand, DWT is a more sophisticated transformation with inherent scalability. In addition, DWT overcomes a drawback of block-based DCT known as blocking artifacts [10].

Quantization and Binary Coding

The final stage of video compression is quantization and binary coding. Quantization is a lossy compression technique where the transform coefficients are approximated by a discrete set of integer values. These quantized values are then represented in bits and additional compression is realized by exploiting the redundancy in the bitstream. Entropy coding techniques such as run length coding, differential coding, and Huffman coding are applied to reduce this redundancy.

1.1.2 Source Rate Control

Source rate control is employed to adapt the source rate to the channel bandwidth variations with the objective of ensuring continuous playback [4]. This mechanism is implemented at the transmitter side. In certain schemes, the receiver monitors the channel condition and the state of decoder buffer [11]. This information is fed back to the transmitter to adapt the source rate to match the available channel capacity. For example, when the available bandwidth decreases, the source rate will be reduced to avoid playback interruption by gracefully degrading the video quality.

Scalable coding is the most elegant source rate control mechanism. Scalable coding provides scalability to heterogeneous network links and video clients. Layered coding and multiple description coding (MDC) are the two classes of scalable coding. Layered coding encodes the video sequence into a base layer and multiple enhancement layers. The base layer provides a version of the original video sequence with minimum acceptable quality, whereas enhancement layers provide incremental improvement to the video quality when they are received. Nevertheless, enhancement layers can not be decoded without the base layer. Examples of layered coding techniques are spatial, temporal, and SNR layered coding. Figure 1.1 provides a general illustration of these layered coding techniques.

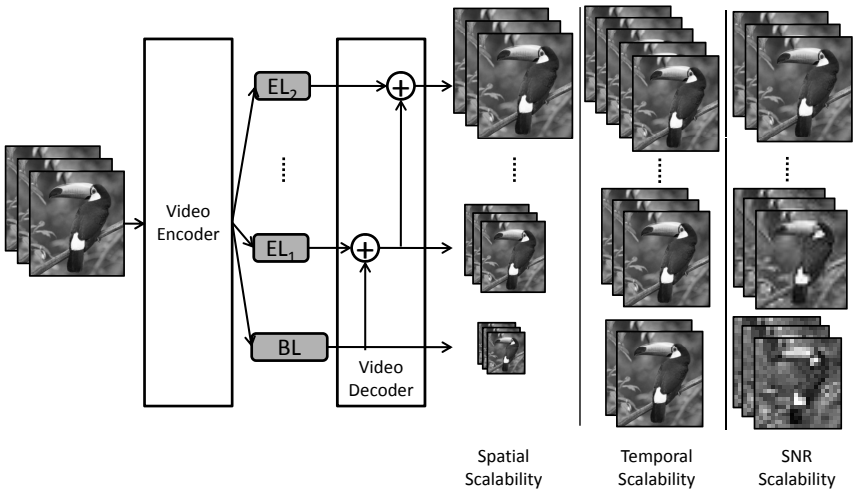


Figure 1.1: Layered Coding Techniques

On the other hand, MDC encodes the video sequence into multiple descriptions (bitstreams). Unlike enhancement layers in layered coding, descriptions in MDC can be separately decoded. With each additional description, incremental improvement in video quality is achieved [2].

1.1.3 Bitstream Switching

Bitstream switching is another adaptation technique which could be used when the other techniques do not guarantee the continuity of the playback. It requires the availability of several pre-encoded versions of the same video source with different encoding bit rates. Switching between the different bitstreams occurs according to the channel variations and decoder buffer occupancy. Usually, switching takes place at I frames to avoid the error drift problem [12]. Nevertheless, new types of encoded video frames (SI and SP) are introduced in [13] to facilitate a more flexible drift-free switching.

1.1.4 Error Control

Error control mechanisms in video streaming can be classified into 4 classes. These classes are forward error correction (FEC) , automatic repeat request (ARQ) , error resilient coding, and error concealment.

Forward Error Correction

FEC is a channel encoding technique which adds redundancy to the bitstream. The introduced redundancy is structured in relation to the original data of the bitstream. An error in the received data will alter this structure and hence can be detected or even corrected. Hamming code, Reed-Solomon code, and convolutional codes are examples of FEC. FEC methods can improve throughput and can be static or adaptive. Adaptive FEC provides a more effective error control method where the FEC code rate is adapted to the channel state. In general, FEC introduces transmission

latency due to the added redundancy. Nevertheless, this latency can be reduced by reducing the source rate to accommodate the FEC bits at the cost of slight reduction in video quality [11]. In addition, FEC can be jointly designed with the source coder to achieve effective video transmission [14]. This is often referred to by joint source channel coding (JSCC) [15] in which the channel coder provides different levels of protection based on the importance of source information.

Automatic Repeat Request

Another class of error control is retransmission or automatic repeat request. In this class, error detection techniques (e.g. parity check and CRC) are applied at the receiver to detect errors. The receiver sends acknowledgment (ACK) or negative acknowledgment (NACK) messages to the transmitter to indicate whether a transmitted message was received correctly or not. In addition, the transmitter may initiate retransmission based on a timeout if ACK or NACK messages are delayed more than expected. This timeout is set relative to an estimated round trip time (RTT) which is the time between sending a message and receiving a positive ACK following the last successful retransmission of the same message. RTT can be estimated using a moving average of previously measured RTTs as shown below:

$$RTT_{estimated} = \delta RTT_{old} + (1-\delta) RTT_{sample}, \tag{1.1}$$

where δ is a weighting factor ($0 \leq \delta \leq 1$) to specify the sensitivity of estimated RTT to changes in RTT sample values. There are 4 main types of ARQ protocols, namely, Stop-and-Wait (SW) , Go-back-N (GBN) , Selective Repeat (SR) , and hybrid ARQ which is a combination of ARQ and FEC [2]. The operation of SW ARQ is depicted in Figure 1.2. The transmitter sends a packet and waits for its acknowledgment. SW ARQ is inefficient compared to GBN and SR because of the idle time spent waiting for an ACK or NACK. In GBN ARQ, the transmitter sends packets continuously. At the receiver, if a packet is received in error, it will be discarded and a NACK will be sent to

the transmitter. The receiver continues to discard frames until the originally discarded frame is received correctly. Upon receiving a NACK, the transmitter resends all packets that have not yet been positively acknowledged as shown in Figure 1.3. SR ARQ is similar to GBN ARQ with the difference that only negatively acknowledged packets are retransmitted as shown in Figure 1.4.

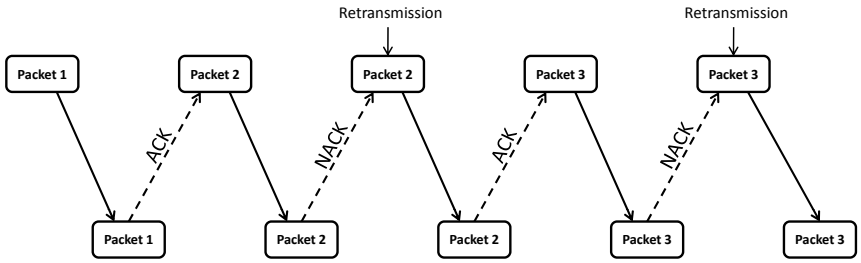


Figure 1.2: Stop-and-Wait ARQ

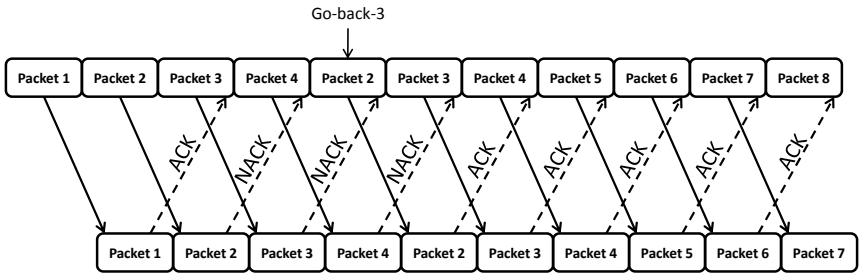


Figure 1.3: Go-back-N ARQ

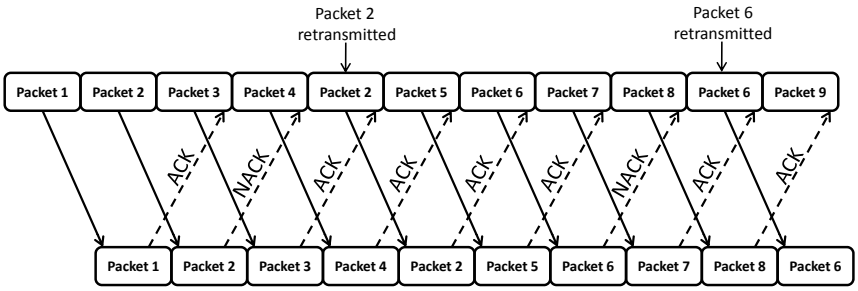


Figure 1.4: Selective Repeat ARQ

Error Resilient Coding

Error resilient coding is a source error control method which improves the immunity of encoded video against errors or packet loss. Scalable coding, especially MDC, is considered a type of error resilient coding. When part of the bitstream (e.g. an enhancement layer) is corrupted by errors the remaining bit stream can still be decoded to reconstruct video frames with slight degradation in quality. Another example of error resilient coding is slice structured coding in which the video frame is spatially partitioned into groups of blocks (GoBs) . Each slice is then transmitted in a separate network packet introducing multiple synchronization points. In the event of a packet loss, the associated GoB is lost but the remaining parts of the frame can still be successfully decoded. In addition, data partitioning is another scheme which divides the different parts of a bit stream into groups according to their importance. For example high frequency transform coefficients are grouped together and considered of low importance. Data partitioning is usually combined with an unequal error protection scheme.

Error Concealment

Error concealment is another class of error control schemes which is implemented at the receiver with the objective to conceal data loss. Most error concealment techniques exploit spatial and/or temporal interpolation. Spatial interpolation approximates missing pixel values using neighboring pixel values. On the other hand, temporal interpolation approximates lost data from previous video frames [16].

1.1.5 Adaptive Modulation

Adaptive modulation is a possible solution in which the modulation level is changed according to the channel condition and/or the buffer occupancy for effective bandwidth utilization and continuous playback. Increasing the level of modulation or the number of constellation points allows more bits per symbol, but at the same time

increases the bit error rate (BER) for a given signal to noise ratio (SNR) [17]. Hence, when the channel is in a bad state, robust low-level modulation schemes such as binary phase shift keying (BPSK) can be used whereas when the channel is in a good state higher level modulation schemes such as 64-quadrature amplitude modulation (64-QAM) can be used to achieve higher data rates. Adaptive modulation can be jointly designed with forward error correction (FEC) while it could also be integrated with the source encoder to achieve effective video transmission [14].

Hierarchical Modulation

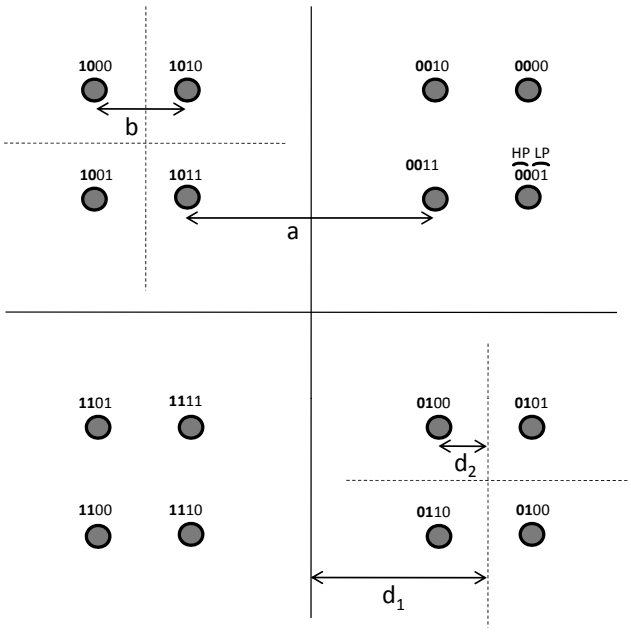
Hierarchical modulation is an interesting variation of conventional modulation. It virtually divides a transmission channel into multiple sub-channels with unequal error protection without an increase in bandwidth [18]. A single bitstream can be separated into several multiplexed sub-streams with different levels of priority. The degree of protection of a sub-stream and the levels of hierarchy are controlled by the distances between constellation points (or regions) [19]. Figure 1.5 shows two examples of hierarchical constellations, one for 16-QAM and the other for 64-QAM. The highest priority (HP) sub-stream is transmitted using the most significant bits (MSBs) while the lower priority (LP) sub-streams are transmitted using the subsequent bits.

Hierarchical QAM (HQAM) is one of the popular hierarchical modulation schemes. It has already been incorporated in some digital video transmission standards such as DVB-T [20]. Hierarchical modulation can also be applied to other modulation schemes. [21] is an example of implementing hierarchical DPSK modulation.

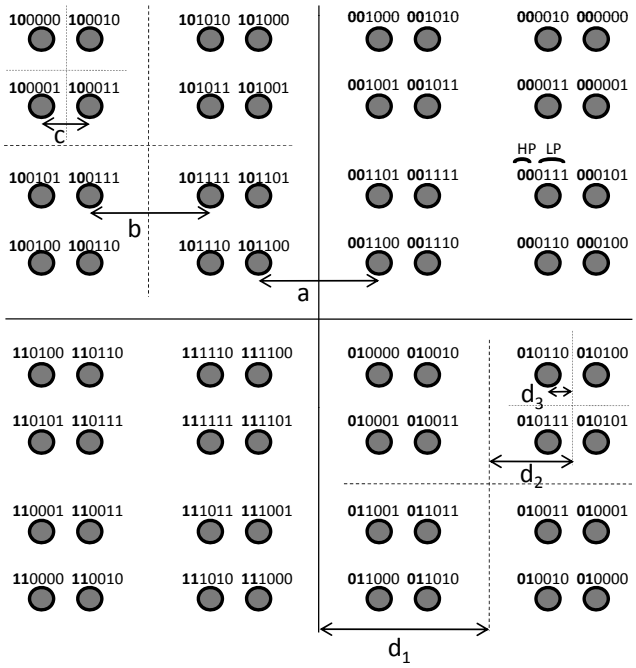
A classical HQAM is the 64-HQAM where the conventional 64-QAM is transformed into three levels such that each level is associated with 2 bits. It is also possible to group two levels to be considered as one level and assign 4 bits to it.

To control the relative degrees of protection between the levels, the ratios between the constellation distances ($\alpha = a/b$, $\beta = b/c$) are adjusted. a, b, and c are

defined in Figure 1.5. Figure 1.6 shows the effect of changing α on the bit error rate (BER) performance of the 16-HQAM.

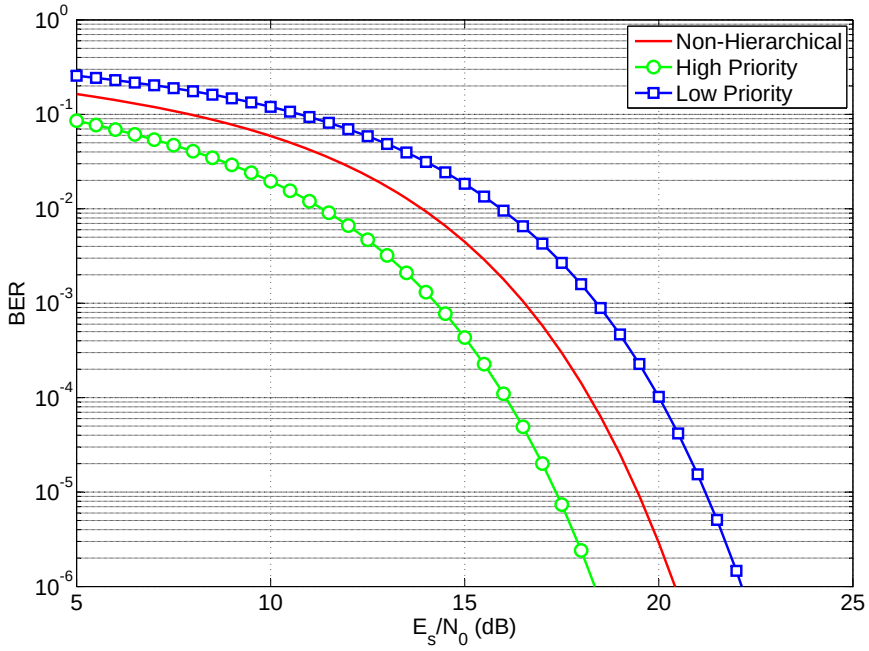


(a) Hierarchical 16-QAM

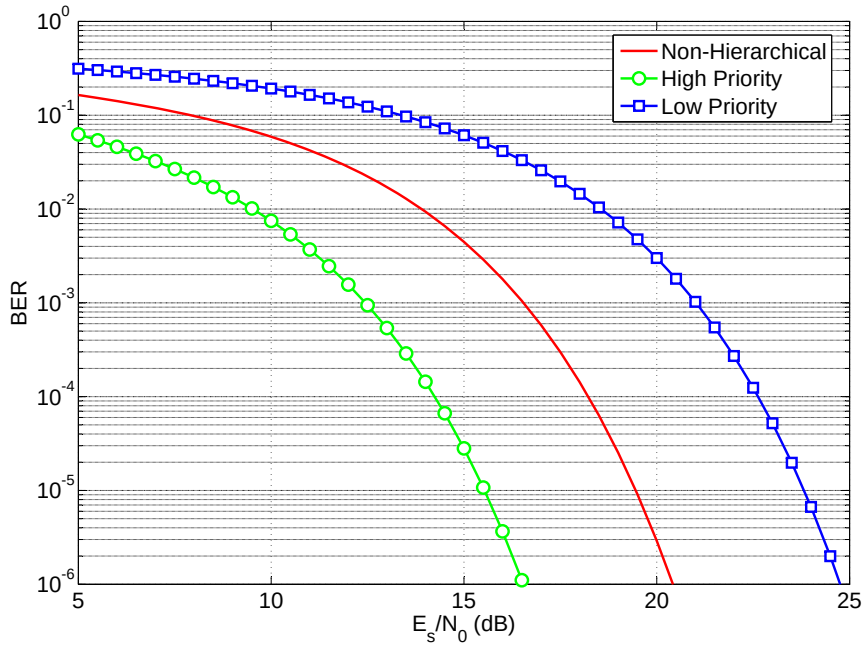


(b) Hierarchical 64-QAM

Figure 1.5: Hierarchical constellation for a)16-QAM, and b)64-QAM



(a) 16-HQAM with $\alpha = 1.5$



(b) 16-HQAM with $\alpha = 2.5$

Figure 1.6: BER performance for a)16-HQAM with $\alpha = 1.5$, and b)16-HQAM with $\alpha = 2.5$

1.1.6 Adaptive Playback

Adaptive playback controls the video playback rate in an attempt to maintain a desired buffer occupancy at the video decoder. When the decoder buffer occupancy is below a predefined threshold, the playback rate is reduced to allow the buffer occupancy to increase. Conversely, when the decoder buffer occupancy is above the threshold reflecting good channel condition, the playback rate is increased to drain possible accumulation in the slow phase to prevent the video sequence from being desynchronized.

1.2 Video Quality Metrics

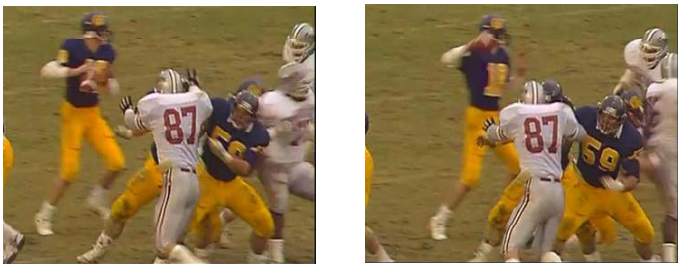
Various video quality assessment techniques are proposed in the literature [22, 23, 24, 25]. Assessment techniques in which quality metrics are mainly based on mathematical quantification are classified as objective approaches. Other assessment techniques that rely on viewers perception of the video quality are classified as subjective. In general, video quality has two aspects: spatial and temporal. Spatial video quality is typically measured using peak signal to noise ratio (PSNR) metric. Temporal quality pertains to the viewer perception of the screen changes with time. It is usually measured using a subjective approach such as the mean opinion score (MOS) .

Each of the two commonly used measures (PSNR or MOS) has its own drawbacks. MOS assessment is time consuming, slow, and expensive. PSNR is a full-reference metric that requires a priori knowledge of the original video sequence which is typically not available at the client side. In addition, it is known that PSNR values are not necessarily correlated with perceptual quality. For example, consider Figure 1.7. This figure shows frames 65 and 68 of the "football" video sequence that was encoded using the H.264/AVC JM encoder [26]. The original two frames are shown in Figure 1.7(b) for the sake of comparison. The transmission process was intentionally disturbed to result in the loss of frame 65. In Figure 1.7(a), we concealed the loss of this frame by freezing the previous frame. Frame 68 of Figure 1.7(a) shows the impact of error propagation when frame copy is used as the concealment method. On the other hand, in Figure 1.7(c) the loss of frame 65 was concealed by motion copy [27]. Similarly, frame 68 of Figure 1.7(c) shows the impact of error propagation when motion copy is used. Clearly, the frame in Figure 1.7(a) is of a better perceptual quality when compared to that in Figure 1.7(c). Nevertheless, the PSNR in Figure 1.7(c) is 2 dB higher than that of Figure 1.7(a). However, this may not be the case for future frames that might reference this concealed frame. As a result, a high MOS value could be associated with a relatively low PSNR. Therefore, we argue that PSNR, besides being full-reference metric, is not enough to assess the video quality

in the presence of transmission errors. These errors could lead to playback buffer starvation which in turn degrades the temporal quality.

Frame 65 Frame 68

(a) Concealed Frame by Freezing Previous Frame



Frame 65 Frame 68

(b) Original Frame

Frame 65 Frame 68

(c) Concealed Frame by Motion Copy

Figure 1.7: Frames 65 and 68 of the"football" sequence

We also argue that, in addition to perceptual quality metrics, an efficient streaming scheme should also consider a *transmission quality metric* which quantifies the ability of the underlying wireless links to reliably transport video. While this metric reflects the quality of service as provided by the network rather than the quality of the reconstructed video stream, it has a direct impact on both spatial and temporal quality. This is true because of the inherent frame-interdependencies existing in

current video coding schemes, whereby correct decoding of a given frame requires correct decoding of previous (and sometimes future) "reference" frames. Hence, timely delivery for reference frames must be guaranteed with a higher probability than for other frames. This is not always possible due to the variable bit rate (VBR) nature of video compression if a constant perceptual quality is required. The resulting frame size varies depending on the scene dynamics and the types of compression involved (e.g., intra-coding, motion prediction, etc.). Therefore, when the video stream is generated and transported at a constant frame rate, it displays a VBR traffic pattern that is difficult to transport efficiently over any packet network let alone wireless ones.

To quantify the effect of losing frames when frame-interdependencies exist, we propose a spatiotemporal measure that complements PSNR and could replace it. This measure reflects the temporal quality through the continuity of the played back video. Namely, we propose a metric that we call the "*skip length*" as a measure for temporal quality. On the occurrence of any starvation instant, the skip length indicates how long (in frames) this starvation will last on average. The rationale behind skip length as a metric for temporal quality is the fact that it is better for the human eye to watch a continuously played back video at a lower quality rather than watching a higher quality video sequence that is frequently interrupted. We propose an additional temporal quality metric that emanates from the skip length called the "*inter-starvation distance*". It is the distance in frames that separates successive starvation instants. This metric complements the skip length in the sense that if the latter is small but very frequent then the quality of the played back video would be degraded. Therefore, large inter-starvation distances in conjunction with small skip lengths would result in a better played back video quality. Figure 1.8 illustrates the definitions of these two metrics.

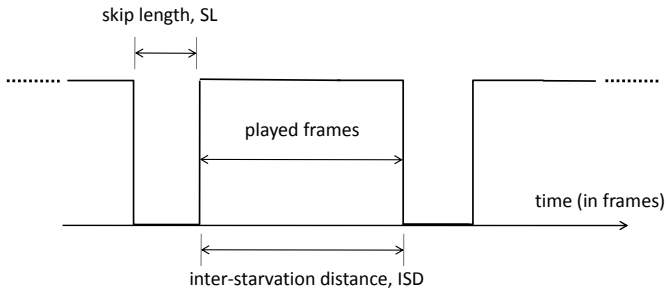


Figure 1.8: Definitions of skip length and inter-starvation distance metrics

We propose a PSNR estimation approach that is based on the skip length SL, the inter-starvation distance ISD, and additional bitstream information. The estimation can be performed at the decoder without the need for the reference video unlike the conventional computation of PSNR as shown in Equation 1.2.

$$PSNR = 10\log_{10}\left(\frac{1}{M\ N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\frac{255^2}{\left[f(i,j)-g(i,j)\right]^2}\right). \tag{1.2}$$

$f(i,j)$ is the original video frame and $g(i,j)$ is the distorted version of the video frame after being received and decoded. 255 is the largest (peak) possible pixel value when the number of bits per pixel is 8.

We argue that the proposed approach is a better alternative to full-reference PSNR calculation since the latter cannot be computed at the receiver. Moreover, the proposed metrics can be used to indicate the average PSNR at the client side. The validity of this statement is experimentally demonstrated in what follows.

To study the impact of the proposed metrics on the achieved PSNR, the transmission process of the encoded football sequence was disturbed such that it caused the losses of either 1, 2, or 3 frames for different values of inter-starvation distances. Figure 1.9 depicts the impact of inter-starvation distance on the PSNR for different skip length values. Intuitively, this figure shows that the best PSNR is achieved at smaller skip length values and starvation instants that are distant apart.

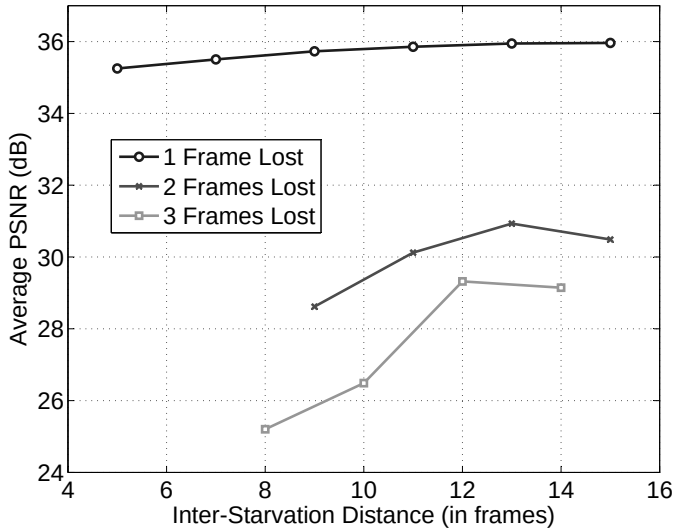


Figure 1.9: Average PSNR vs. inter-starvation distance

To gauge the impact of skip length alone on the PSNR, random frame loss of I, P, and B frames was also simulated. The received video was decoded with concealment. Figure 1.10 shows that the relationship between the average PSNR and the skip length was not monotonically decreasing. This is explained by the fact that PSNR degradation is not only related to the skip length but also to the type of the lost frames. Losing an I frame has a worse effect on the video quality than losing a P or B frame due to the error drift problem. Indeed, Figure 1.10 shows that the average PSNR when 8 frames were lost was higher than the average PSNR when 4 frames were lost. That is because in the latter case an I frame was lost thus hindering the PSNR more severely than the former case where no I frames were lost. This clearly shows the importance of the type of lost frames on the video quality.

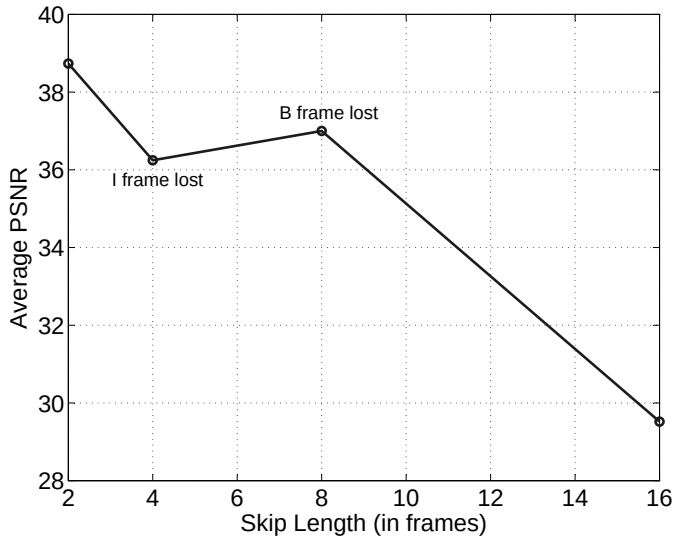


Figure 1.10: Average PSNR vs. skip length when I, P, or B frames are lost and concealed

Chapter 2

Methodology

Figure 2.1 describes our video streaming model. The video bitstream is transmitted over an unreliable forward channel whereas feedback information is transmitted over a reliable reverse channel. Scalable coding or bitstream switching along with adaptive playback, adaptive modulation, and adaptive channel coding are implemented based on the channel condition and playback buffer occupancy. Further detail on the system model and the proposed schemes will be presented in Sections 2.1 and 2.2.

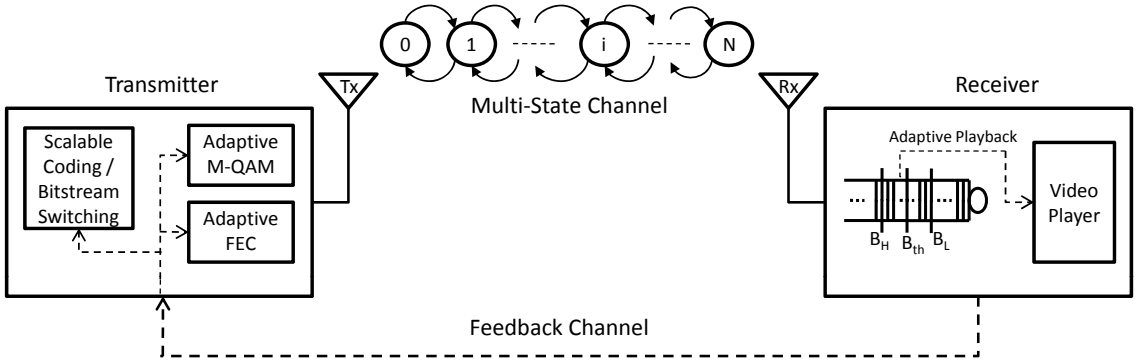


Figure 2.1: Architecture of a video streaming system over a varying wireless channel

2.1 Adaptive Playback and Bitstream Switching

In this section, we introduce adaptive playback combined with bitstream switching. At the transmitter, multiple versions of a video sequence are pre-encoded and made available for bitstream switching. On the receiver side, adaptive playback is implemented. Based on the playback buffer occupancy, the video player adapts the playback rate in a manner to reduce the number of buffer underflow events. The adaptive playback rate, f_p is decided using:

$$f_p = \begin{cases} 0.75 f_n & \text{if } B_i < 0.75 B_{th} \\ \dfrac{B_i}{B_{th}} f_n & \text{if } 0.75 B_{th} \leq B_i \leq 1.25 B_{th}, \\ 1.25 f_n & \text{if } B_i > 1.25 B_{th} \end{cases} \tag{2.1}$$

where f_n is the nominal encoding rate, B_i is the occupancy (state) of the playback buffer, and B_{th} is a specified buffer threshold. We set B_{th} to a small value, relatively, to limit its effect on the end-to-end delay. However, an optimal value of B_{th} can be selected depending on the channel coherence time [4]. Moreover, the playback rate is not allowed to deviate by more than $\pm\%25$ of the nominal playback rate (i.e. $0.75 f_n \leq f_p \leq 1.25 f_n$) so that the variations in playback will be unnoticeable [28]. Hence, we allow f_p to be a function of B_i only when $0.75 B_{th} \leq B_i \leq 1.25 B_{th}$. The receiver also sends information about its buffer state to the transmitter. Based on this information, the transmitter calculates an important parameter in this model which is the budget time (T_b). It is the time within which the frame candidate for transmission must be correctly received. We define T_b as follows:

$$T_b = \begin{cases} \dfrac{\gamma_1}{f_p} & \text{if } B_i < B_L \\ \dfrac{\gamma_2}{f_p} & \text{if } B_L \leq B_i \leq B_H. \\ \dfrac{\gamma_3}{f_p} & \text{if } B_i > B_H \end{cases} \tag{2.2}$$

where B_L is a low occupancy threshold while B_H is a higher occupancy threshold specified for the playback buffer (see Figure 2.1). γ_1, γ_2, and γ_3 are design parameters which can be functions of B_i or simply constants with the condition $\gamma_1 < \gamma_2 < \gamma_3$. Therefore, T_b reflects the urgency of frame arrivals at the playback buffer. For example, when the playback buffer is in an underflow state (i.e. $B_i < B_L$), T_b is set to a small value compared to values of T_b when $B_i \geq B_L$. The smaller the budget time, the more urgently frames should arrive to avoid starvation. A possible design can be: $\gamma_1 = 0.5\gamma_2, \gamma_3 = \gamma_2(B_i - B_H)$. For simplicity, it was assumed that each video frame is contained in one packet. Stop-and-Wait ARQ was also assumed. This assumption is justified for typical indoor wireless environments where the round trip time (RTT) between the access point and the wireless client is in the order of a few microseconds, which is much smaller than typical packet transmission times (in the order of several tens of microseconds or more). Hence, the transmitter estimates the transmission time (T_r) using:

$$T_r = \frac{S_f}{C} + RTT, \tag{2.3}$$

where S_f is the video frame size in bits and C is the channel bitrate in bits/sec.

Subsequently, bitstream switching is implemented at the transmitter by comparing T_r with T_b and switching between the available bitstreams to match T_r with T_b in an attempt to avoid buffer starvation. For example, if T_r is found to be greater than T_b then the transmitter will switch to a bitstream with lower encoding bitrate (smaller frame size) to achieve faster arrival of frames at the decoder buffer. We define the desired frame size (S_d) in Equation 2.4 such that a frame candidate for transmission will be received early enough to avoid buffer starvation.

$$S_d = \begin{cases} \frac{T_b}{T_r} S_f & \text{if } T_b < T_r,\ T_b \neq 0 \\ S_f & \text{if } T_b \geq T_r \end{cases} \tag{2.4}$$

2.2 Scalable Coding with Adaptive Modulation and Channel Coding

In this scheme, streaming of scalable compressed video is considered. Based on feedback information such as the playback buffer occupancy, the transmitter controls the encoding bitrate of the scalable compressed video and adapts the modulation level and channel coding rate to reduce the probability of decoder buffer starvation. The wireless channel is characterized by its BER denoted by p_i which is a function of the ratio of the energy per symbol (E_s) to the noise power spectral density (N_0). In M-ary modulation schemes, increasing the order of modulation level (i.e. increasing the number of bits per symbol) will increase the error-free channel bitrate by $\log_2 M$ at the expense of the BER performance. For square M-QAM (i.e. $\log_2 M$ is even) the analytical expression of the BER, in additive white Gaussian noise (AWGN) channels, is given in [29]. On the other hand, for the BER over Rayleigh fading channels, the expression can be obtained from [29] and [30].

Let C be the error-free channel bitrate for 2-QAM and $\bar{N}_{r_i}$ the average number of retransmissions needed to successfully transmit a packet in the presence of errors. For Selective Repeat ARQ, the number of retransmissions (including the first transmission attempt) is a geometric random variable with mean $\bar{N}_{r_i} = 1/P_{c_i}$ [31] where P_{c_i} is the probability of correctly receiving a packet which is given by [17]:

$$P_{c_i} = \sum_{j=0}^{\tau_{\max_i}} \binom{S_p}{j} p_i^j (1-p_i)^{S_p - j}, \tag{2.5}$$

where τ_{max_i} is the number of bits which can be corrected by an FEC scheme and S_p is the packet size including the FEC bits. In addition, when channel coding is implemented an overhead is added to the transmitted packets. Therefore, the effective channel bitrate C_i can be approximated by:

$$C_i = P_{c_i} \frac{k_i}{S_p} C \log_2 M, \tag{2.6}$$

where k_i is the payload size. Let $\epsilon_i = P_{c_i} k_i / S_p$. Equation 2.6 is now given by:

$$C_i = \epsilon_i C \log_2 M. \tag{2.7}$$

Clearly, $0 \leq \epsilon_i \leq 1$ and reflects the channel condition. For fixed FEC, τ_{max_i} is usually predefined and has a fixed value. On the other hand, in adaptive FEC, an "optimal" desired value $\tau^*_{max_i}$ could be determined based on the channel condition and the packet size. In [11] a reasonable approximation for $\tau^*_{max_i}$ is given by:

$$\tau^*_{max_i} \approx \left\lceil p_i S_p + 3\sqrt{p_i S_p (1 - p_i)} \right\rceil, \tag{2.8}$$

where $\lceil \cdot \rceil$ is the ceiling function which rounds a real number to the smallest following integer. Therefore, the transmission efficiency η_i for Selective Repeat ARQ is:

$$\eta_{i_{SR}} = \frac{C_i}{C} = P_{c_i} \frac{k_i}{S_p} \log_2 M. \tag{2.9}$$

Similarly, based on the analysis in [31], the transmission efficiency for Go-back-N and Stop-and-Wait ARQ protocols can be given by:

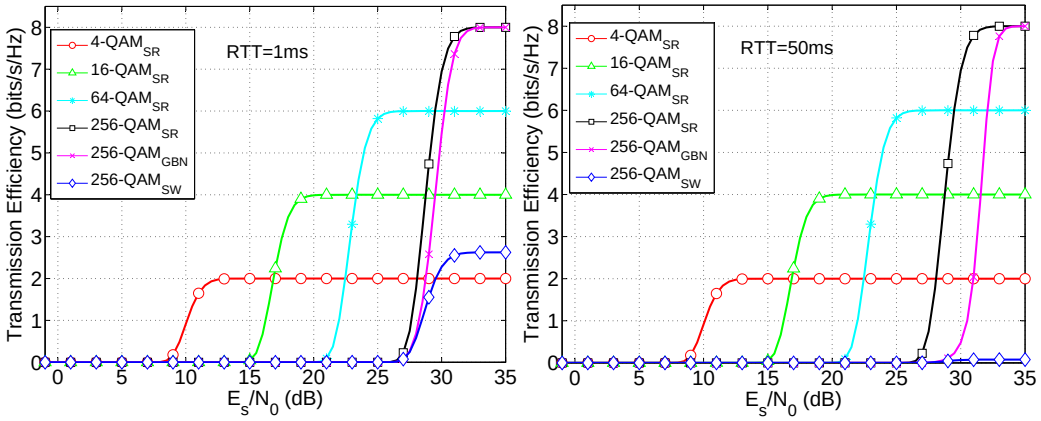
$$\eta_{i_{GBN}} = \frac{P_{c_i}}{P_{c_i} + K(1 - P_{c_i})} \frac{k_i}{S_p} \log_2 M, \tag{2.10}$$

$$\eta_{i_{SW}} = \frac{P_{c_i}}{K} \frac{k_i}{S_p} \log_2 M, \tag{2.11}$$

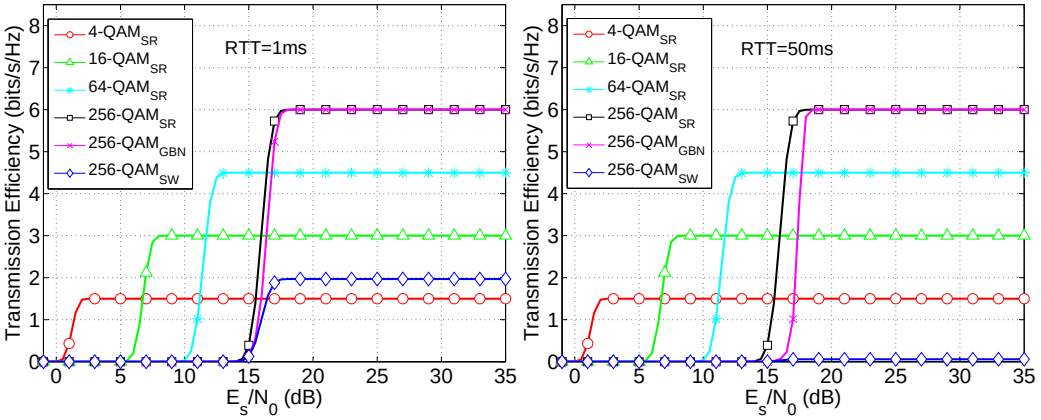
where $K - 1$ is the number of packets that could be transmitted during the RTT ($K = \dfrac{RTT \cdot C \cdot \log_2 M}{S_p} + 1$). For the GBN analysis, it was assumed that the window size of the retransmission buffer is selected such that the channel is kept busy all the time.

Figure 2.2 compares η_i of Selective Repeat ARQ for different QAM modulation levels with no FEC, fixed FEC, and adaptive FEC. η_i of Go-back-N and Stop-and-Wait is also shown for 256-QAM. Two RTT values were considered to depict the

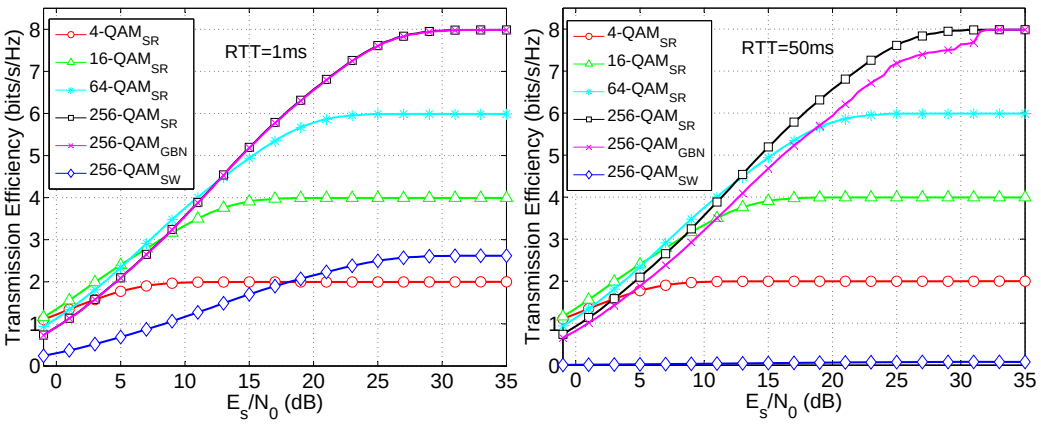
effect of RTT on η_i. The plots were generated assuming an AWGN channel, Reed-Solomon FEC, $S_p = 1000\ bits$, and $C = 256\ Kbps$. For fixed FEC, a code rate $CR = k_i/S_p = 3/4$ was assumed whereas for adaptive FEC $CR = (S_p - 2\tau^*_{max_i})/S_p$.



(a) No FEC (b) No FEC



(c) Fixed FEC (CR=$\frac{3}{4}$) (d) Fixed FEC (CR=$\frac{3}{4}$)



(e) Adaptive FEC (f) Adaptive FEC

Figure 2.2: Transmission efficiency of ARQ protocols for different QAM modulation levels (for (a), (c), and (e) RTT=1 ms and for (b), (d), and (f) RTT=50 ms)

Figure 2.2a shows that, with no FEC, 256-QAM achieves the highest transmission efficiency for $E_s/N_0 > 28.5$ dB when compared to the other modulation levels. However, for lower values of E_s/N_0 the BER performance of 256-QAM is significantly degraded. As a result, lower modulation levels can provide higher transmission efficiency in that range. 64-QAM provides the highest transmission efficiency for $23.5 < Es/N_0 < 28.5$ dB. On the other hand, 16-QAM efficiency is the highest for $16.9 < Es/N_0 < 23.5$ dB while 4-QAM efficiency is the highest for $E_s/N_0 < 16.9$ dB. Moreover, Figure 2.2c shows that fixed FEC improves the transmission efficiency for low E_s/N_0 values. Notice that the curves are shifted to the left when compared to the case with no FEC. This shift reflects the coding gain which is the difference between the E_s/N_0 values of the uncoded system and the coded system to achieve the same BER performance when FEC is used. However, at high E_s/N_0 values, unnecessary overhead is incurred preventing the modulation scheme from achieving its highest possible transmission efficiency which is equal to $\log_2 M$. Figure 2.2e shows that adaptive FEC outperforms fixed FEC. With adaptive FEC, the transmission efficiency is improved for even smaller E_s/N_0 values. At the same time, no unnecessary overhead is added during channel good states (i.e. high E_s/N_0 values) allowing for the realization of the maximum error-free bitrate. Based on these plots a decision can be made to use adaptive FEC with 16-QAM for $E_s/N_0 < 5$ dB, 64-QAM for $5 < E_s/N_0 < 12$ dB, and 256-QAM for $E_s/N_0 > 12$ dB for a packet size of 1000 bits to achieve the best bandwidth utilization.

In Figure 2.2, we see that Selective Repeat ARQ is the most efficient scheme and Stop-and-Wait ARQ is the least efficient. As shown in Equations 2.9, 2.10, and 2.11 Selective Repeat ARQ performance is not affected by the RTT. However, the performance of Stop-and-Wait ARQ and Go-back-N ARQ degrades when $RTT \cdot C \cdot \log_2 M$ is relatively large (relative to S_p). For large RTT values, the transmission efficiency of the Stop-and-Wait ARQ becomes unacceptable, whereas the bandwidth efficiency of Go-back-N ARQ drops rapidly as the channel SNR decreases when fixed FEC (or no FEC) is used (see Figure 2.2b and Figure 2.2d).

Moreover, when adaptive FEC is used, the difference in the performance between Selective Repeat ARQ and Go-back-N ARQ is significantly reduced even for a relatively large RTT value as shown in Figure 2.2e and Figure 2.2f. That is because, in adaptive FEC, $P_{c_i} \approx 1$ which makes $\eta_{i_{SR}} \approx \eta_{i_{GBN}}$ (see Equations 2.9 and 2.10). In other words, when $P_{c_i} \approx 1$, each packet is transmitted once on average making Go-back-N ARQ less detrimental when compared to a case with higher average number of retransmissions.

Source rate control is integrated with adaptive modulation and channel coding. The goal of this multi-level adaptive scheme is to reduce the probability of decoder buffer starvation by controlling the bitrate of the scalable compressed video. Moreover, adaptive modulation and channel coding are implemented to reduce the amount of required scaling thereby enhancing the quality of the received video. An expression for the probability of correctly receiving a video frame within a time constraint is obtained and used in our proposed rate control algorithm. For the sake of simplicity, a slowly varying channel is assumed where the channel state does not change during a frame delivery time. A frame may consist of multiple packets and each packet may require several retransmissions.

The time needed to transmit a packet until it is received correctly, $T_p^{(i)}$, is a linear function of a geometric random variable which is the number of retransmissions. This time can be approximated by an exponential distribution of mean $\lambda_i^{-1} = E(T_p^{(i)}) = k_i/\eta_i C$. Based on the analysis in [31] and [32], λ_i^{-1} for the three ARQ schemes is given as follows:

$$\lambda_{i_{SR}}^{-1} = \frac{Sp}{C \log_2 M} \frac{1}{P_{c_i}}, \tag{2.12}$$

$$\lambda_{i_{GBN}}^{-1} = \frac{Sp}{C \log_2 M} + \left(\frac{Sp}{C \log_2 M} + RTT \right) \frac{1 - P_{c_i}}{P_{c_i}}, \tag{2.13}$$

$$\lambda_{i_{SW}}^{-1} = (\frac{Sp}{C \log_2 M} + RTT)\frac{1}{P_{c_i}}. \tag{2.14}$$

For a given video frame size S_f and a network packet size S_p, the number of required packets N_p to contain the video frame is computed as:

$$N_p = \left\lceil \frac{Sf}{Sp - S_{o_i}} \right\rceil, \tag{2.15}$$

where S_{o_i} is the number of FEC bits. Hence, the total time $T_f^{(i)}$ needed to successfully deliver the whole video frame is gamma distributed with parameters λ_i and N_p. Accordingly, the probability of correctly receiving a frame within a time constraint is given by [11]:

$$F(T_b, i) = P(T_f^{(i)} \leq T_b) = 1 - e^{-\lambda_i T_b} \sum_{n=0}^{N_p - 1} \frac{(\lambda_i T_b)^n}{n!}, \tag{2.16}$$

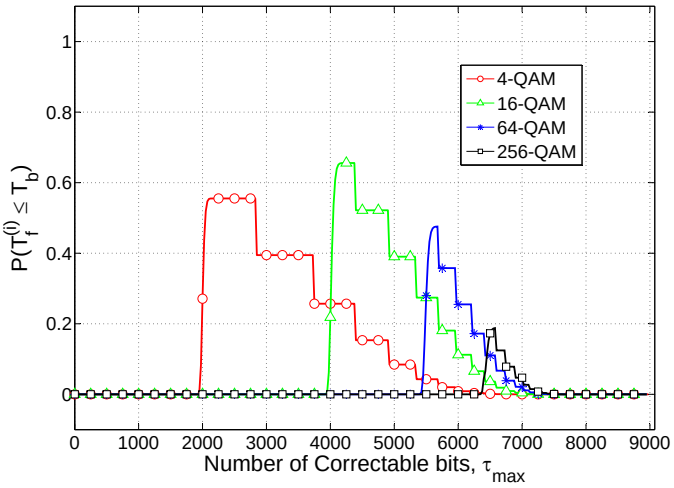
where T_b is the budget time defined in Section 2.1.

In this scheme, the transmitter determines T_b based on the buffer occupancy feedback information. Every time a frame is to be transmitted, the transmitter computes $F(T_b, i)$ for the different modulation levels and selects the level that achieves the highest $F(T_b, i)$. Nevertheless, if none of the modulation levels can achieve $F(T_b, i) \geq \omega$ where ω is a predefined probability, the transmitter reduces the size of the video frame by a scaling increment θ such that $S_f^{(new)} = \theta S_f$. The video frame size is reduced by discarding enhancement layers. Then, the transmitter recomputes $F(T_b, i)$ and repeats the process, if necessary, until $F(T_b, i) \geq \omega$. When compared to other rate control techniques which requires adjustment of encoding parameters, scalable coding is less complex and allows real time adjustment of the video frame size.

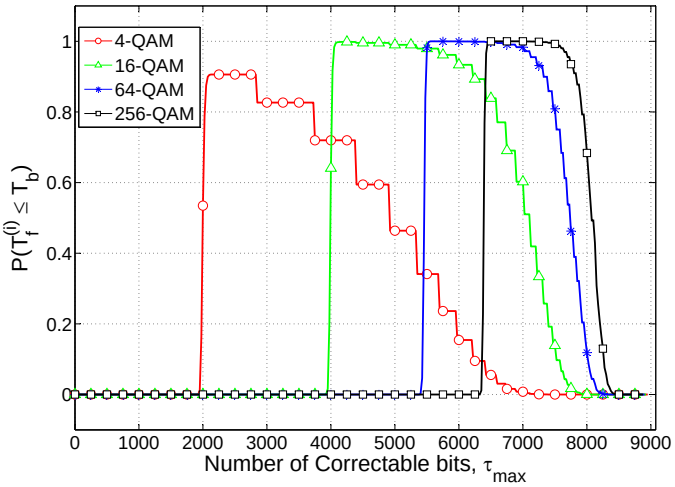
We now study the effect of channel coding (τ_{max}), channel condition (E_s/N_0), packet size, and frame size on $F(T_b, i)$ for different modulation levels with different

ARQ schemes. The modulation levels are 4-QAM, 16-QAM, 64-QAM and 256-QAM. A Rayleigh fading channel is assumed in the generation of the following analysis results.

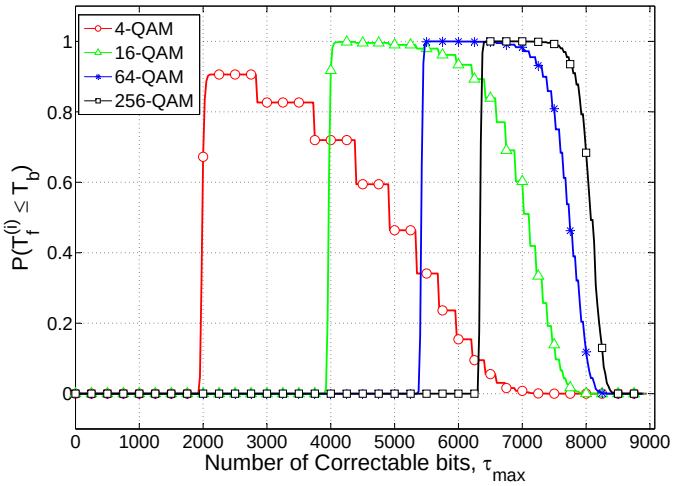
Figure 2.3 shows the effect of changing the amount of FEC (τ_{max}) on $F(T_b, i)$ for different levels of QAM modulation and different ARQ schemes. Increasing τ_{max} improves the performance (i.e. increases $F(T_b, i)$) of the modulation levels up to an optimum point after which the performance starts to degrade. This is due to the fact that increasing the number of FEC bits improves the probability of correctly receiving a packet, but at the same time, the number of required packets per frame increases hindering timely delivery of the video frame. As the modulation level increases the amount of required FEC increases for a low channel SNR (e.g. E_s/N_0=5 dB). In other words, arbitrary selection of the desired number of correctable bits can have a destructive effect on the performance of a transmission system. Moreover, we see that SR ARQ and GBN ARQ outperform SW ARQ. For the assumed parameters as shown in Figure 2.3, the difference in performance between SR and GBN is unnoticeable. However, at τ_{max}=2000 bits, it can be noticed that SR achieves higher $F(T_b, i)$. The staircase behavior in the plots is attributed to the ceiling function in Equation 2.15. $S_f = 9383$ *byte* is the average video frame size of the Harry Potter HD sequence when encoded with quantization parameters 28,28, and 30 for I, P, and B frames respectively [33]. $S_p = 2272$ *byte* is the MTU (Maximum Transmission Unit) in IEEE 802.11. Notice that $T_b = 167 = \frac{5}{30}$ *ms* can correspond to a scenario where 5 frames are available in the playback buffer with a playback rate equal to 30 fps.



(a) SW ARQ



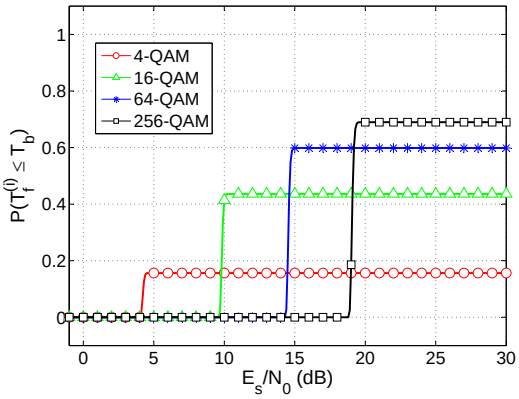
(b) GBN ARQ



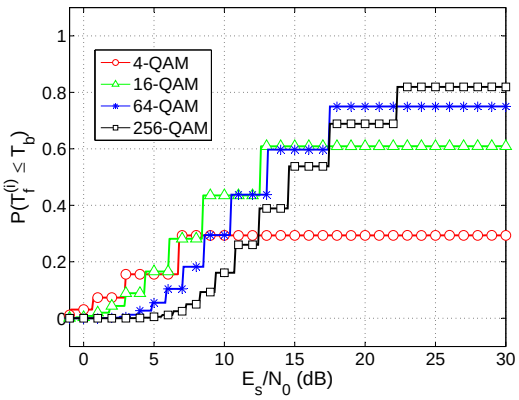
(c) SR ARQ

Figure 2.3: The probability of correctly receiving a frame within a time constraint vs. τ_{max} ($S_f = 9383\ byte, S_p = 2272\ byte, RTT = 10\ ms, E_s/N_0 = 5\ dB, T_b = 167\ ms, C = 512\ Kbps$)

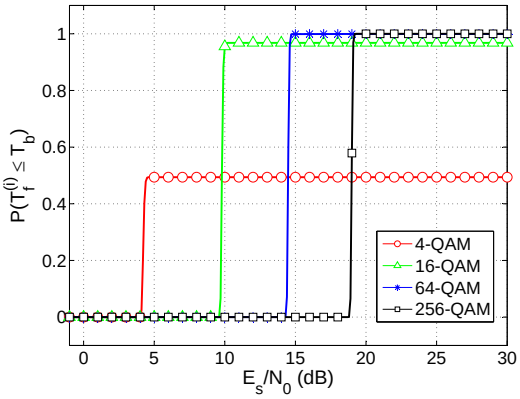
To study the effect of channel state on the modulation levels performance, $F(T_b, i)$ was plotted for different values of E_s/N_0 in Figure 2.4. Fixed FEC and adaptive FEC were considered. The plots exhibit a similar trend to the transmission efficiency plots in Figure 2.2. In Figure 2.4a, Figure 2.4c, and Figure 2.4e fixed FEC is used. It is observed that 256-QAM achieves the highest $F(T_b, i)$ for $Es/N_0 > 19.5$dB. However, for lower values of channel SNR, lower modulation levels can provide better performance. Moreover, adaptive FEC significantly improves $F(T_b, i)$ especially for high modulation levels as shown in Figure 2.4b, Figure 2.4d, and Figure 2.4f. The plots also support the argument that SR and GBN outperform SW.



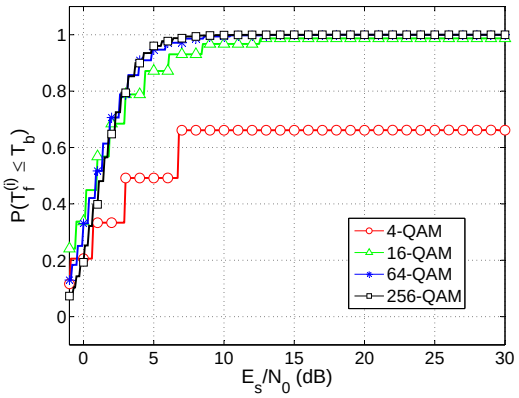
(a) SW with fixed FEC (CR=$\frac{3}{4}$)



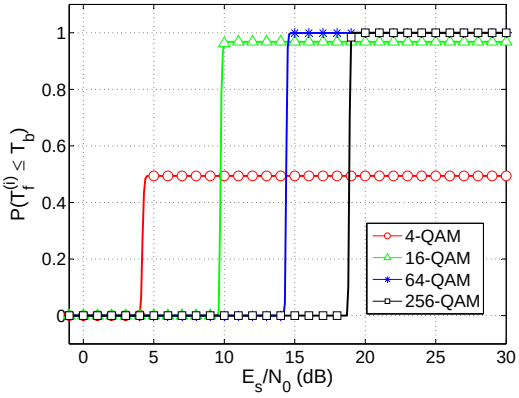
(b) SW with adaptive FEC



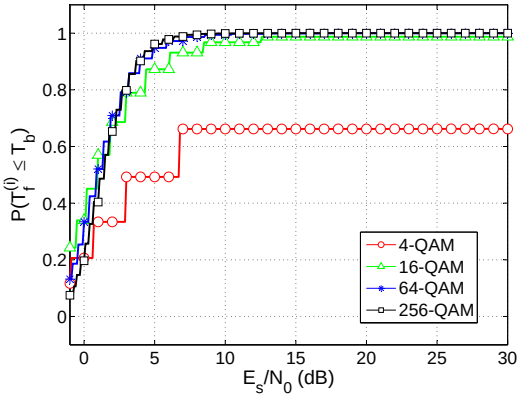
(c) GBN with fixed FEC (CR=$\frac{3}{4}$)



(d) GBN with adaptive FEC



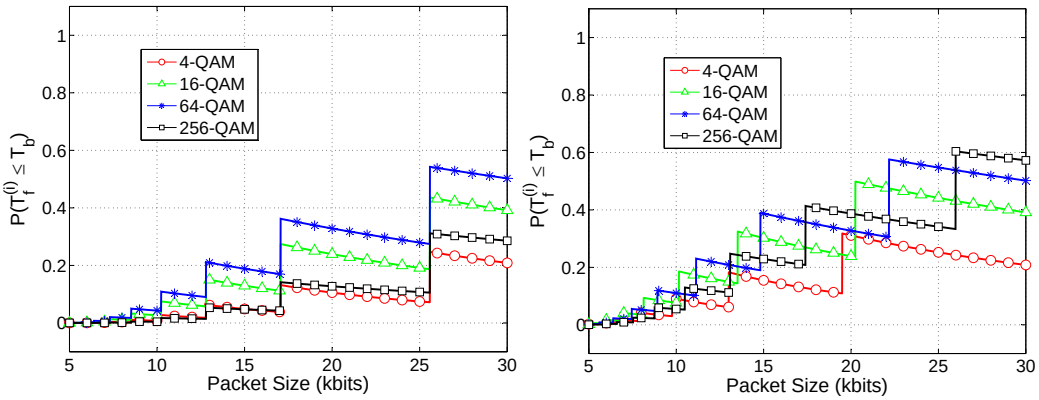
(e) SR with fixed FEC (CR=$\frac{3}{4}$)



(f) SR with adaptive FEC

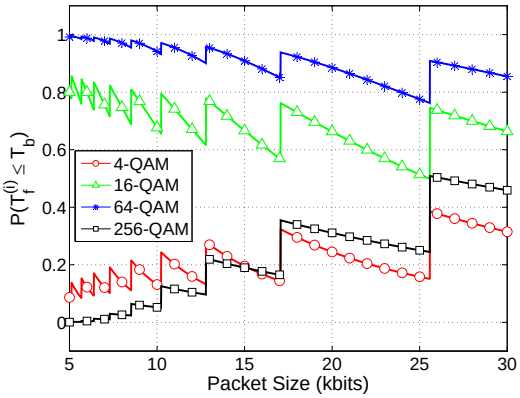
Figure 2.4: The probability of correctly receiving a frame within a time constraint vs. E_s/N_0 ($S_f = 9383\ byte, S_p = 2272\ byte, RTT = 10\ ms, T_b = 100\ ms, C = 512\ Kbps$)

Proper selection of the packet size can improve $F(T_b, i)$. Small packet sizes in SW can severely deteriorate the transmission performance as shown in Figure 2.5a and Figure 2.5b. That is because with small packet sizes the number of required packets per frame is increased which is unfavorable due to the idle time between packet transmissions in SW. The ramp behavior in the plots (e.g. $17 \leq S_p < 25.5$ kbit in Figure 2.5a) is attributed to the truncation function in Equation 2.15. In these ramp intervals, N_p is the same. Hence, $F(T_b, i)$ will decrease as the packet size is increased (larger packet sizes require longer transmission time). Figure 2.5c and Figure 2.5e show significant difference in performance between GBN and SR for 256-QAM when fixed FEC is used. SR ARQ outperforms GBN especially for small packet sizes. That is because transmission efficiency of GBN degrades when RTT $\cdot\, C \cdot log_2 M$ is large relative to Sp as shown in Equation 2.10. Moreover, adaptive FEC improves the performance of the different modulation levels and ARQ schemes. This improvement is significant for 256-QAM with GBN as shown in Figure 2.5d. Adaptive FEC makes P_{c_i} approach 1, i.e. the average number of retransmission of a packet becomes almost one, thus making Go-back-N less disadvantageous when compared to a case with higher average number of retransmissions.

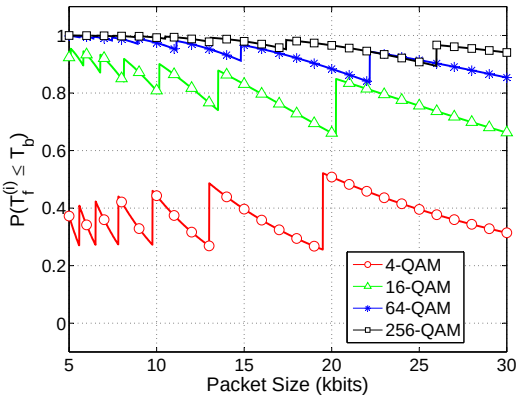


(a) SW with fixed FEC (CR=$\frac{3}{4}$)

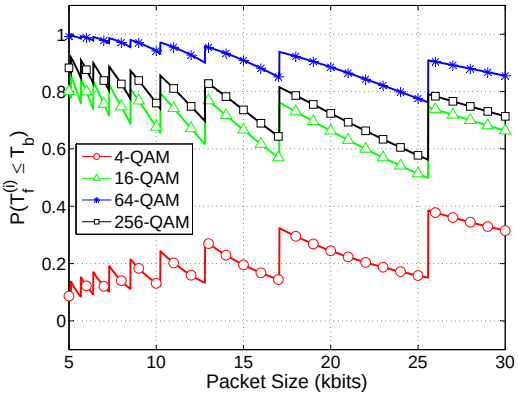
(b) SW with adaptive FEC



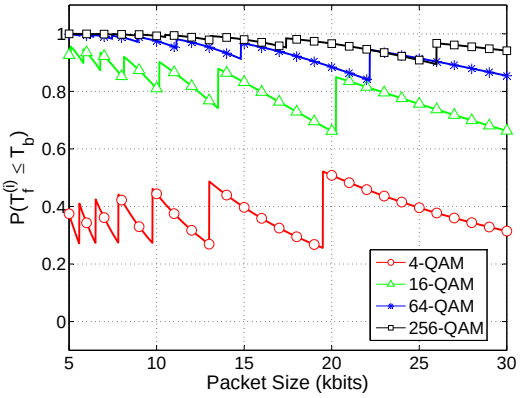
(c) GBN with fixed FEC (CR=$\frac{3}{4}$)



(d) GBN with adaptive FEC



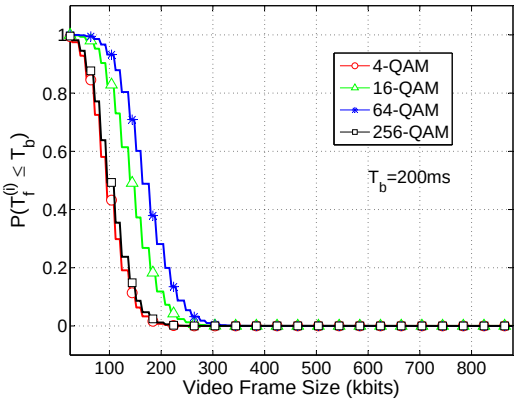
(e) SR with fixed FEC (CR=$\frac{3}{4}$)



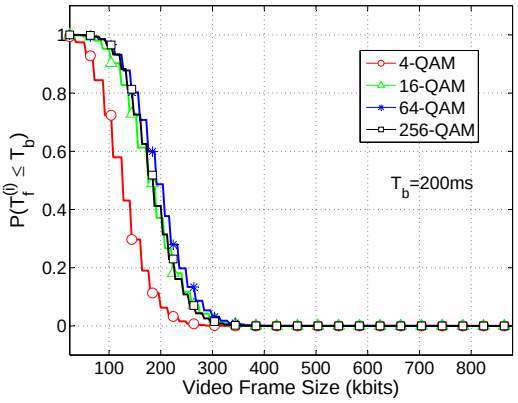
(f) SR with adaptive FEC

Figure 2.5: The probability of correctly receiving a frame within a time constraint vs. the packet size ($S_f = 4797\ byte, RTT = 10\ ms, E_s/N_0 = 19\ dB, T_b = 33\ ms, C = 512\ Kbps$)

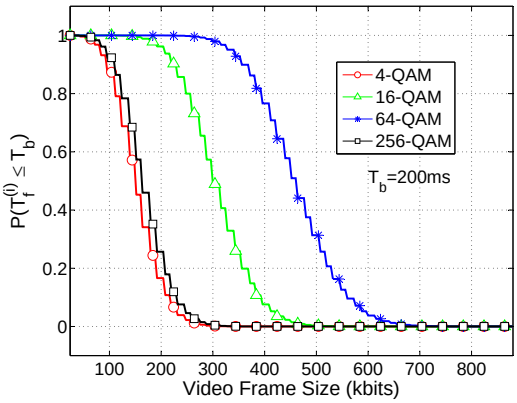
The performance of the modulation levels with different ARQ schemes for different video frame sizes is shown in Figure 2.6. Intuitively, as the frame size is increased, $F(T_b, i)$ is decreased. For Sp=2272 byte, E_s/N_0=19 dB, T_b=200 ms, RTT=10 ms, and C=512 Kbps, the performance of 256-QAM matches the performance of 4-QAM when SW and GBN are used with fixed FEC as shown in Figure 2.6a and Figure 2.6c. Figure 2.6e shows that 256-QAM is capable of better performance with the efficient SR ARQ. Moreover, adaptive FEC improves the performance of the modulation levels and ARQ schemes. Adaptive FEC with GBN or SR considerably enhances the performance of 256-QAM and allow it to maintain high $F(T_b, i)$ for relatively large frame sizes as shown in Figure 2.6d and Figure 2.6f. Moreover, Figure 2.6f shows the effect of T_b on $F(T_b, i)$. Intuitively, for larger T_b (i.e. larger playback buffer occupancy) the probability of timely delivery of video frames increases and the likelihood of playback buffer starvation decreases.



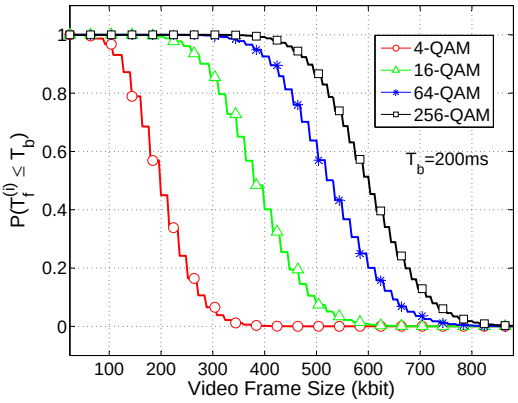
(a) SW with fixed FEC (CR=$\frac{3}{4}$)



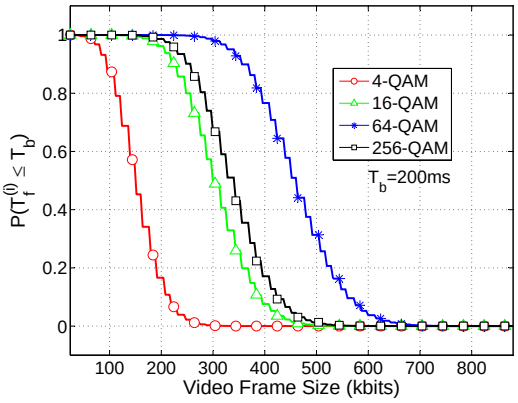
(b) SW with adaptive FEC



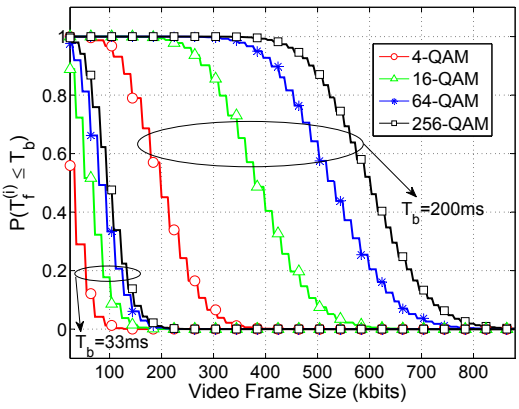
(c) GBN with fixed FEC (CR=$\frac{3}{4}$)



(d) GBN with adaptive FEC



(e) SR with fixed FEC (CR=$\frac{3}{4}$)



(f) SR with adaptive FEC

Figure 2.6: The probability of correctly receiving a frame within a time constraint vs. the frame size ($S_p = 2272\ byte, RTT = 10\ ms, E_s/N_0 = 19\ dB, C = 512\ Kbps$)

2.3 Proposed Video Quality Assessment Scheme

We propose a video quality assessment system that is capable of estimating the PSNR of the received video frames without having the original video available as a reference. Therefore, it provides a better alternative to the conventional PSNR calculation approach. The proposed system takes into account two new temporal quality measures, namely, the skip length and the inter-starvation distance. These two measures are combined with additional bitstream information to construct distinctive feature vectors. A two-tier polynomial classifier is then used to predict the PSNR values based on the extracted features at the video decoder.

2.3.1 Predictors' Extraction

Predictors' extraction is the process of extracting information from the received bitstream to constitute distinctive feature vectors. These feature vectors facilitate the estimation of PSNR of decoded/concealed video frames without the need of the reference video. In the event of frame losses, error concealment is employed to reconstruct estimates of the lost frames. To monitor the quality of the reconstructed video, predictors/features are extracted and used to predict PSNR values of not only the concealed frames but also their dependent frames. Hence, in this work, two sets of predictors are considered. The first set caters for the prediction of PSNR of lost frames, while the second set caters for the prediction of PSNR of frames that are correctly received but reconstructed from concealed frames. We refer to the two cases as Case 1 and Case 2 respectively.

Table 2.1 lists the predictors for the two different cases. In Case 1, entire frame losses are assumed. Therefore, to help us predict the PSNR quality of a lost frame, we extract information from the preceding decodable (correctly received) frame. Figure 2.7 illustrates the definition of the SL, ISD, L, and D parameters.

Case 1 Predictors	Case 2 Predictors
Skip Length (SL)	Skip Length (SL)
Inter-starvation Distance (ISD)	Inter-starvation Distance (ISD)
Frame Location in the Starvation Interval (L)	Frame Distance from the Starvation Interval (D)
PSNR at the Encoder of the Preceding Decodable Frame	PSNR at the Encoder
Motion Vectors ($MV_{x,y}$) Means of the Preceding Decodable Frame	Motion Vectors ($MV_{x,y}$) Means
Motion Vectors ($MV_{x,y}$) Standard Deviations of the Preceding Decodable Frame	Motion Vectors ($MV_{x,y}$) Standard Deviations
Percentage of Intra-coded Macroblocks in the Preceding Decodable Frame	Percentage of Intra-coded Macroblocks
Size in bits of the Preceding Decodable Frame	Frame Size in bits
Quantization Parameter (QP) of the Preceding Decodable Frame	Quantization Parameter (QP)

Table 2.1: Video quality predictors

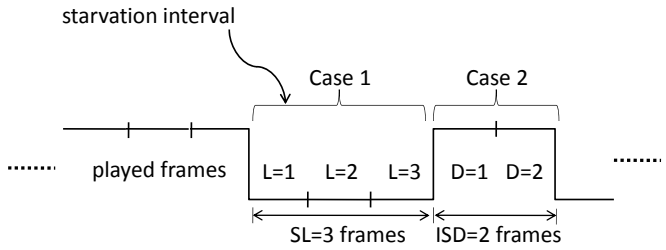


Figure 2.7: Illustration of the SL, ISD, L, and D parameters

2.3.2 PSNR Prediction

Reduced polynomial networks have been recently introduced in [34]. Such polynomial networks can be used to achieve a nonlinear mapping between the extracted feature vectors and the true PSNR. In the training phase, the feature vectors are expanded

using polynomial expansion [35] into a given order. The objective of this expansion is to improve the separation between classes in the expanded vector space [36]. The model parameters are estimated through multivariate regression which entails minimizing the L2 norm of the model's prediction error. The L2 norm of a vector $x = [x_1\ x_2\ ...\ x_n]$ is given by:

$$\|x\| = \sqrt{\sum_{k=1}^{n} x_k^2}. \tag{2.17}$$

In our simulation, the feature vectors were split into 2 halves. One half is designated as a training dataset used for model estimation and the other half is designated as a testing set to validate the model.

In this work we also propose a two-tier PSNR estimation architecture as illustrated in Figure 2.8. Basically, in the first tier, the training feature vectors are expanded into a given polynomial order. The model parameters or polynomial weights are calculated using the mentioned vectors and the true PSNR. The model parameters are then used to estimate the PSNR of both the train and test feature vector sets. In the second tier, the predicted PSNR of a previous frame is concatenated with the training and testing feature vectors of the current frame. Hence an additional predictor or variable is added to the set of feature variables. The training process is repeated where the training set is expanded and the model parameters are regenerated. The final PSNR estimate is obtained based on testing scores calculated as the inner product of the regenerated model parameters with the expanded test feature vectors. The reported experimental results show that such an architecture improves the accuracy of the PSNR estimation process.

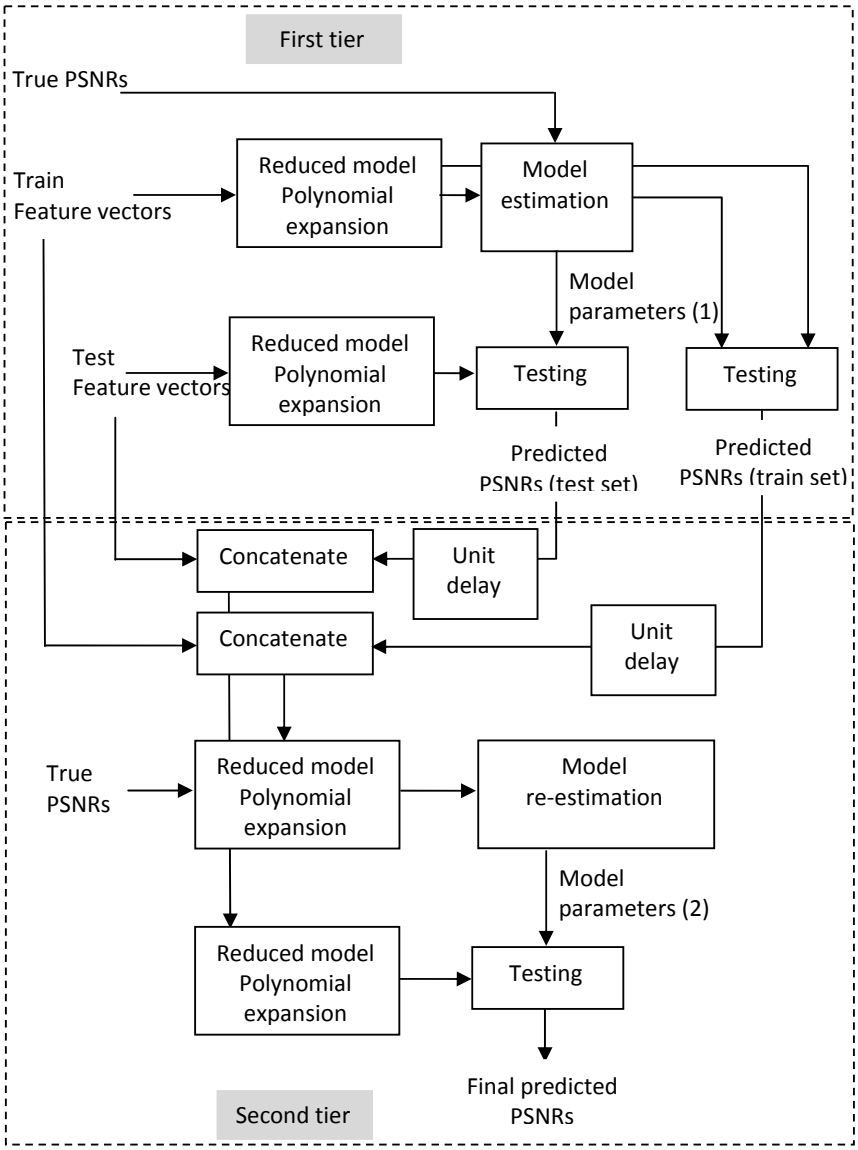


Figure 2.8: Two-tier PSNR identification block diagram

Chapter 3

Results

In this chapter, we show our simulation results for the multi-level adaptive video streaming methods described in Chapter 2. In addition, experimental results for the proposed no-reference video quality estimation system are also presented.

3.1 Adaptive Playback and Bitstream Switching

In this section, we demonstrate the effect of adaptive playback and bitstream switching on the performance of a video streaming system when the two techniques are implemented separately and jointly. An event-based simulator (MATLAB SimEvents) was used to simulate the video streaming system. The"news" video sequence was encoded using different quantization parameters (QP) [37] such that multiple bitstreams with different average encoding bitrates are generated. Table 3.1 shows the QP values and the generated streams bitrates. These bitstreams are stored and made available for the transmitter to dynamically switch between them depending on the channel condition or feedback information from the receiver. The video stream was encoded using H.264/AVC JM Reference Software [26]. The highest average bitrate of the generated bitstreams was made equal to the offered channel capacity. In addition, the encoding sequence type was $IBPBP...$ with a Group of Picture (GoP)=16. The forward transmission channel was assumed a varying two-state channel. Following a continuous-time Markov chain, the probability of channel being in good state is denoted by P_l. The packet error rate ($P_{e_i} = 1 - P_{c_i}$) during a good state is P_{e_0} and during a bad state is P_{e_1}. Stop-and-wait ARQ is considered to control the

transmission of video frames. Reliable data transport protocol is assumed where acknowledgment (ACK) and negative acknowledgment (NACK) messages are sent over a reliable reverse channel. The simulation parameters are given in Table 3.2. B_{pr} is the preroll decoder buffer state after which playback starts.

Stream #	1	2	3	4	5	6
QP	28	29	30	31	32	34
Average Bitrate (kbit/s)	293.10	258.79	232.30	210.61	184.65	148.35

Table 3.1: Average bitrates of the encoded bistreams for different QP values

Parameter	Value
f_n	30 fps
P_l	0.5
P_{e_0}	0.01
P_{e_1}	0.1
$B_{th} = B_H$	10 frames
$B_{pr} = B_L$	5 frames

Table 3.2: Simulation parameters

In Figure 3.1, the performances of non-adaptive streaming and adaptive streaming are compared in terms of the playback buffer occupancy under the same channel realization. Figure 3.1a shows the playback buffer occupancy with starvation events when neither adaptive playback nor bitstream switching were applied. We then implemented adaptive playback and bitstream switching separately. The buffer starvation events were reduced but not eliminated as shown in Figure 3.1b and Figure 3.1c. Finally, bitstream switching and adaptive playback were jointly implemented and starvation events were avoided as shown in Figure 3.1d.

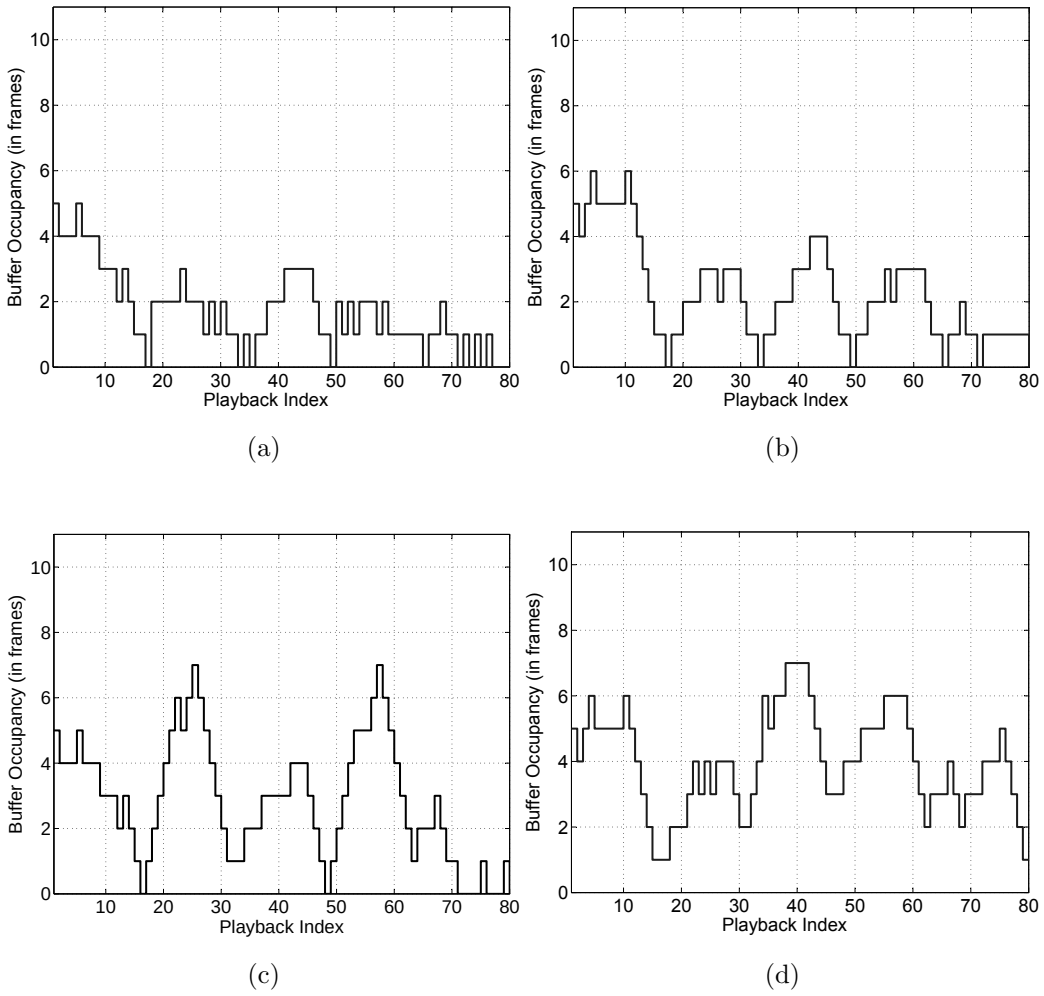


Figure 3.1: Playback buffer occupancy for the "news" sequence (a) with no adaptive playback and no bitstream switching, (b) with adaptive playback and no bitstream switching, (c) with bitstream switching and no adaptive playback, (d) with adaptive playback and bitstream switching

The bitstream switching mechanism successfully eliminated playback buffer starvation. However, reduction in the decoded video quality was introduced. Figure 3.2 compares the PSNR of the highest quality bitstream available at the transmitter with the PSNR of the received video stream when bitstream switching is implemented. Switching between bitstreams was only allowed at I frames (at intervals of 16 frames) to avoid the error drift problem. In conclusion, it can be argued that the overall performance of the video streaming system was improved with the adaptive

playback and bitstream switching mechanism. That is because slight degradation in video quality is usually considered less annoying than interruption in video playback.

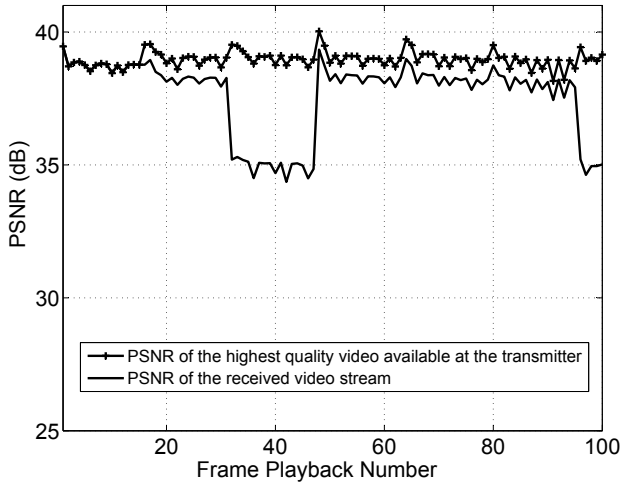


Figure 3.2: PSNR of the highest bitrate video available at the transmitter vs. PSNR of the received video sequence when adaptive playback and bitstream switching are jointly applied for the "news" sequence

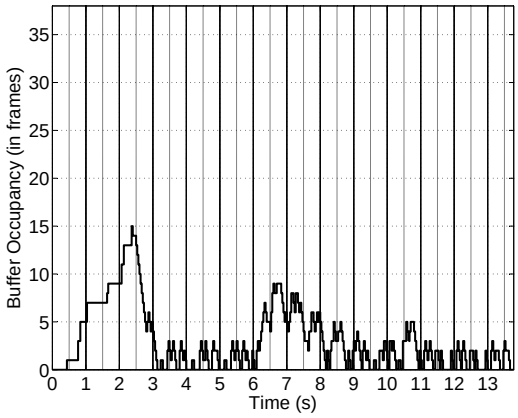
3.2 Scalable Coding with Adaptive Modulation and Channel Coding

MATLAB (SimEvents) was used to test our multi-level adaptive algorithm described in Section 2.2. In our simulations, we considered two video sequences, the "football" sequence and the "Harry Potter" HD sequence. The "football" sequence is a short sequence (260 frames) which was obtained in YUV format. On the other hand, the "Harry Potter" HD sequence is a long sequence (86384 frames). Its encoding trace file was obtained from [33].

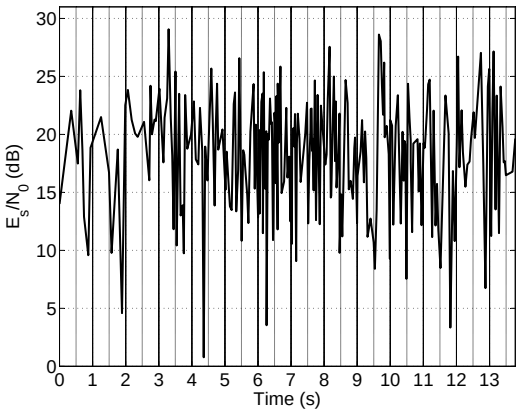
The "football" video sequence was encoded with 1 base layer and 10 quality enhancement layers using the Medium Grain Scalability (MGS) option in the JSVM H.264/SVC Reference Software [9, 38]. This option encodes a video frame and arranges the frame bits in a way that allows discarding parts of the video frame bits (i.e. enhancement layers) while the truncated frame will still be decodable. A Rayleigh fading channel with an exponentially distributed E_s/N_0 that changes per video frame was assumed. Go-back-N ARQ and Fixed FEC (code rate $CR = 3/4$) were used. The following figures show the performances of different fixed QAM modulation levels in addition to the performance of adaptive QAM. The performance is shown in terms of decoder buffer occupancy, percentage of video frame scaling/truncation, and decoded video PSNR. Every time a frame is to be transmitted, the transmitter computes the probability of correctly receiving the frame before its playback deadline/budget time. The transmitter scales down, if necessary, the video frame by a scaling increment $\theta = 0.95$ ($S_f^{(new)} = \theta S_f$) until a high probability is met ($\omega = 0.9$). In the adaptive QAM scheme, before scaling a frame, the transmitter computes the probability of successful delivery for the different modulation levels and selects the level that achieves the highest probability. Nevertheless, if none of the modulation levels could achieve a high probability, scaling is then implemented as necessary. encodes a video frame and arranges the frame bits in a way that allows discarding parts of the video frame bits (i.e. enhancement layers) while the truncated frame will still be decodable. A Rayleigh fading channel with an exponentially distributed E_s/N_0

that changes per video frame was assumed. Go-back-N ARQ and Fixed FEC (code rate $CR = 3/4$) were used. The following figures show the performances of different fixed QAM modulation levels in addition to the performance of adaptive QAM. The performance is shown in terms of decoder buffer occupancy, percentage of video frame scaling/truncation, and decoded video PSNR. Every time a frame is to be transmitted, the transmitter computes the probability of correctly receiving the frame before its playback deadline/budget time. The transmitter scales down, if necessary, the video frame by a scaling increment $\theta = 0.95$ ($S_f^{(new)} = \theta S_f$) until a high probability is met ($\omega = 0.9$). In the adaptive QAM scheme, before scaling a frame, the transmitter computes the probability of successful delivery for the different modulation levels and selects the level that achieves the highest probability. Nevertheless, if none of the modulation levels could achieve a high probability, scaling is then implemented as necessary.

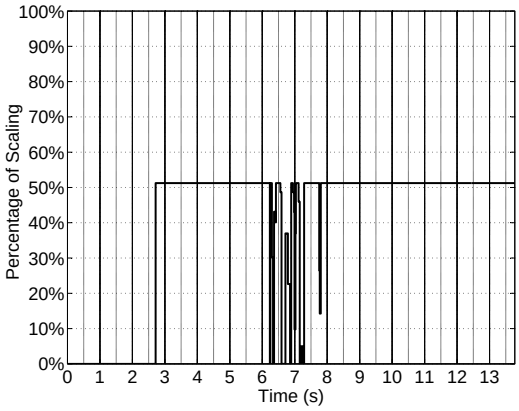
Figure 3.3 describes the video streaming system performance when 4-QAM is used. The preroll threshold is set to 15 frames. During the preroll period scaling is not implemented. We see that the occupancy builds up until there are 15 frames in the buffer. Clearly, this is a very slow start (2.4 s) for only 15 frames. This indicates the poor data rate when low level modulation (4-QAM) is used. When buffer occupancy reaches 15 frames, playback starts and the buffer is drained at 30 fps. When the buffer started to approach starvation at t=2.7s, scaling was invoked. Nevertheless, the frame arrival rate could not keep up with the playback rate and starvation could not be avoided even though maximum scaling was in effect. Scaling is limited to 50% which is approximately the portion of all enhancement layers in the ecncoded frames. Within the period $6.3 - 7.5s$ the buffer occupancy started to increase and scaling was not needed at some instants. Notice that during this period the video frame sizes are relatively small as shown in Figure 3.3d which allowed the buffer occupancy to slightly increase.



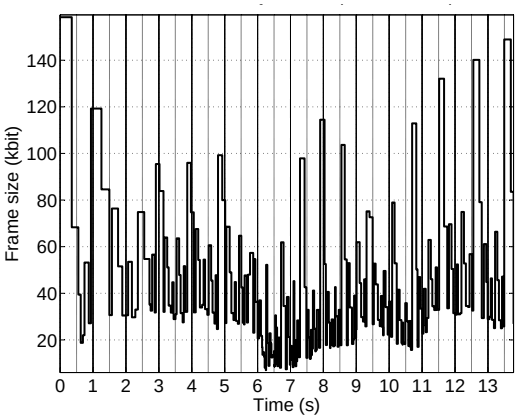
(a) Decoder buffer occupancy



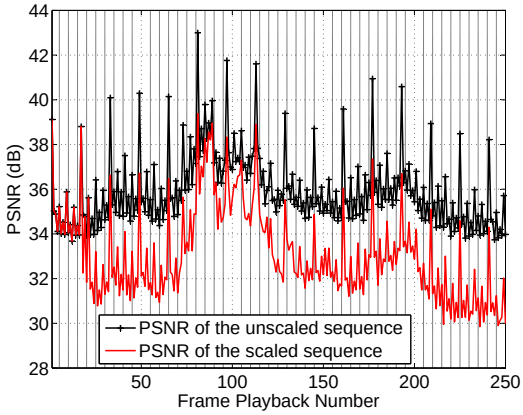
(b) Channel SNR per symbol



(c) Percentage of scaling



(d) Frame sizes



(e) Decoded video PSNR

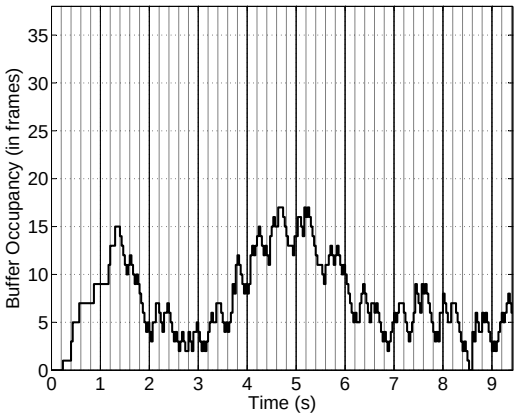
Figure 3.3: Performance of 4-QAM with GBN and fixed FEC for the "football" sequence (Average E_s/N_0=18 dB)

The scaling affected the quality of the decoded video as shown in Figure 3.3e. For example, Figure 3.4 illustrates the visual quality differenece between the unscaled and scaled frame 216. The quality degredation in 3.4b can be observed in the blurry grass and the writing on the back of player 82.

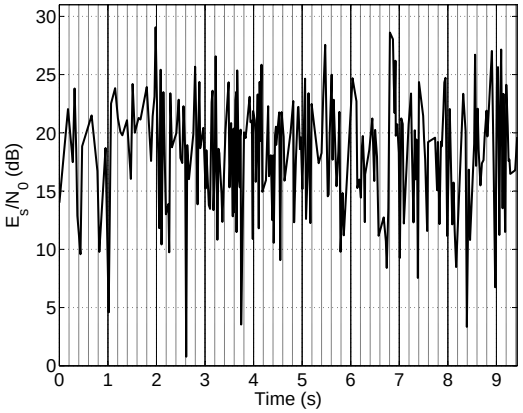
(a) Unscaled (b) Scaled

Figure 3.4: Visual quality difference between the unscaled and scaled frame 216 when 4-QAM is used

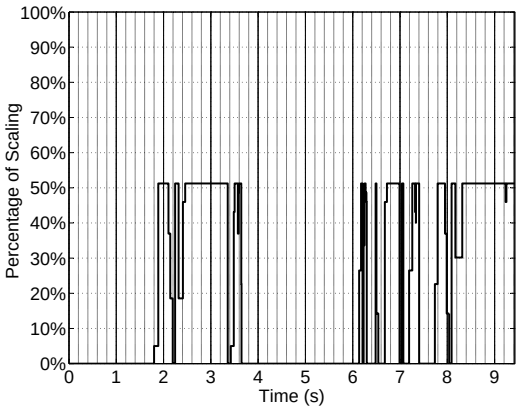
The following figures (Figure 3.5, Figure 3.6, and Figure 3.7) show the performance of the streaming system when other fixed modulation levels are used while Figure 3.8 shows the performance when adaptive modulation is used. Moreover, Figure 3.9 compares the modulation level performances in terms of the amount of scaling, i.e. the average reduction percentage in the original frame sizes and the percentage of the truncated frames with respect to the total number of transmitted ones. The performances are also compared in terms of the skip length, and the inter-starvation distance statistics. It can be seen that adaptive modulation outperforms the fixed modulation levels. Adaptive modulation managed to eliminate starvation and reduced the amount of required scaling, hence, enhancing the temporal and spatial quality of the decoded video.



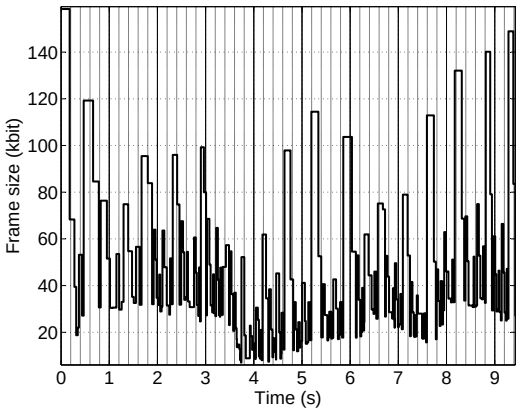
(a) Decoder buffer occupancy



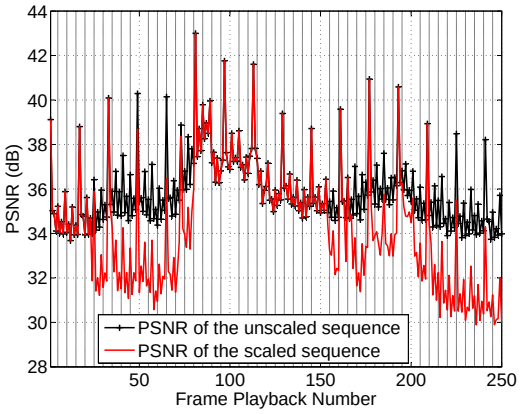
(b) Channel SNR per symbol



(c) Percentage of scaling

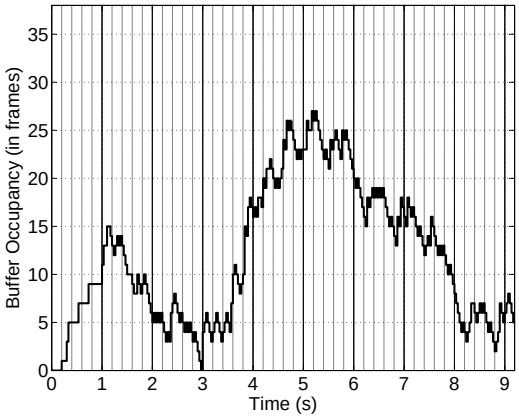


(d) Frame sizes

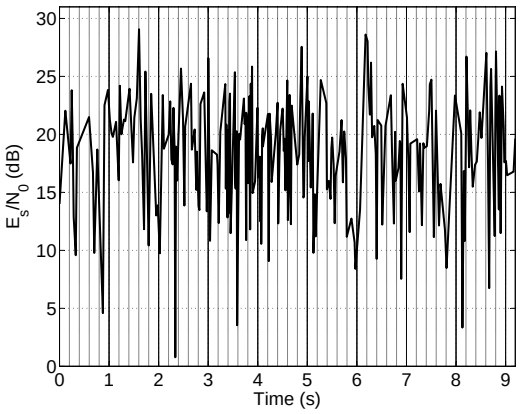


(e) Decoded video PSNR

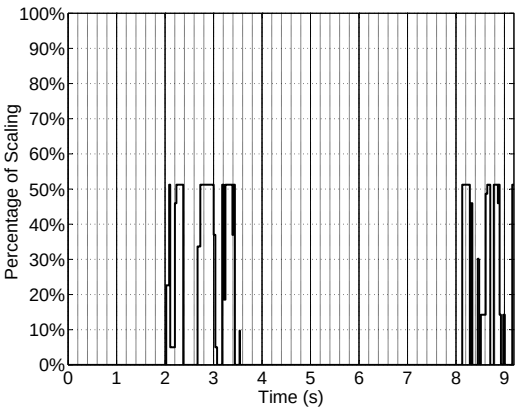
Figure 3.5: Performance of 16-QAM with GBN and fixed FEC for the "football" sequence (Average E_s/N_0=18 dB)



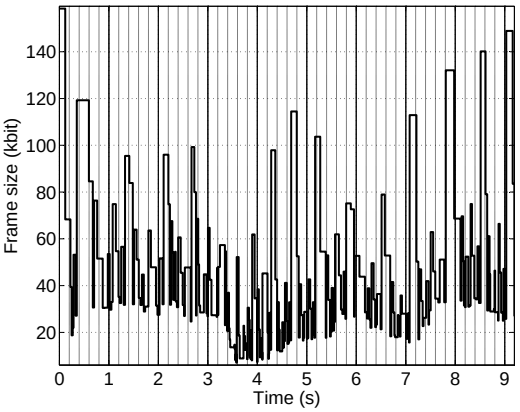
(a) Decoder buffer occupancy



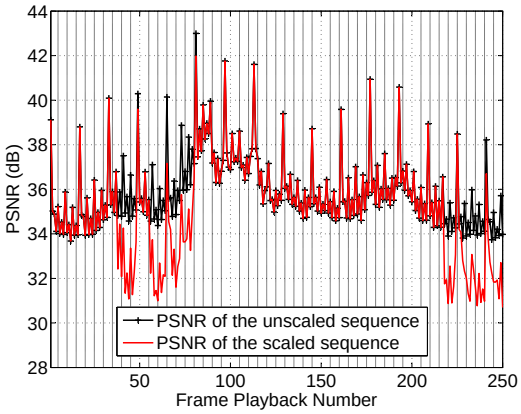
(b) Channel SNR per symbol



(c) Percentage of scaling

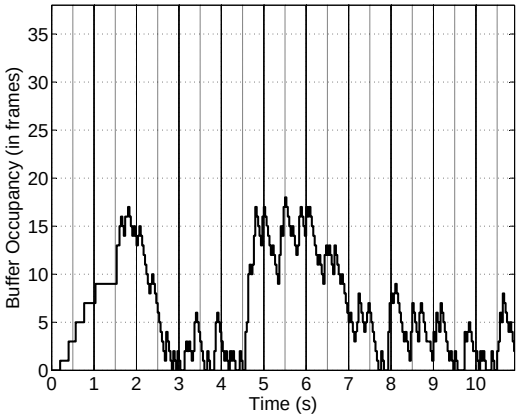


(d) Frame sizes

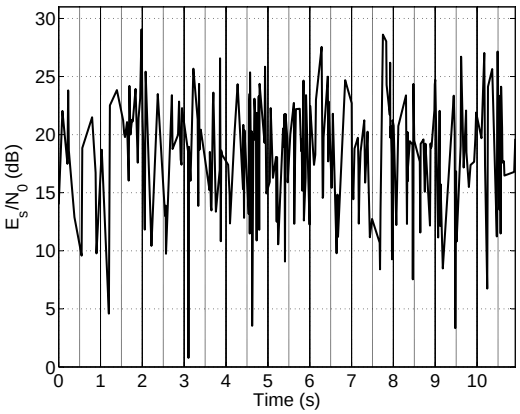


(e) Decoded video PSNR

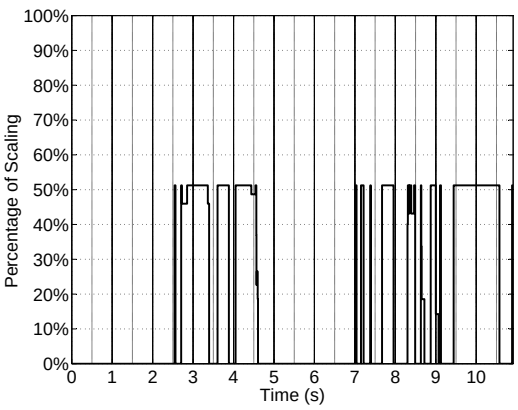
Figure 3.6: Performance of 64-QAM with GBN and fixed FEC for the "football" sequence (Average E_s/N_0=18 dB)



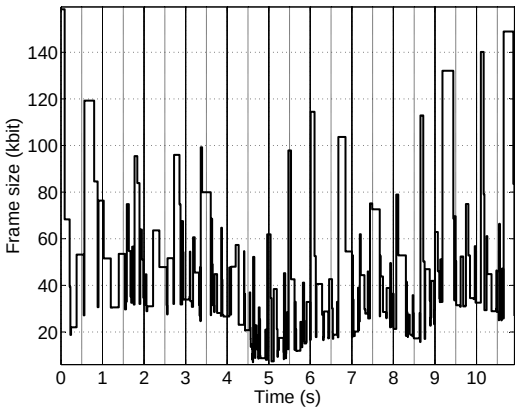
(a) Decoder buffer occupancy



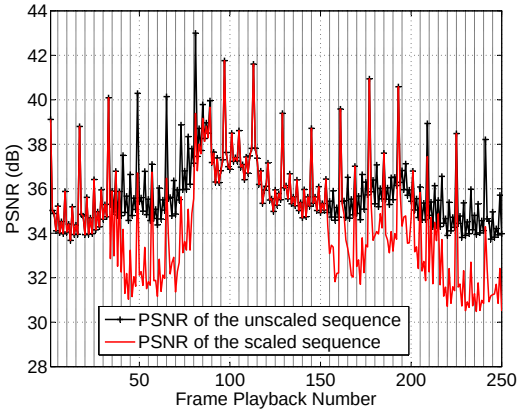
(b) Channel SNR per symbol



(c) Percentage of scaling

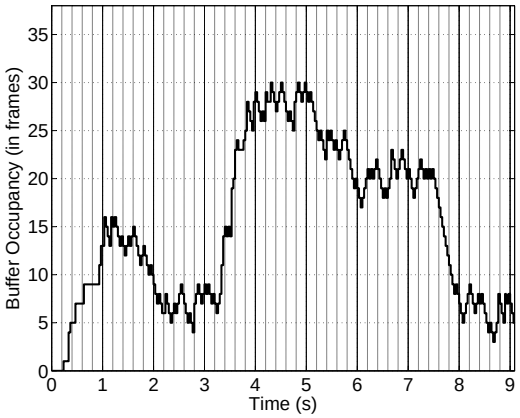


(d) Frame sizes

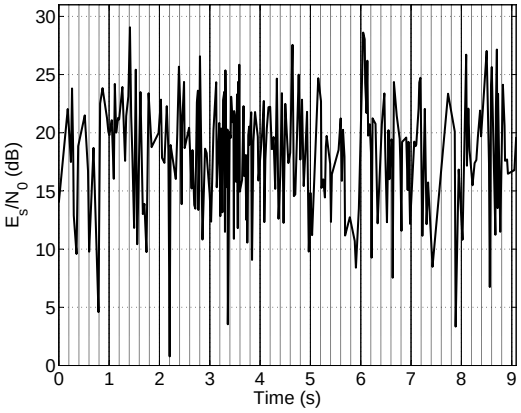


(e) Decoded video PSNR

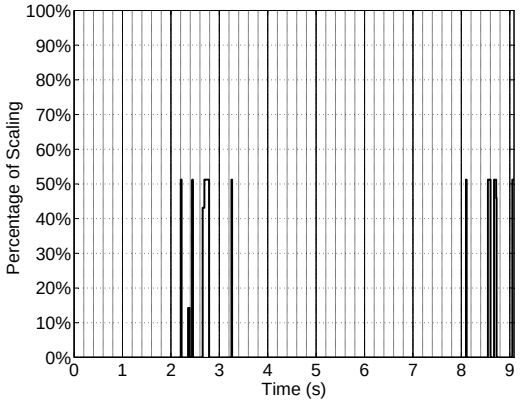
Figure 3.7: Performance of 256-QAM with GBN and fixed FEC for the "football" sequence (Average E_s/N_0=18 dB)



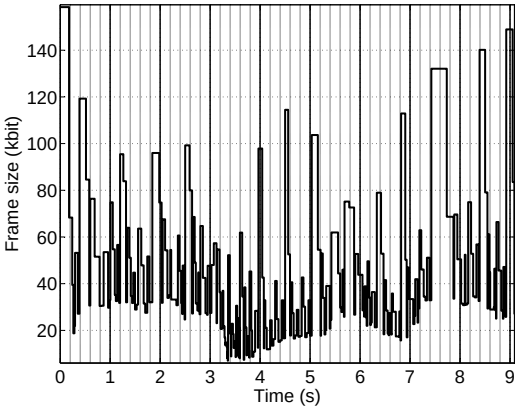
(a) Decoder buffer occupancy



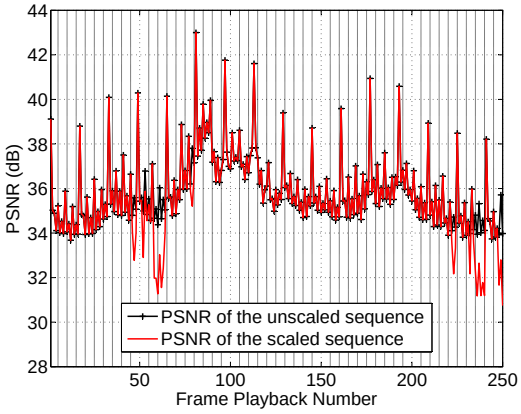
(b) Channel SNR per symbol



(c) Percentage of scaling

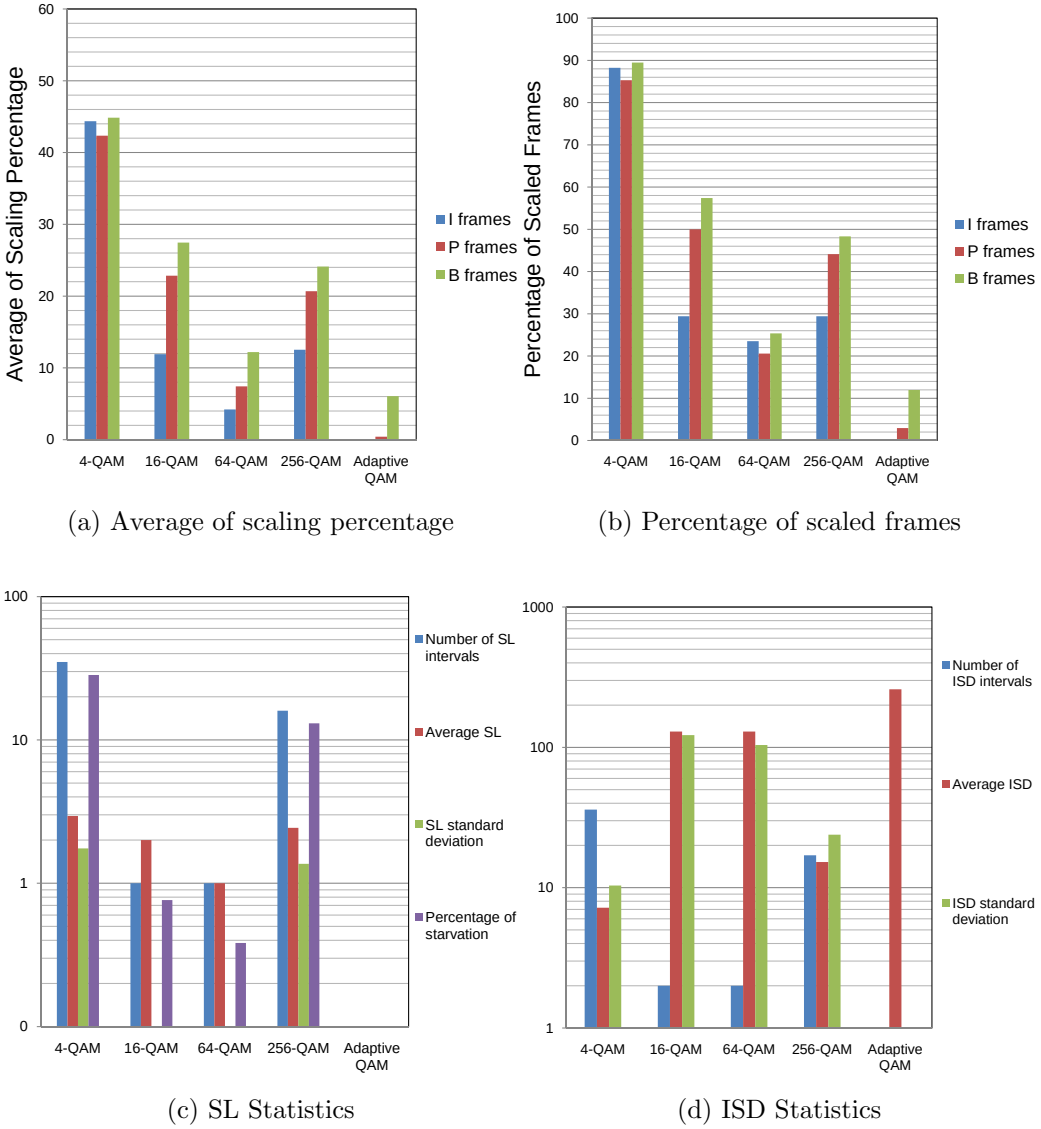


(d) Frame sizes



(e) Decoded video PSNR

Figure 3.8: Performance of adaptive QAM with GBN and fixed FEC for the "football" sequence (Average E_s/N_0=18 dB)



(a) Average of scaling percentage

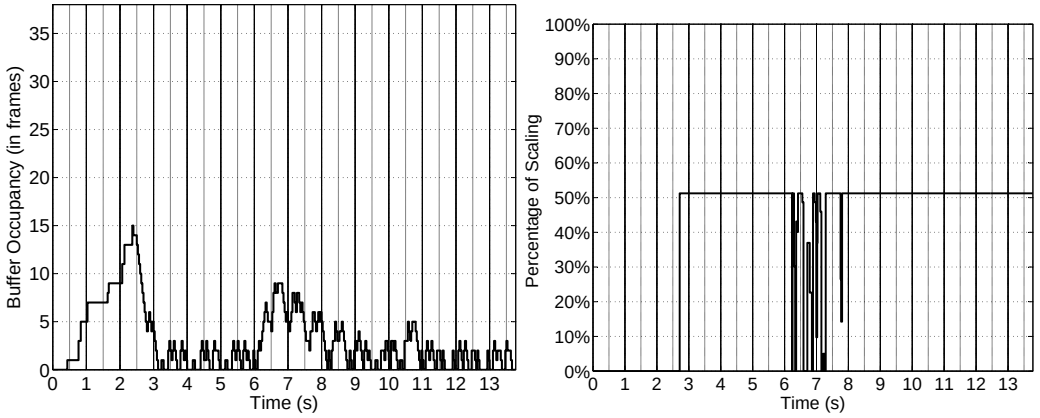
(b) Percentage of scaled frames

(c) SL Statistics

(d) ISD Statistics

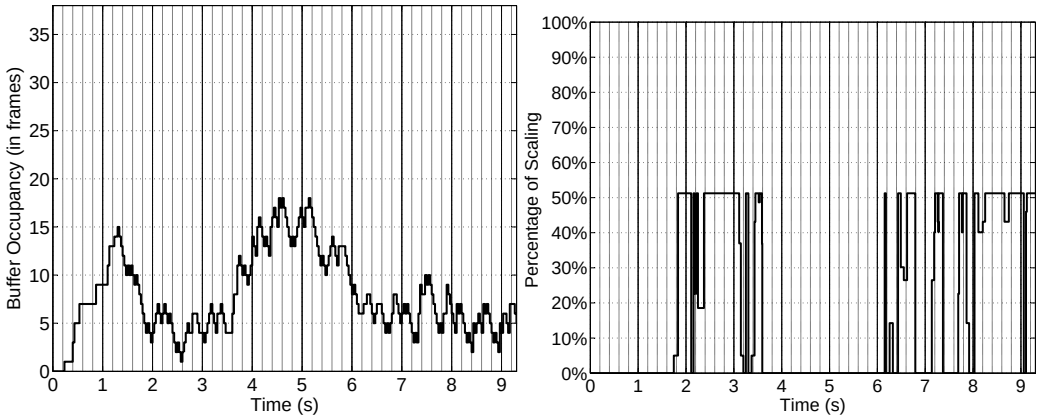
Figure 3.9: Performance of different modulation levels with GBN ARQ and fixed FEC for the "football" sequence (C=256Kbps, RTT=10ms)

Figure 3.10, Figure 3.11, Figure 3.12, Figure 3.13, and Figure 3.14 show the performance of the "football" streaming system for a different channel realization with higher SNR per symbol (Average E_s/N_0=20 dB). It can be noticed that 4-QAM performance did not improve due to its data rate limitation. On the other hand, higher modulation levels performances improved especially for 256-QAM.



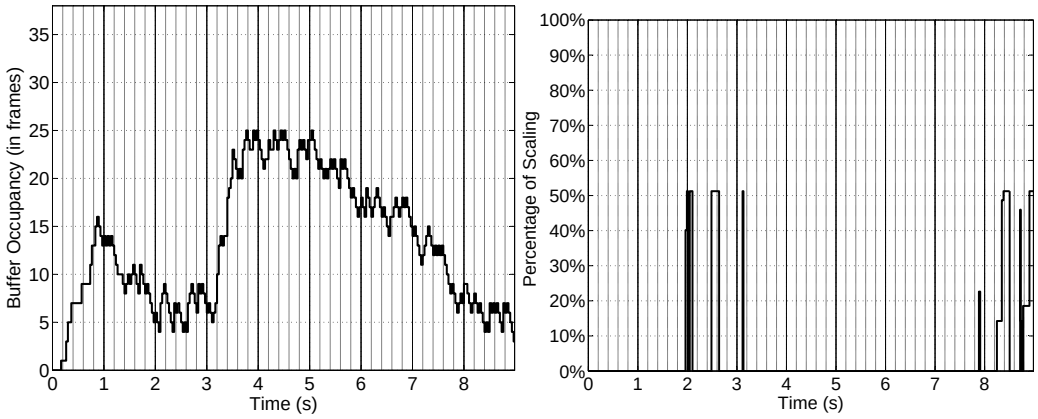
(a) Decoder Buffer Occupancy (b) Percentage of Scaling

Figure 3.10: Performance of 4-QAM with GBN and fixed FEC for the "football" sequence (Average E_s/N_0=20 dB)



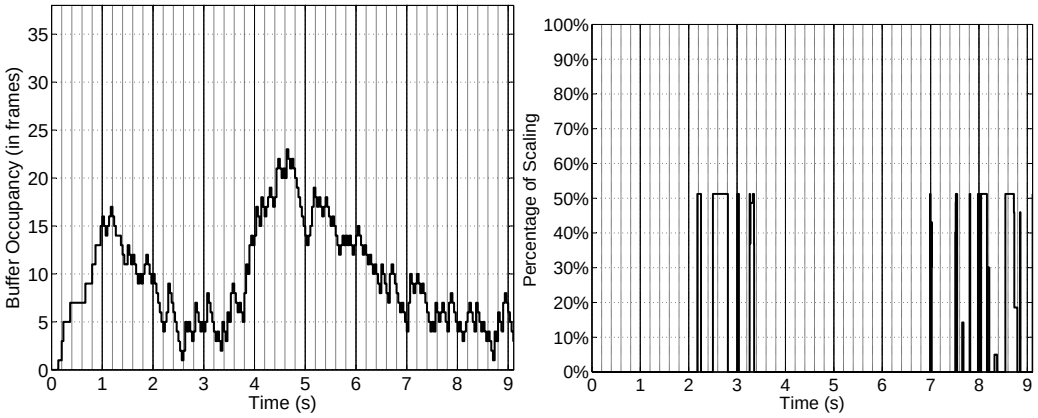
(a) Decoder Buffer Occupancy (b) Percentage of Scaling

Figure 3.11: Performance of 16-QAM with GBN and fixed FEC for the "football" sequence (Average E_s/N_0=20 dB)



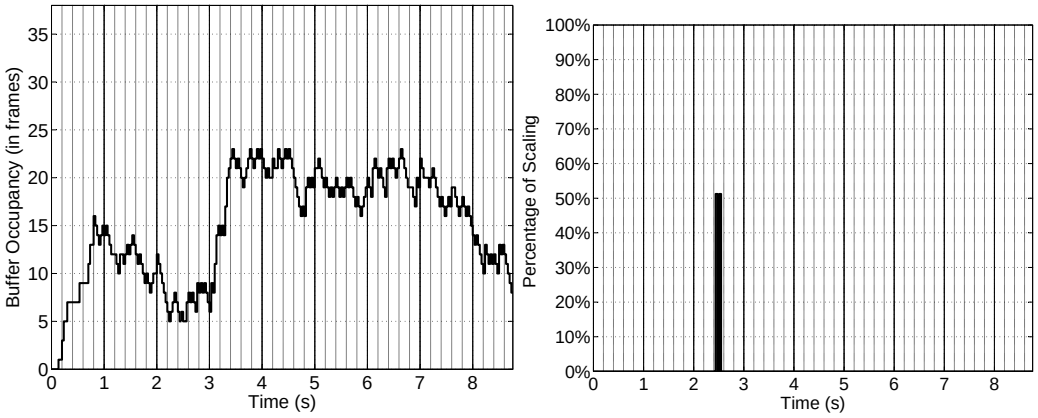
(a) Decoder Buffer Occupancy (b) Percentage of Scaling

Figure 3.12: Performance of 64-QAM with GBN and fixed FEC for the "football" sequence (Average E_s/N_0=20 dB)



(a) Decoder Buffer Occupancy (b) Percentage of Scaling

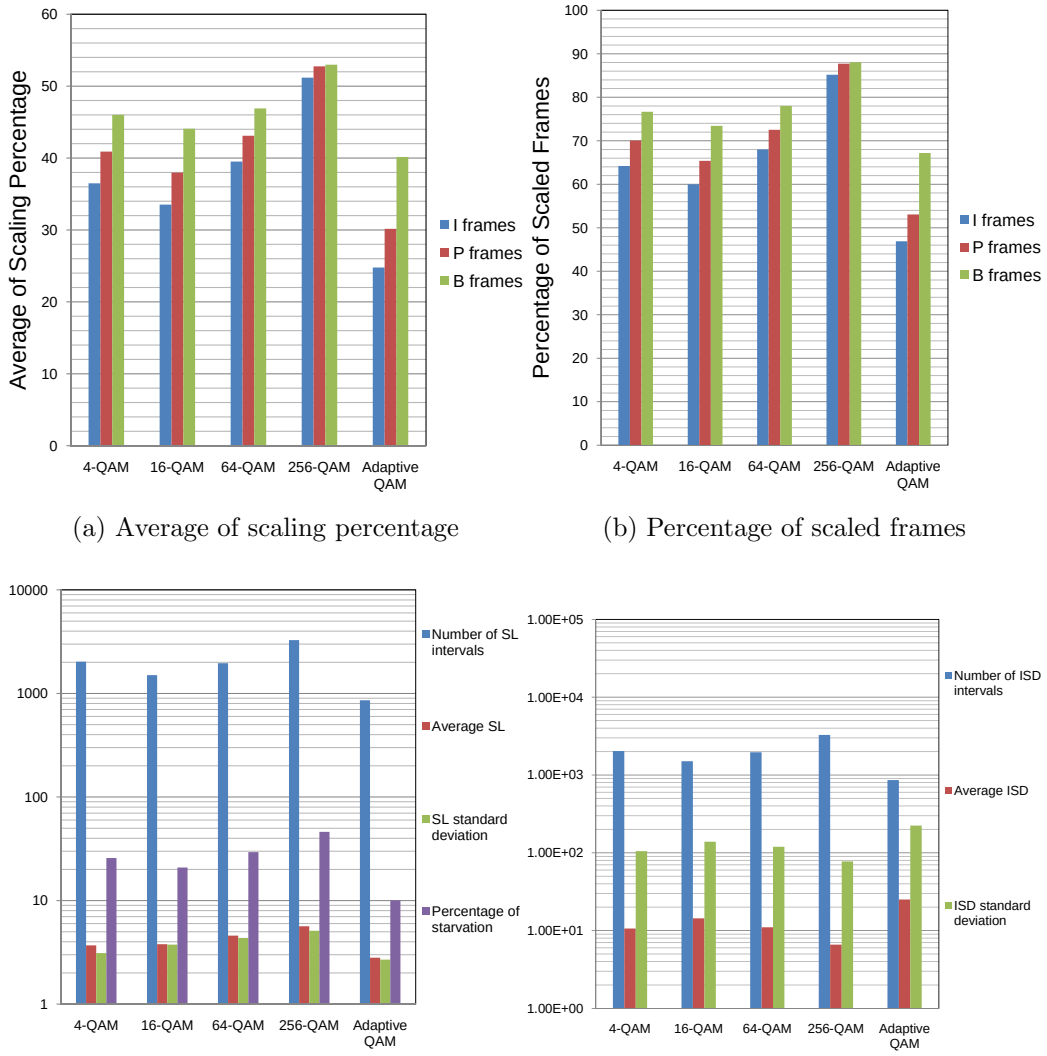
Figure 3.13: Performance of 256-QAM with GBN and fixed FEC for the "football" sequence (Average E_s/N_0=20 dB)



(a) Decoder Buffer Occupancy (b) Percentage of Scaling

Figure 3.14: Performance of adaptive QAM with GBN and fixed FEC for the "football" sequence (Average E_s/N_0=20 dB)

The following figures (Figure 3.15, Figure 3.16, Figure 3.17, and Figure 3.18) compare the performances of the different QAM schemes for the first 15 minutes of the "Harry Potter" HD sequence. The comparison is in terms of the average applied scaling, percentage of scaled frames, and the SL and ISD statistics. The simulation was performed with the SW ARQ and the GBN ARQ. Each ARQ scheme was combined with fixed FEC and adaptive FEC for comparison. Adaptive FEC provides considerable performance improvement for all modulation levels. Moreover, adaptive modulation provides significant performance enhancement especially when fixed FEC is employed. For GBN with adaptive FEC, it can be noticed that the performance of 256-QAM is the best and matches the performance of adaptive QAM.



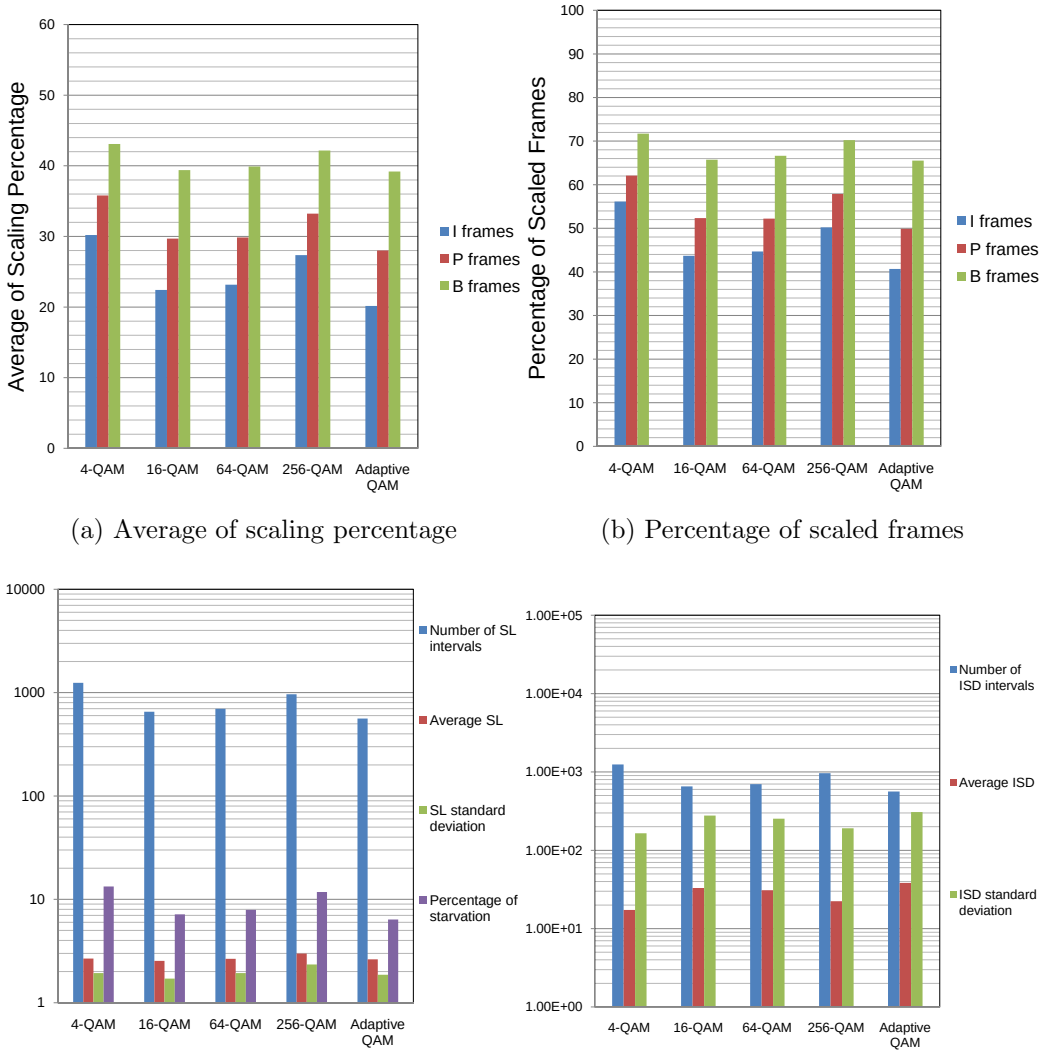
(a) Average of scaling percentage

(b) Percentage of scaled frames

(c) SL Statistics

(d) ISD Statistics

Figure 3.15: Performance of different modulation levels with SW ARQ and fixed FEC for the "Harry Potter" HD sequence (C=1Mbps, RTT=10ms)



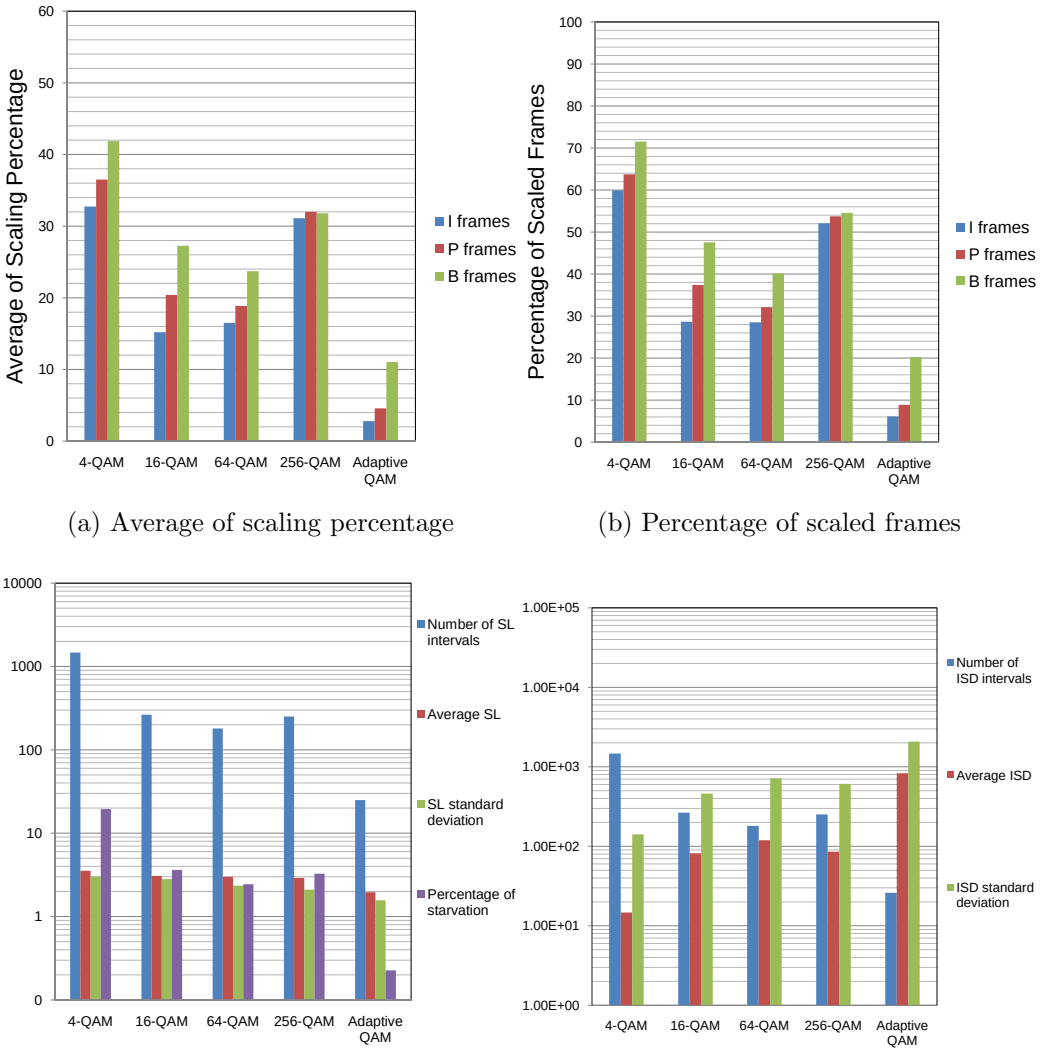
(a) Average of scaling percentage

(b) Percentage of scaled frames

(c) SL Statistics

(d) ISD Statistics

Figure 3.16: Performance of different modulation levels with SW ARQ and adaptive FEC for the "Harry Potter" HD sequence (C=1Mbps, RTT=10ms)



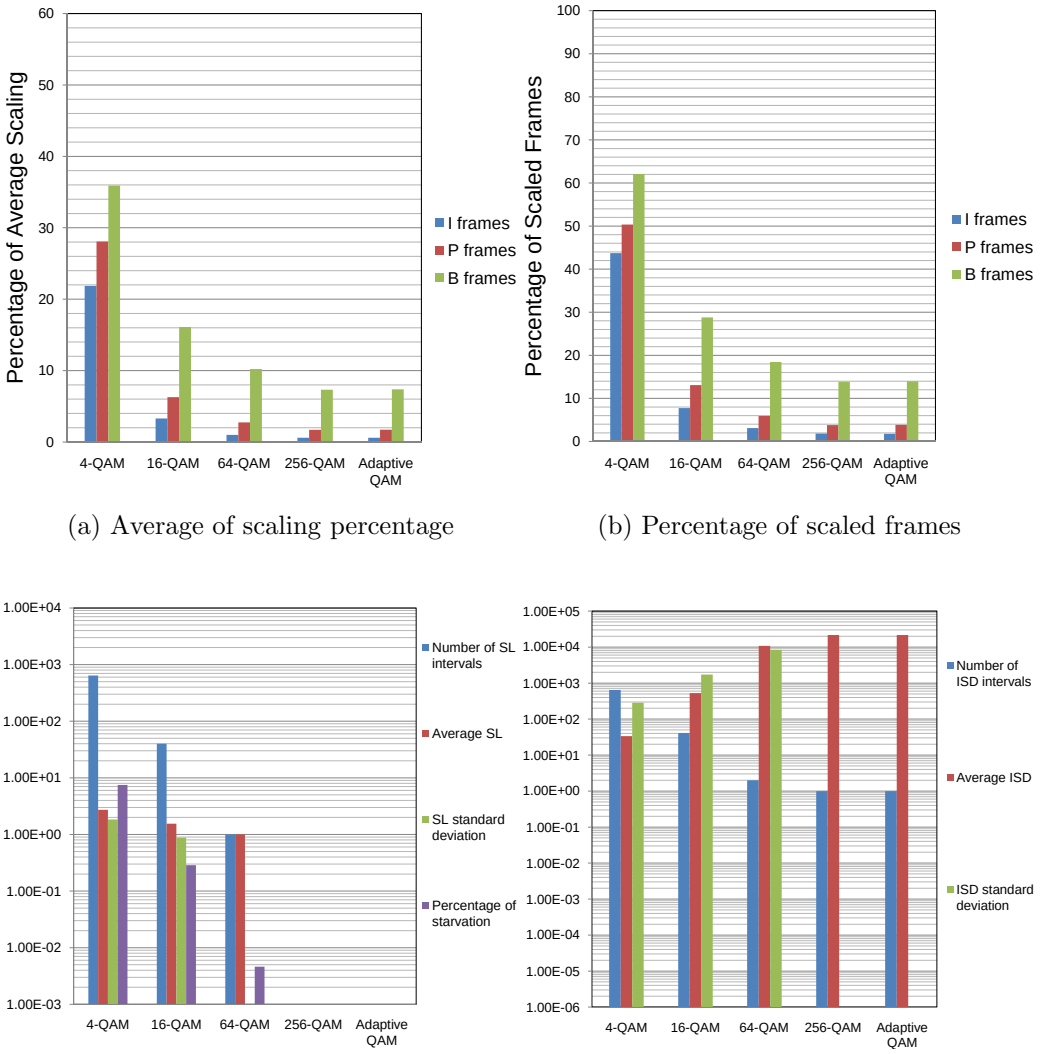
(a) Average of scaling percentage

(b) Percentage of scaled frames

(c) SL Statistics

(d) ISD Statistics

Figure 3.17: Performance of different modulation levels with GBN ARQ and fixed FEC for the "Harry Potter" HD sequence (C=512Kbps, RTT=10ms)



(a) Average of scaling percentage

(b) Percentage of scaled frames

(c) SL Statistics

(d) ISD Statistics

Figure 3.18: Performance of different modulation levels with GBN ARQ and adaptive FEC for the "Harry Potter" HD sequence (C=512Kbps, RTT=10ms)

3.3 PSNR Prediction

In this section, the "football" video sequences is used in generating the experimental results, namely, the total number of frames in the sequence is 250. The sequence is CBR compressed using AVC. A GoP size of 16 is used without bidirectionally predicted frames. After encoding the sequence, one frame per GoP was dropped to simulate frame loss. At the receiver's side, the lost frames are concealed by copying the MVs of the previous frame. The task at the receiver is then to estimate the PSNR of both the lost frames (referred to as Case 1) and the PSNR of correctly received frames but reconstructed from lost concealed frames (referred to as Case 2).

The skip length and inter-starvation distance can be measured at the receiver. Also note that the other features required for PSNR prediction are available from the bitstream and can be extracted at the receiver. For the purpose of simulation results, we use 50% of the feature vectors to generate the model parameters and the rest of the vectors are used for testing and validation. Note that the testing feature vector set is unseen by the model which makes the PSNR prediction more realistic.

In the following experiment we report the correlation factor and the Mean Absolute Difference (MAD) between the predicted and the true PSNRs. The results are shown in Figures 3.19 and 3.20. Figure 3.19 shows that the PSNR of lost frames at the receiver's side can be predicted with a MAD of 1.78 dB. The standard deviation of the prediction error is 0.96 dB. Figure 3.20 shows that predicting the PSNR of the correctly received frames but suffer from temporal error propagation can also be predicted. In the first tier of prediction the MAD is 2.65 dB and the standard deviation of the error is 5 dB which is rather high. However in the second tier of prediction (as introduced in Figure 2.8) the MAD is reduced to 1.53 dB and standard deviation of the error is reduced to 1.1 dB. It is also shown that the predicted PSNR positively correlates with the true PSNR. Namely, the correlation factor between the predicted and true PSNRs of the first tier is 83% and in the second tier it is increased to 92%.

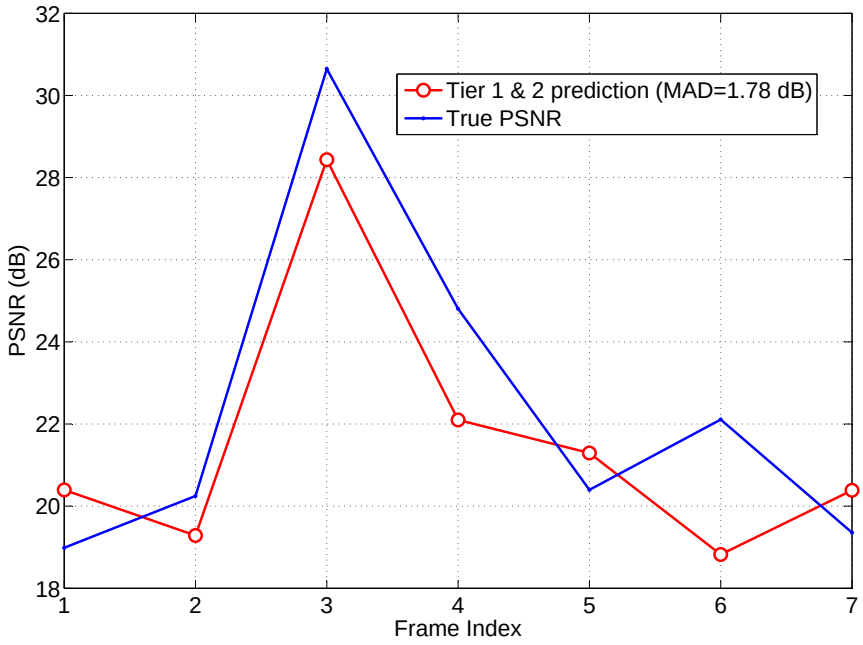


Figure 3.19: Comparison between the true and the predicted of PSNRs for Case 1

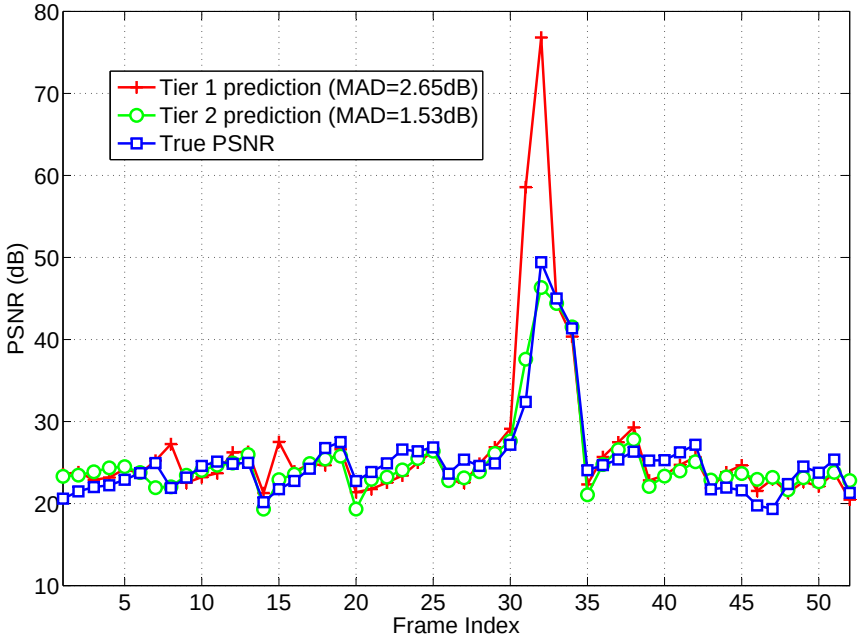


Figure 3.20: Comparison between the true and the predicted of PSNRs for Case 2

Chapter 4

Conclusions

Video streaming over wireless channels was studied in this thesis. Multi-level adaptive methods were proposed to overcome the inherent difficulties in wireless channels.

First, adaptive playback and bitstream switching were combined to ensure continuous playback. A video streaming system was simulated and its performance was evaluated with and without the proposed adaptive technique. The simulation results showed that the overall performance was improved with the adaptive scheme. Interruption of video playback was avoided at the cost of some reduction in the quality of the decoded video.

Second, scalable video coding was integrated with adaptive modulation and channel coding. A per-frame rate control technique was implemented based on the channel condition and the decoder buffer occupancy. Unlike other source rate control techniques which requires adjustment of video encoding parameters, the proposed scheme is less complex and allows real time adjustment of video frame sizes. Video streaming performance was studied for the three main ARQ schemes, Stop-and-Wait, Go-back-N, and Selective Repeat. The analysis and simulation results confirm the advantage of GBN and SR schemes over SW ARQ in transmission efficiency. It was also shown that the performance of GBN closely matches the performance of SR when adaptive FEC is used. This makes GBN with adaptive FEC a practical and less expensive choice in terms of complexity and buffering requirements when compared to SR. In addition, bandwidth utilization can be significantly enhanced with adaptive modulation and adaptive channel coding. It was shown that adaptive modulation and channel coding reduce not only the probability of buffer starvation, but also the

amount of required frame size reduction, hence, achieving better temporal and spatial video quality when compared to the fixed modulation systems.

Finally, new temporal video quality measures were introduced. The skip length and inter-starvation distance were combined with additional bitstream information to estimate the PSNR quality of video frames at the client side without the need of the reference video. High prediction accuracy was achieved as shown in the experimental results. The prediction system allows the sender or service provider to monitor the video quality on the client side and act accordingly to maintain a required QoS level.

In future work, fast varying channels can be considered where the channel can change within a video frame delivery time. Moreover, additional techniques such as hierarchical modulation can be considered and integrated with the proposed schemes to further improve the adaptability in video streaming over wireless channels.

References

[1] J. She, F. Hou, P. Ho, and L. Xie, "IPTV over WiMAX: Key Success Factors, Challenges, and Solutions [Advances in Mobile Multimedia]," *IEEE Communications Magazine*, vol. 45, no. 8, pp. 87–93, 2007.

[2] P. Chou and M. van der Schaar, *Multimedia over IP and Wireless Networks: Compression, Networking, and Systems*. Academic Press, 2007.

[3] 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, pp. 756–773, 1999.

[4] M. Hassan, L. Atzori, and M. Krunz, "Video Transport over Wireless Channels: a cycle-based approach for rate control," in *Proceedings of the 12th annual ACM international conference on Multimedia*, pp. 916–923, ACM, 2004.

[5] M. Hassan and M. Krunz, "A Playback-Adaptive Approach for Video Streaming Over Wireless Networks," in *IEEE Global Telecommunications Conference, 2005. GLOBECOM'05*, vol. 6.

[6] H. Chuang, C. Huang, and T. Chiang, "Content-Aware Adaptive Media Playout Controls for Wireless Video Streaming," *IEEE Transactions on Multimedia*, vol. 9, no. 6, pp. 1273–1283, 2007.

[7] Y. Li, A. Markopoulou, N. Bambos, and J. Apostolopoulos, "Joint Power-Playout Control for Media Streaming over Wireless Links," *IEEE Transactions on Multimedia*, vol. 8, no. 4, pp. 830–843, 2006.

[8] F. Zhai, Y. Eisenberg, T. Pappas, R. Berry, and A. Katsaggelos, "Joint Source-Channel Coding and Power Allocation for Energy Efficient Wireless Video Communications," in *Proceedings of the Annual Allerton Conference on Comunication Control and Computing*, vol. 41, pp. 1590–1599, Citeseer, 2003.

[9] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H. 264/AVC Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, 2007.

[10] S. Saha, "Image Compressionfrom DCT to Wavelets: A Review," *Crossroads*, vol. 6, no. 3, p. 21, 2000.

[11] M. Hassan and M. Krunz, "Video Streaming Over Wireless Packet Networks: An Occupancy-Based Rate Adaptation Perspective," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 8, pp. 1017–1027, 2007.

[12] X. Sun, F. Wu, S. Li, W. Gao, and Y. Zhang, "Seamless Switching of Scalable Video Bitstreams for Efficient Streaming," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 291–303, 2004.

[13] M. Karczewicz, R. Kurceren, N. Center, N. Inc, and T. Irving, "The SP-and SI-Frames Design for H. 264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 637–644, 2003.

[14] A. Argyriou, "Error-Resilient Video Encoding and Transmission in Multirate Wireless LANs," *IEEE Transactions on Multimedia*, vol. 10, no. 5, pp. 691–700, 2008.

[15] Y. Zhang, C. Zhu, and K. Yap, "A Joint Source-Channel Video Coding Scheme Based on Distributed Source Coding," *IEEE Transactions on Multimedia*, vol. 10, no. 8, pp. 1648–1656, 2008.

[16] D. Wu, Y. Hou, W. Zhu, Y. Zhang, and J. Peha, "Streaming Video over the Internet: Approaches and Directions," *IEEE transactions on circuits and systems for video technology*, vol. 11, no. 3, pp. 282–300, 2001.

[17] J. Proakis and M. Salehi, *Digital Communications*. McGraw-Hill New York, 2008.

[18] B. Barmada, M. Ghandi, E. Jones, and M. Ghanbari, "Prioritized Transmission of Data Partitioned H. 264 Video with Hierarchical QAM," *IEEE Signal Processing Letters*, vol. 12, no. 8, pp. 577–580, 2005.

[19] P. Vitthaladevuni and M. Alouini, "A Recursive Algorithm for the Exact BER Computation of Generalized Hierarchical QAM Constellations," *IEEE Transactions on Information Theory*, vol. 49, no. 1, pp. 297–307, 2003.

[20] A. Schertz and C. Weck, "Hierarchical Modulation - The transmission of Two Independent DVB-T Multiplexes on a Single Frequency," *EBU Technical review*, pp. 1–13, 2003.

[21] D. Kwon, W. Kim, K. Suh, H. Lim, and H. Kim, "A Higher Data-Rate T-DMB System Based on a Hierarchical A-DPSK Modulation," *IEEE Transactions on Broadcasting*, vol. 55, no. 1, pp. 42–50, 2009.

[22] C. J. B. Lambrecht and O. Verscheure, "Perceptual Quality Measure using a Spatio-Temporal Model of the Human Visual System," in *Proceedings of SPIE*, vol. 2668, pp. 450–461, March 1996.

[23] M. Pinson and S. Wolf, "A New Standardized Method for Objectively Measuring Video Quality," *IEEE Transactions on Broadcasting*, vol. 50, pp. 312–322, September 2004.

[24] A. C. B. Z. Wang, L. Lu, "Video Quality Assessment Using Structural Distortion Measurement," *Signal Processing: Image Communication, special issue on Objective video quality metrics*, vol. 19, pp. 121–132, February 2004.

[25] N. Damera-Venkata, T. Kite, W. Geisler, B. Evans, and A. Bovik, "Image Quality Assessment Based on a Degradation Model," *IEEE Transactions on Image Processing*, vol. 9, no. 4, pp. 636–650, 2000.

[26] H.264/AVC JM Reference Software, Available: http://iphome.hhi.de [Accessed: May 3, 2010].

[27] Z. Wu, J. Boyce, C. Res, I. Thomson, and N. Princeton, "An Error Concealment Scheme for Entire Frame Losses Based on H.264/AVC," in *2006 IEEE International Symposium on Circuits and Systems, 2006. ISCAS 2006. Proceedings*, p. 4, 2006.

[28] M. Kalman, E. Steinbach, and B. Girod, "Adaptive Media Playout for Low-Delay Video Streaming Over Error-Prone Channels," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 6, pp. 841–851, 2004.

[29] K. Cho and D. Yoon, "On the General BER Expression of One- and Two-dimensional Amplitude Modulations," *IEEE Transactions on Communications*, vol. 50, no. 7, pp. 1074–1080, 2002.

[30] M. Simon and M. Alouini, *Digital Communication over Fading Channels*. IEEE, 2005.

[31] S. Lin and D. Costello, *Error Control Coding*. Prentice-Hall Englewood Cliffs, NJ, 1983.

[32] A. León-García and I. Widjaja, *Communication Networks: fundamental concepts and key architectures*. McGraw-Hill Science/Engineering/Math, 2004.

[33] Video Trace Library, Available: http://trace.eas.asu.edu/ [Accessed: May 3, 2010].

[34] K.-A. Toh, Q. L. Tran, and D. Srinivasan, "Benchmarking a Reduced Multivariate Polynomial Pattern Classifier," *IEEE Transaction on on pattern analysis and machine intelligence*, vol. 26, June 2004.

[35] W. Campbell, K. Assaleh, and C. Broun, "Speaker Recognition with Polynomial Classifiers," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 4, pp. 205–212, 2002.

[36] T. Shanableh and K. Assaleh, "Feature Modeling using Polynomial Classifiers and Stepwise Regression," *Neurocomputing*, 2010.

[37] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H. 264/AVC Video Coding Standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 560–576, 2003.

[38] H.264/SVC JSVM Reference Software, Available: http://ip.hhi.de [Accessed: May 3, 2010].

Vita

Husameldin Mukhtar was born on August 27, 1982, in Omdurman, Sudan. He received his primary, preparatory, and secondary education in Abu Dhabi, UAE. He received his B.S. degree in Electrical and Electronic Engineering from the American University of Sharjah, UAE, in 2004 and graduated with cum laude.

From 2005 to 2008, he worked as a technical support and project engineer in the field of security systems in Citytec, Dubai, UAE.

In 2008, he joined the Electrical Engineering master's program in the American University of Sharjah as a graduate teaching assistant. He was awarded the Master of Science degree in Electrical Engineering in 2010. During his master's study, he co-authored more than 3 papers which were presented in international conferences. His research interest are in multimedia streaming, wireless communications, and digital image and video processing.