UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

VIDEO CODEC WITH ADAPTIVE FRAME RATE CONTROL FOR INTELLIGENT TRANSPORTATION SYSTEM APPLICATIONS

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

DOCTOR OF PHILOSOPHY

By

EKASIT VORAKITOLAN
Norman, Oklahoma
2014

Downloaded from SHAREOK. Persistent Handle URL: https://hdl.handle.net/11244/10375

VIDEO CODEC WITH ADAPTIVE FRAME RATE CONTROL FOR INTELLIGENT TRANSPORTATION SYSTEM APPLICATIONS

A DISSERTATION APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY

_____
Dr. Joseph P. Havlicek, Chair

_____
Dr. David J. Baldwin

_____
Dr. James J. Sluss Jr.

_____
Dr. Monte P. Tull

_____
Dr. Ronald D. Barnes

_____
Dr. Mohammed Atiquzzaman

© Copyright by EKASIT VORAKITOLAN 2014
All Rights Reserved.

Acknowledgements

I would like to thank you my committee members for more than kindness and generous with their valuable and precious time. A special tank you to my committee chairman, Dr. Joseph P. Havlicek, for countless hours of guidance, encouraging, proofreading, and opportunity throughout the entire doctorate program. I wish to thank you and appreciate Dr. David Baldwin, Dr. James J. Sluss, Dr. Monte Tull, Dr. Ronald Barnes, and Dr. Mohammed Atiquzzaman for the excellent advice, guidance and agree to serve on my committee.

I would like to acknowledge and say thank you to the Technology Services Division (TSD) at the Oklahoma Department of Transportation (ODOT) for their support. Thank to my colleague who worked with me at the OU ITS lab and the ODOT for a lot of encouragement and advice.

I would like to dedicate this dissertation research to my parents for their love and support. Thank you to my wife for encouragement and understanding. I wish to thank my kind son, Pundit, for giving me inspiration throughout the process.

Table of Contents

Acknowledgements iv

List of Tables viii

List of Figures ix

Abstract xi

Chapter 1: Introduction 1

Chapter 2: Background 6

2.1 The closed circuit television (CCTV) camera in ITS applications 7

2.1.1 Traffic Control, Management, and Congestion Reporting 8

2.1.2 Remote Weather Information System (RWIS) 9

2.1.3 Smart Work Zone 10

2.2 Basic Rate Controller for H.264 Video Encoding Standard 11

2.2.1 Adaptive Rate Control Based on Spatial Compensation 14

2.2.2 Adaptive Rate Control Based on Temporal Compensation 15

2.2.3 Adaptive Rate Control for Wireless Communication 17

2.2.4 Adaptive Rate Control based on the Structural Similarity Index (SSIM) . 18

2.2.5 Adaptive Rate Control based on the Video Quality Metric (VQM) 20

2.3 Data Communication Schemes on Internet Protocol (IP) Network 21

2.3.1 The Unicast and Broadcast Schemes 21

2.3.2 The Multicast Network Scheme 22

2.3.3 The Internet Group Management Protocol (IGMP) 23

2.3.4 IP Multicast Routing 24

2.4 Recent Technology for Open Source Video Compression 28

2.4.1 The Open-Source VP8 Video Codec versus H.264 29
2.4.2 The Open-Source VP9 Video Codec versus H.264 33
2.5 Summary 34
Chapter 3: Streaming Video Transmission 37
3.1 An alternative Method to Enhance Streaming Video Transmission 37
3.2 Network Simulation and Experiment 38
3.2.1 Experimental Procedure 39
3.2.2 Experimental Results and Discussion 46
3.3 Summary 47
Chapter 4: The Video Frame Rate Experiment 50
4.1 The Ping Test Network Tool Experiment 50
4.2 The Adaptive Video Frame Rate Control Experiment 53
4.2.1 Proposed Rate Control Strategy 53
4.2.2 Video Quality Testing Procedure 56
4.2.3 The Experimental Result Analyzer 61
4.3 Summary 65
Chapter 5: The Exploration of the Hardware Integration 66
5.1 The FPGA Hardware Development 66
5.1.1 Video Input Module 67
5.1.2 The Video Data Partition and the Macro-Block 70
5.1.3 The Video Hardware Compression Core 73
5.1.4 Completed FPGA Hardware Encoder 74
5.1.5 Results and Summary for the FPGA Video Encoder Implementation 78

5.2 Embedded Linux Hardware Development 79

5.2.1 Analog Video Input and the Hardware Encoder 79

5.5.2 Results and Summary for the Linux-Based Video Encoder..... 81

5.3 Summary..... 81

Chapter 6: Conclusion and Recommendations for Future Research..... 83

6.1 Conclusion..... 83

6.2 Contribution..... 84

6.3 Recommendations for Future Research..... 86

References 87

Appendix A: The Ping Test Result..... 99

Appendix B: The Video Quality Measurement and Analysis Results 105

Appendix C: The Sample Raw Data in the PPM Format..... 122

List of Tables

Table 1. Network data utilization in packets per second for several unicast/multicast combinations..... 48

Table 2. Video quality measured between original and received video streams for various network errors. 65

Table 3. Declaration for the high resolution EDID at 1280x720 pixels..... 68

Table 4. Declaration for the lower resolution EDID at 640x480 pixels..... 69

List of Figures

Figure 1. Illustration of the relationship between QP, MAD, and RDO in H.264 rate control, showing the circular dependency that is often referred to as the "chicken and egg dilemma"..... 13

Figure 2. Network simulation setup for testing video transmission. 39

Figure 3. Show stream menu choice..... 42

Figure 4. The device driver and video input configuration. 42

Figure 5. The transcoding activation. 43

Figure 6. RTP configuration for the multicast scheme..... 43

Figure 7. Multicast IP address and port configuration. 44

Figure 8. RTSP configuration for the unicast and multi-unicast schemes. 44

Figure 9. Both RTP and RTSP can be configured simultaneously. 45

Figure 10. Enabling all elementary streams 45

Figure 11. The example of the data utilization from port 23..... 48

Figure 12. WireShark results with multicast. 49

Figure 13. WireShark results with multicast and unicast. 49

Figure 14. Network diagram for the ping test. 51

Figure 15. Illustration of pixel errors in actual video from the Oklahoma statewide ITS. The errors are usually rejected to moving object such as vehicles..... 54

Figure 16. Rate control scheme for dropping and repeating frame. Top: original video stream at 30 fps. Bottom: the rate controller drops two out of every three frames. 54

Figure 17. Benchmark diagram for the perceptual video quality experiment. 58

Figure 18. H.264 encoder configuration for perceptual video quality testing..... 58

Figure 19. Packet limit configurations for the WANem network simulation. 59

Figure 20. Configuration of the MSU Video Quality Measurement Tool and setup of the CSV output files. 63

Figure 21. The descriptive statistics function to calculate the confidence interval 63

Figure 22. Decoded video frames received through a dropped packet network. 64

Figure 23. Illustrate a RGB single frame capture using For loop. 68

Figure 24. Sample of a single 720p video frame retrieved from memory..... 68

Figure 25. The 4:2:0 YC_rC_b sampling format illustrated for a 4x4 image. 71

Figure 26. The flow chart for the MB data partition. 72

Figure 27. Block diagram of the H.264 video compression core (from [76]). 73

Figure 28. ISE software configuration for the Atlys™ development board. 75

Figure 29. The MPMC interface for DDR2 memory. 76

Figure 30. The schematic diagram of the completed microprocessor subsystem. 76

Figure 31. The completed design for the MicroBlaze subsystem. 77

Figure 32. Block diagram of the video encoder implemented using an FPGA. 78

Figure 33. Show a list of the commands to configure VLC. 80

Abstract

Video cameras are one of the important types of devices in Intelligent Transportation Systems (ITS). The camera images are practical, widely deployable and beneficial for traffic management and congestion control. The advent of image processing has established several applications based on ITS camera images, including vehicle detection, weather monitoring, smart work zones, etc. Unlike digital video entertainment applications, the camera images in ITS applications require high video image quality but usually not a high video frame rate. Traditional block-based video compression standards, which were developed primarily with the video entertainment industry in mind, are dependent on adaptive rate control algorithms to control the video quality and the video frame rate. Modern rate control algorithms range from simple frame skipping to complicated adaptive algorithms based on optimal rate-distortion theory.

In this dissertation, I presented an innovative video frame rate control scheme based on adaptive frame dropping. Video transmission schemes were also discussed and a new strategy to reduce the video traffic on the network was presented. Experimental results in a variety of network scenarios shown that the proposed technique could improve video quality in both the temporal and spatial dimensions, as quantified by standard video metrics (up to 6 percent of PSNR, 5 percent of SSIM, and 10 percent VQM compared to the original video). Another benefit of the proposed technique is that video traffic and network congestion are generally reduced. Both FPGA and embedded Linux implementations are considered for video encoder development.

Chapter 1: Introduction

In December 1991, congress enacted the Intermodal Surface Transportation Efficiency ACT of 1991 (ISETA). This established a new era of transportation in America and one of the important things that emerged was the federal Intelligent Transportation System (ITS) program. The ITS program has emphasized research, development, operation, and implementation of new technology to improve transportation efficiency and safety. During fiscal years 1992 to 1997, the ISETA was granted 659 million dollars with additional funds making a total of 1.2 billion dollars for ITS [1]. The ITS program has been giving benefit to the whole nation to reduce crashes, fatalities, time, cost, emissions and fuel consumption, while improving throughput and customer satisfaction [2]. Public Law 105-178, the Transportation Equity ACT for the 21st Century (TEA-21), was established in June 1998 to continue improving the ITS program for a six-year period. TEA-21 authorized approximately 1.3 billion dollars [3]. In August 2005, congress signed the law for the Safe, Accountable, Flexible, Efficient Transportation, Equity ACT: A Legacy for Users (SAFETEA-LU) program. This was the end of the official federal ITS development program. However, the ITS research program funded 110 million dollars annually through 2009. On an ongoing basis, ITS programs are authorized to get benefits from regular federal-aid highway funding every year [4]. ITS programs deliver substantial public benefit, including time and fuel saving, reduced surface transportation costs, reduced emissions and pollution, and reduction of all types of accidents including fatality accidents.

ITS devices in Oklahoma include vehicle sensors, dynamic message signs (DMS), roadway weather information systems (RWIS), and video surveillance cameras. More than fifty percent of data traffic on the ITS network are video surveillance cameras. They require more network bandwidth comparing to other devices to deliver high quality video to the consoles. Over 100 stakeholder sites are monitoring and communicating with these devices through their consoles. Most of the stakeholders are involved in public safety and emergency responders such as state agencies, 911 dispatchers, EMS, department of public safety (DPS), and military agencies.

In general, typical media for the ITS can be dial-up modems, wireless radio, fiber optic networks, WIFI, air cards and microwave links [5]. However, reliable backbone networks such as the fiber optic network is not available throughout the entire network due to the cost constraints. Many stakeholder sites connect to the backbone network via lower bandwidth technologies such as private microwave links, 900 MHz radio, T-1 lines (1.544 Mbps) and CDMA modems. During heavy data traffic, these stakeholder sites can receive poor video quality due to the routers or switches dropping packets to prevent over packet limit which is the maximum number of data packets without having a buffer overflow.

The video surveillance cameras is one of the ITS devices that is defined in the National Transportation Communications for ITS Protocol (NTCIP) standard. The first document related to cameras for the ITS program appeared in 1997 and was called TS 3.CCTV. In July 1999, the NTCIP standard began to provide definition of the ITS devices and the control protocols, where the NTCIP 1205 standard applied to CCTV cameras [6]. However, the NTCIP does not define the video compression specification

for ITS applications. Video compressions for ITS cameras currently depend on both analog and digital video technologies, similar to video technology for entertainment industry. Since video compression for ITS applications has different requirement of video quality and movement at a decoder from entertainment industry, current ITS video cameras cannot provide the best video quality at the consoles as expected. For example, the existing ITS video distribution system in Oklahoma usually suffers from the incompatibility of network interfaces and dropping packets during heavy traffic because the regular video compression standards for video entertainment industry does not design for it.

In general video compression standards, output video image at the decoder suffer from a glitch or whole frame loss due to data packet loss. The packet loss usually occurs at the intermediate routers or switches dropping the data packet when data traffic congestion happens to prevent memory overflow. Many researchers have proposed algorithms trying to conceal the packet loss in video compression standards [10-13]. For example, researchers proposed the prediction of individual and multiple packet loss [10]. The linear model can be used to predict the visibility of multiple packet loss in H.264. Packet loss compensation can be done at the decoder. The experimental results show that the mean square error (MSE) between actual and predicted probabilities improves both individual and multiple packet loss [10]. Another effort on an algorithm for video packet loss compensation is a fully self-contained packet at a packet level which uses only information on itself to predict important information for the lost information. This method does not require frame level reconstruction or reference framing [11]. These linear models that are able to predict packet loss can be adapted to

another video compression standard such as MPGE-2. This method is done within a packet and its vicinity. Video error concealment is also done in the decoder [12]. However, new approaches for video packet loss concealment involve a complicated calculation and hardware modification. All proposed algorithms are only designed for packet loss compensation. In practice, not only packet loss can occur on the video transmission process, but also other types of data loss such as packet reorder, packet duplication, packet delay, data jitter, etc.

I presented a new video compression system that provided extremely high quality video for ITS applications at the user consoles. The original contributions of this dissertation include the following:

- I proposed a new video data distribution to reduce the data traffic on the network by implementing video distribution control algorithms based on the unicast and multicast theory. I conducted experiments to prove that the video distribution control algorithm can effectively reduce the video data traffic. When the data traffic on the network is reduced, the dropping of packets from the network equipment due to over packet limit should be reduced as well.
- I developed a new adaptive H.264 rate control scheme based on adaptive frame dropping. The proposed rate control scheme is controlled by the simple ping test module to adjust the video frame rate. The conducted experiments shown that the proposed rate controller could effectively improve the video quality at the user consoles and was suitable for ITS applications.

- I developed FPGA and embedded Linux implementations of the new adaptive H.264 rate control scheme and conducted experiments to characterized and compare their performance.

In this dissertation, the backgrounds of video camera for ITS applications are presented in Chapter 2. Several case studies of video camera application with the final results are reviewed. The problems associated with trying to deploy a generic video encoder in ITS network are introduced. Bitrate and frame rate control for block-based video compression are discussed. The conventional frame layer rate control theories for H.264 are reviewed. The communication type for the Internet Protocol (IP) network is proposed. The video delivery strategy with experiments and outcomes is described in Chapter 3. With the proper data delivery approach, the video data traffic can be reduced. In Chapter 4, the experimental results for the ping test module are described and it is shown that the simple ping test module can be used to adjust the proposed rate control. The innovative video frame rate control with experiment is also presented in this chapter. The design, development, and evaluation of modern and well-known hardware platforms for ITS video encoder based on the proposed video rate control are shown in Chapter 5. Finally, conclusions and recommendations for future research are given in Chapter 6.

Chapter 2: Background

The Intelligent Transportation System (ITS) program in the United States has been well established for more than a decade [1-4]. To achieve interoperability and interchangeability among different manufactured devices, the National Transportation Communications for ITS Protocol (NTCIP) standard was developed to specify the standardized protocol to support communication between computers and ITS control devices. This standard is a joint project of the American Association of State Highway and Transportation Officials (AASHTO), the Institute of Transportation Engineers (ITE), and the National Electrical Manufactures Association (NEMA). In general, ITS devices include several different electronic components such as the Dynamic Message Sign (DMS), the closed circuit television (CCTV) camera, and microwave and inductive loop vehicle detectors. Although the NTCIP specifics standardized ITS device control (for example, NTCIP 1203 for DMS, NTCIP 1205 for CCTV camera, and NTCIP 1202 for vehicle detectors), it does not define the video compression and transportation scheme between the video server and client [6-9]. According to NTCIP 1213, the communication media is not specific for the ITS network. Modern ITS networks are usually a combination of subsystems integrating hybrid network technologies such as telephone dial-up, wireless, microwave, cellular modem, and fiber optic [5]. The digital video encoding scheme is typically based on the general block-based video compression standards such as MPEG-1, MPEG-2, MPEG-4, H.264, etc. Currently, video distribution in the Oklahoma statewide ITS uses one of two fixed configurations: either unicast, where transmission is to a single user with a specific

destination IP address, or mulicast, where transmission is to multiple user using the IP mulicast address. Basically, the existing video compression in the Oklahoma statewide ITS cannot provide an acceptable video quality at user consoles. In this chapter, I will review use of the camera in various ITS applications, rate control for block-based video compression, and video transmission schemes through data networks. VP8 and VP9 are recent open source video compressions. The comparison and contrast of technology between open source video compressions and H.264 are also reviewed in this chapter.

2.1 The closed circuit television (CCTV) camera in ITS applications

The CCTV camera is widely deployed in ITS applications. It is a useful device for providing remote surveillance of the traffic network. With image processing and pattern recognition technology, the camera is able to replace many analog sensor devices (for example, a loop detector, a microwave sensor, and an ultrasonic wave sensor). The new camera sensor devices are cheaper to deploy, easier to maintain, and more robust because the installation can happen immediately without traffic interruption and physical damage to the surface of the road. When the CCTV camera is working with other devices, the new systems can be easily applied to benefit the public in ways such as traffic control, traffic monitoring and congestion reporting, weather information, and smart work zones. However, the accuracy of the new application system is dependent on the quality of the camera images. In this section, I will review some new ITS applications using the CCTV camera as a measurement tool.

2.1.1 Traffic Control, Management, and Congestion Reporting

Traditionally, current traffic conditions and incidents including crashes can be immediately viewed at the Traffic Management Center (TMC) without sending any workers into the field. The operators at the TMC use real time information to control traffic signals along the street by changing the signal timing at each intersection depending on the traffic flow. With the introduction of Digital Signal Processing (DSP) and pattern recognition theory, several research papers about traffic sensors based on CCTV images were written [14-17]. The real time vehicle detector can be implemented by using the embedded DSP hardware module based on the target detection algorithm. One of the most well-known algorithms for the vehicle detector is the vehicle corner detection where the point of interest is at the corner of the vehicle [14]. Image based vehicle detection also requires only a few hardware components, thereby providing sane of the lowest system deployment costs, implementation costs, and maintenance costs. The regular microcontroller and high speed Field-Programmable Gate Array (FPGA) can be designed to work with the CCTV camera for basic low-level image processing at the camera site. Complex calculations and higher level signal processing can be done on a computer at the TMC [15].

In general, camera-based vehicle detection can be used for measuring the traffic flow, where the principle parameters are traffic volume, velocity, and density. To detect vehicle presence or absence, the vehicle detection can examine from the captured images (snap shot) at the interested regions of the images (spatial dimension). The frame rate (temporal dimension) is less important. This type of vehicle detection is possible to operate with the fluctuated video frame rate (not requiring to be fixed at 30

frames per second) [16]. To maximize the accuracy of the traffic flow measurement, the quality of the video image (spatial dimension) is of more concern than the video frame rate (temporal dimension) because only a single video frame is used to determine the vehicle presence or absence [16].

2.1.2 Remote Weather Information System (RWIS)

Inclement weather is a traffic hazard because it can reduce the driver's vision and makes severe road conditions which increase the risk of traffic accidents. In most states, the road maintenance division of the state DOT is responsible for thawing ice and snow plowing. The Remote Weather Information System (RWIS) is one of the ITS devices that provides real-time weather conditions to the TMC or road maintenance officers. In general, the major meteorological parameters reported by the RWIS are wind speed, wind direction, surface temperature, subsurface temperature, relative humidity, etc. Some parameters such as dew point can be derived from a calculation involving the relative humidity and air temperature. In some circumstances, the meteorological parameters alone are not enough information to meet the requirements from road maintenance officers. For example, snow fall can occur over a wide range of temperature and relative humidity. The precise road conditions during inclement weather can be determined using a weather analysis model and meteorological parameters. The road condition prediction can be performed with better than 91 percent accuracy when the analysis of road conditions is combined among the results from the analysis model, meteorological parameters, and video camera images from the field [18]. In general, high quality video images can provide enough information to

determine the basic weather conditions with human vision such as snow, rain, cloudy and sunny. At the similar concept of human vision, the texture analysis theory can be applied to the captured video image to determine the weather conditions. For example, the brightness from a sunny day makes the video images clearer and the vehicle's shadow becomes visible [19]. For stationary areas like medians and paved shoulders, snow fall can be easily detected using digital image processing methods [20]. The CCTV cameras for ITS are working perfectly for the RWIS applications. The RWIS stations provide a big step towards identifying the weather conditions along the roadway. However, the quality of the input video images has a big impact on the system accuracy, effectiveness, and performance.

2.1.3 Smart Work Zone

Work zones are one of the most hazardous areas that frequently cause fatal crashes. Both motorists and road workers have been facing the dangerous conditions along work zones every year [21]. Many researchers have been continuously working on strategies to make work zones safer [22-25]. Traditional measures to improve work zone safety include warning signs, concrete barriers, flashing arrow panels, rumble strips, etc. ITS devices can be put together to improve traffic flow and reduce traffic accidents along work zones. A smart work zone is normally designed using vehicle detectors and portable DMS that work together. The real-time traffic conditions are sent back to the motorists via the portable DMS. However, the traditional vehicle sensor, similar to a loop sensor, is very difficult to deploy in a work zone. With the recent technology, machine vision integrated with a video image processing module could be

implemented in a work zone. Such a system was installed and tested at two work zones on I-35 and I-94 in Minnesota [22]. Traffic information obtained from the machine vision processing includes traffic speed, and volume, incident detection, and vehicle intrusion into the work zone. The overall project successfully reached expected goals except for some accuracy issues resulting from problems with the video image feeding the machine vision processing. The main data communications for these projects was wireless. Low bandwidth and unstable connection led to noisy video with undesirable fluctuations [22]. Other simulation models have also indicated that work zone safety can be achieved using video image processing [23, 24].

With the image processing method, a smart work zone that has the camera as a traffic sensor provides a convenient scheme and a cheaper solution to reduce the traffic congestions and accidents. However, the system accuracy is based on the quality of the video inputs feeding the image processing module.

2.2 Basic Rate Controller for H.264 Video Encoding Standard

The purpose of rate control is primarily to regulate the video bit stream and make sure that the best video quality is achieved without violating the encoder buffer size and available bandwidth. In block based video compression, the Quantization Parameter (QP) controls the level of DCT coefficient quantization, providing a tradeoff between image quality and bitrate. There is an inverse relationship between QP values and data size. At lower QP values, the bitrate is higher and the video quality is higher. On the other hand, when the QP value is higher more spatial image detail is discarded through lossy compression and both the video quality and the bitrate are reduced.

In practice, the QP value needs to be changed in response to changes in the video and network environments such as time variations in the complexity of the video, the latency of the network, and the system noise. Manually applying open loop control of the QP value alone is not enough to produce good video output quality. To achieve high video quality at low bitrate, it is generally necessary to introduce a video rate controller in order to implement closed loop control of the video quality and bitrate.

In the H.264 standard, which provides for sophisticated rate control, many prediction modes for data compression are defined including seven modes for inter-frame prediction (temporal) and nine modes of intra-frame prediction (spatial) [33]. The question is how to select an appropriate prediction mode in any given situation due to the varying of network environment and video information content. H.264 specifies a Rate Distortion Optimization (RDO) algorithm for selecting the prediction mode by pre-calculation. The RDO evaluates the result from each perspective prediction mode and chooses the mode based on optimum metric results [33].

In block-based video compression, a single video frame is divided into 8x8 blocks of non-overlapping pixels. Four blocks of 8x8 are grouped together to be a new big block called Macro block (MB). The difference between a prediction MB from the last frame and the corresponding MB of the current frame is called the residual. Considering two different video sequences, the number of moving objects and speeds on each video sequence are different. The difference between the video sequences can be defined by the source complexity. Therefore, each video sequence can have different encoding complexity. The Mean Absolute Difference (MAD) is closely related to

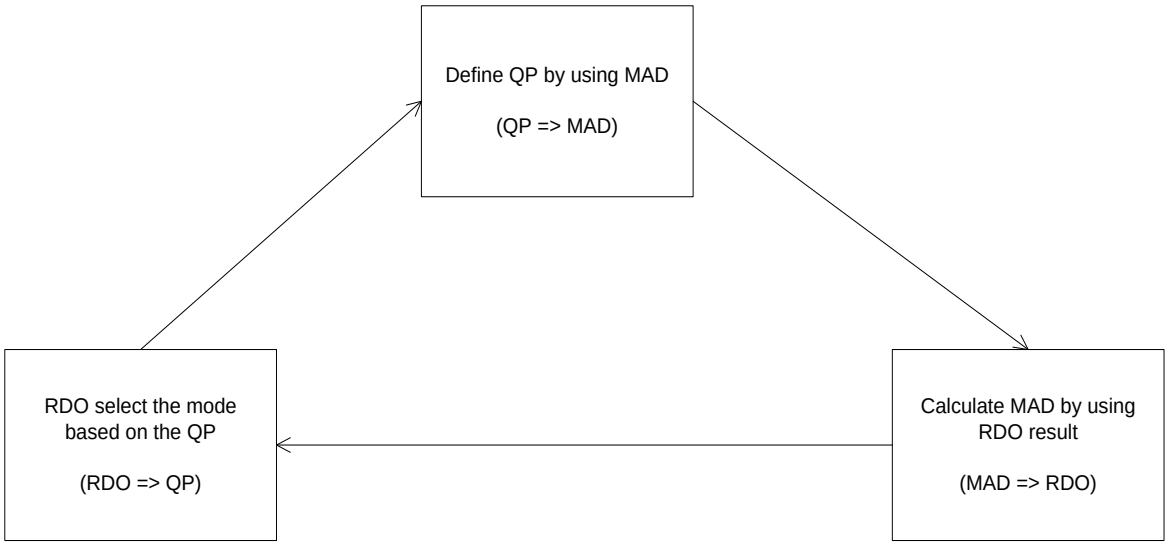


Figure 1. Illustration of the relationship between QP, MAD, and RDO in H.264 rate control, showing the circular dependency that is often referred to as the "chicken and egg dilemma".

encoding complexity of the leftover residual from the intra-frame or inter-frame prediction modes [33].

The closed loop control involves a relationship between QP, MAD, and RDO as shown in Figure 1. They rely on each other to produce an optimum encoding. This results in a "chicken and egg dilemma" [26]. Definition of QP requires the MAD of the current frame. To get the MAD of the MB from current frame, the RDO must be run first to select the prediction mode. But running the RDO requires the QP value. So there is a circular dependency between QP, MAD, and RDO.

There are several proposed solutions to resolve this dilemma with the closed loop control. One of them is the scalable rate control (SRC), which is accepted as a part of the H.264 standard. The SRC works at the frame, video object, and macroblock (MB) levels, improving the rate-distortion model and providing a very low bit rate frame work. Some additional new methods to improve the accuracy of the rate-distortion

model are given in [27]. The sliding-window method smooths temporal variations during scene changes. Frame skipping control prevents video buffer overflow. The scalable MAD improves the accuracy at the RDO. Basic rate control algorithms can be applied to improve spatial resolution, temporal resolution, wireless communication, structural similarity index, and video quality metric. Most of these will be reviewed in the following sections.

2.2.1 Adaptive Rate Control Based on Spatial Compensation

Many people have proposed solutions that break the circular dependency by initially predicting the MAD value rather than calculating the actual MAD value [26]. Prediction of MAD can be done by a linear method. Comparison between fixed quantization and adaptive quantization using a linear model has shown that the adaptive method improves the average Peak Signal to the Noise Ratio (PSNR) by up to 0.75 dB over the fixed model [28]. Another possibility is to predict QP at the first frame of a Group Of Pictures (GOP), a group of successive pictures within a encoding video stream, by using the Initial Quantization Parameter Prediction (IQPP) algorithm and the Remaining Quantization Parameter Prediction (RQPP) algorithm [29]. IQPP gives an average PSNR of 0.59 dB more than the original method while the RQPP can reduce buffer overflow and maintains video quality under low high network latency [29]. In general video coding, not all of the video frames are coded independently. Only the first frame of a GOP is coded independently, known as an I-frame [30]. The frame that refers to the previous decoded frame and contains the motion compensation of different information between the current frame and previous frame is called P-frame [30]. The

advanced linear model is defined using the relationship between the average MAD prediction value of the previous P-frame in the GOP and the actual MAD value. Experimental results showed that the MAD prediction error is reduced by 34 percent compared to the traditional linear model [30].

Some people claim that MAD prediction using the traditional linear model is not accurate [31]. It has been proposed to adjust the QP value based on the encoded frame information and to change the video frame rate based on the frame complexity coefficient. This method can reduce the bit error rate by half and improves average PSNR by 0.69 dB compared to the original algorithm [31]. The linear model can provide the best results at higher bitrates. However, it cannot accommodate the dramatic data variations that occur during scene changes. The methods to support scene changes will be described in the next section.

2.2.2 Adaptive Rate Control Based on Temporal Compensation

In this section, the rate control strategies to improve video quality during the scene changes (temporal) are introduced. The basic technique for typical video compression standards to encode the video content is trying to reduce the redundant data. For inter frame coding, the video compression algorithm is usually to select and transmit only the part of changing data of current frame comparing to the previous frame. Therefore, the video sequence that has slow moving objects can reduce more redundant video data than the video sequence with fast moving objects. This scenario also applies to the number of scene change on a video sequence because the moving object and speed are based on the scene changes. Motion vector is a two-dimensional

vector used for inter frame prediction. When video frame is not changing or slow moving object, most of the MB is prefect match and motion vector is approach to zero. Due to the relationship between adjacent frames on a video sequence, some people proposed the solution to improve the prediction frame based on the previously statistical encoded frames associated with MAD. The MAD can use to estimate the video frame complexity allowing the video encoding system to allocate the suitable video buffer [32]. When the video buffer has allocated enough memory to handle the video data, the buffer overflow should not occur. Some novel rate control algorithm that applies to both frame level and MB level is able to alleviate the video error during the scene change [33]. On the frames level, the traditional linear model alone cannot be used to solve the chicken-egg dilemma circular dependency during the scene change because the previous MAD value is larger than the current average MAD. To improve these problems, the new scene change indication parameter can be defined and used as reference for the MAD determination. On the MB level, the QP can be defined in two steps. The calculation of the coarse QP can be done right after the defined MAD value. The fine QP can be implemented later to get the accurate value. By integrating the motion estimation between the coarse QP and fine QP, the RDO should be able to pick the best mode and improve the scene change problems [33]. The traditional rate control methods adjust the QP value to maintain the video quality and buffer. In case of buffer overflow, arbitrary frame skipping can be used to maintain the performance [34]. However, arbitrary frame skipping may reduce the video quality. Some people proposed active frame skipping where the video skipping frames are based on the video quality in both spatial and temporal dimensions [34]. Experimental results showed that the active

frame skipping algorithm can free more video buffer and improve both average PSNR and video consistency. In many circumstances, the perceptual quality is actually more important than the average PSNR value [34].

Although rate control is not a new research topic, it is an attractive problem for scientists to find out a new solution to improve the rate control for H.264 and new video coding standards. Rate control is still an important issue for unstable networks and low bandwidth environments such as wireless network links that will be discussed in the next section.

2.2.3 Adaptive Rate Control for Wireless Communication

Fiber network is one of the reliable communication media that provide huge data bandwidth transmission up to Gigabit Ethernet (GigE.). However, the cost for deployment, operation and maintenance for the fiber network is high. Because the underground fiber optic installation involves trenching and digging, it is impossible to implement the fiber network for the entire areas. Wireless communication is cheaper and faster to deploy and maintain than the fiber optic and wire network. However, wireless has several issues to consider including but not limited to security, interference, latency, jitter, etc. The wireless network also delivers lower bandwidth than the fiber optic network. The examples of alternative media types (wire network) providing low bandwidth include dedicate circuit switch, Plain Old Telephone (POT), and telephony [35]. The wire networks provide reliability, consistent delay and stable jitter making it easy to transmit streaming video in the reliable network environment. However, a novel wireless network (including microwave, 3G, and 4G) suffers from the fluctuating delay,

jitter, and unstable bandwidth. Some scientists introduced the interactive video transmission system based on the client/server configuration [35]. While the video server encodes the video image and transmits it to the client, the client will calculate the jitter value of the media network between the server and client. The jitter estimation is sent back to the server as a reference to adjust the parameter for encoding the video stream. There are two ways to modify the size of the video stream: by either changing the video frame rate or changing the bitrate. The modification of QP based on the variable frame rate and variable GOP condition is another option. This is an example of real-time applications used for the 3G network at the Telefonica Investigation Y Desarrollo [35].

Because the characteristics of the mobile communication network are changing, it is very difficult to transport the video stream using a mobile network. Several methods have been presented by converting encoded video from high bitrate to low bitrate based on network environment fluctuations [36]. For example, the re-quantization of pre-encoded video by changing the QP reduces the residual size and bitrate. The spatial resolution can be reduced to lower the bitrate [36]. A new way has been provided to improve the video quality by selecting and skipping the frame based on visual complexity (spatial) and temporal coherence [36].

2.2.4 Adaptive Rate Control based on the Structural Similarity Index (SSIM)

The structural similarity index (SSIM) is a metric that compares the similarity of two images [37]. The SSIM metric has been developed to replace other metrics that have been used for a long time but have been demonstrated to be inconsistent with

human visual perceptual such as PSNR and Mean Square Error (MSE) [37]. In the previous sections, many adaptive rate control strategies based on the spatial and temporal compensation adjust the QP values and judge the video quality based on the average PSNR [28, 29, 31, 33]. However, QP adjustment algorithms that are judged by the PSNR do not provide perfect results. For example, some studies indicate that the traditional method of QP adjustment to meet the required bitrate only works perfectly in cases where the bitrate is high [38]. The result from encoding at low bitrate will reduce the video quality (spatial) to maintain the video frame rate (temporal). The video passive frame skipping is applied when buffer overflow occurs.

The SSIM metric can be easily used to create an effective rate control algorithm. One solution that has been proposed is an adaptive frame skipping scheme based on the SSIM metric rather than on changing the QP [38]. Most traditional rate control algorithms do not seriously take RDO into account [39]. The SSIM can be used as a metric for RDO modeling. The bit allocation optimization and rate control can be developed depending on the RDO modeling. The experimental results for the individual test module show that the video bitrate is reduced by 25 percent compared to the original [39]. The SSIM index and Q-step quadratic distortion model have been proposed to measure and estimate the distortion optimization. Consequently, the Lagrange multiplier method can be implemented to calculate the result. The new solution calls a SSIM optimal MB layer rate control scheme [40].

2.2.5 Adaptive Rate Control based on the Video Quality Metric (VQM)

In the previous sections, several video quality metrics have been reviewed. Although they can be used to judge the visual quality at some levels, the best way to evaluate the video quality is using human perception. Subjective visual quality assessment refers to the process of the rating the quality of images or video by human perceptions [41, 42]. In some circumstances, people may feel unpleasant with the visual video sequence that does not have smooth video frames although the video result shows high average PSNR. The video quality metric (VQM) is another measurement based on spatial-temporal distortions that is designed for measuring the consistency of video sequences [44]. VQM is an objective metric. To evaluate the performance of an objective metric, the correlation between subjective scores and the objective metric value is examined to assess the accuracy of the metric compared to human perception of the images or video [44]. It has been proposed that the result of VQM can be used as a reference value to measure visual quality consistently [43]. To optimize between the video buffer and visual quality consistency, the VQM-based window model can be applied based on the VQM reference value. Finally, the window-level rate control algorithm replaces the original rate controller depending on the VQM reference value and VQM-based window model. The experimental results judged by the human visual system (HVS) show that the visual quality consistency is improved compared to the original [43, 44]. In addition, the VQM value can be used as a reference to control the QP and frame rate, similar to a traditional rate controller where the linear model is the prediction for QP [45]. In addition, frame rate and QP control by VQM has been shown to be an excellent option for the mobile network environment [45].

2.3 Data Communication Schemes on Internet Protocol (IP) Network

Unicast, multicast and broadcast are the typical routing schemes used in an Internet Protocol (IP) network. Each of these schemes is designed to support different types of applications depending on the structure of the desired data distribution. Either unicast or multicast is normally the major routing scheme in the Oklahoma ITS digital video distribution. The system administrator makes a decision about what is routing scheme to deploy on each video link from the camera locations along the roadways to the TMC or other video end user nodes.

2.3.1 The Unicast and Broadcast Schemes

In the unicast routing scheme, data packets are sent from one source to a single destination or receiver (point-to-point). The original source defines an explicit destination receiver before forwarding the data packet to the router. The router determines the shortest data routes based on the unicast routing table and forwards the data packet to the destination through the shortest route. This is in contrast to the broadcast scheme, where the data packet is sent to all receivers throughout the network, simultaneously (point-to-multipoint). The broadcast scheme can be identified by the IP address of the host portion. On the broadcast scheme, the IP address values on the host portion are set to one. The router understands the broadcast packet and hence distributes the packet to all remote devices in the local sub-network.

The advent of new digital entertainment, business data casting, and digital media communication requires sending data packets from one source to a large selective group of receivers (point-to-multipoint). Actually, the unicast scheme can be used to send

multiple separate data transmissions with the same content to multiple destinations. This is called multi-unicast. The multi-unicast scheme is not appropriate to use in the Oklahoma ITS network because the multiple data packets of the same content are transmitted around the network increasing the bandwidth occupancy and increasing server loads to support the increased number of packets and destination devices.

2.3.2 The Multicast Network Scheme

In early 1980, a distributed operating system project was developed at Stanford University [46]. This project was made up of several computers connected together using a single Ethernet segment. All computers on the Ethernet segment were integrated and communicated with one another using special messages on the operating system level. A computer in the segment was permitted to send a message to other computers in the group by using the Media Access Control (MAC) layer multicast packet. Subsequently, it becomes necessary to add more computer systems to the project. However, the new machines were located in another part of the network group. To achieve communication between all of the computers in the project, it was necessary to extend the communication and MAC layer to support multicast operations on layer 3 of the Open System Interconnection (OSI) layered model [49].

The multicast scheme is designed to support data packet transmission from one source to a group of multiple selective remote devices. A special IP address group has been defined for the multicast network application called the *IP multicast group address* [49]. The Transmission Control Protocol (TCP) used for unicast is a guaranteed data delivery scheme, which provides reliability for IP unicast data. Unlike the unicast

scheme, the multicast scheme involves delivery of data to multiple destinations from one host. The multicast scheme normally uses the User Datagram Protocol (UDP), which is a "best effort" approach to data delivery [49]. Although data delivery with the multicast scheme is faster than the unicast scheme due less overhead in the packet header and a lack of data handshaking, a video coding algorithm for use in multicast environment is still facing data loss, latency, jitter, etc. In addition, any part of the network that desires to subscribe to the IP multicast is required to have the multicast protocol enabled. The basic concept of the multicast protocol does allow subscribers to join the multicast stream while also supporting a pruning mechanism on non-subscriber portions of the network.

2.3.3 The Internet Group Management Protocol (IGMP)

The Internet Group Management Protocol (IGMP) operates on layer 2 of the OSI layered model to communicate between a host and an intermediated multicast router [97]. IGMP is classified as a group membership protocol (a dynamic host registration). It enables all nodes in the network to specify whether or not they want to receive multicast traffic. IGMP allows a host to dynamically register (join) a particular multicast traffic stream for a specific group with the multicast router [49]. From time to time, the multicast router will send a query-response process to find out which hosts still want to remain a member of the group. The hosts then send a confirmation message back to the router if they wish to remain a member. However, any host can send the "leave group" message to the router at any time [49]. The multicast router will then prune the network by removing the specific host from the member list.

2.3.4 IP Multicast Routing

In a single local area network (LAN) topology, the multicast scheme can be implemented to operate at layer 2 of the OSI model [47]. However, when the network comprises multiple LAN subnets, multicast is required to be implemented in layer 3 of OSI model. The multicast network is required to have a mechanism to create a distribution tree. The distribution tree is responsible to send a data packet to each network branch where the multicast memberships are located. In the unicast scheme, a router forwards the data packet to the next hop that provides the shortest path link based on the unicast routing table. However, with multicast, a single unique IP address applies to multiple destination hosts. For example, in a simple multicast scheme, routing may be done by looking at the source IP address instead of the destination IP address and then forwarding data packets away from the source. This is called reverse path forwarding [49]. With reverse path forwarding, the router checks incoming traffic and decides whether to forward each packet or to discard it to prevent loops. Although reverse path forwarding is a simple way to implement multicast distribution, it is high in consumption of router resources including bandwidth and memory. This simple algorithm also cannot support large scale network topologies [49].

❖ *Multicast Forwarding Algorithms*

The source tree and shared tree structures are two other types of multicast distribution trees that can be used as alternatives to simple reverse path forwarding [47]. The source tree structure is a simple distribution tree that disseminates multicast traffic from the source to the destinations using the shortest path route. The source tree requires more memory on the routers to keep the route information for reference.

However, it provides the minimum latency route between a source and a set of receivers [47]. Performance of the source tree is generally excellent for a multicast network where the number of sources is small but the number of receivers is large [49]. A digital television broadcasting station is an example of such a network.

For a shared tree, one location in the network is chosen to be the core or center from which multicast traffic will be distributed. This core or center is called the Rendezvous Point (RP) [47]. For a shared tree, the source sends multicast traffic to the RP where it is forwarded throughout the network to the active receivers. The shared tree requires less router memory than a source tree [47]. However, it may create delays and may duplicate multicast packets from source to RP and RP to receivers. The shared tree topology is suitable for use with many senders in low bandwidth applications [47].

❖ *Multicast Routing Protocols*

The mechanism that manages the distribution tree structure is called a multicast routing protocol. With multicast routing protocols, the distribution tree can be examined via the routing and forwarding table. Actually, the routing and forwarding table contains the unicast reachable information [49]. There are several types of multicast routing protocols for the management of multicast distribution trees, such as Protocol Independent Multicast (PIM), Distance Vector Multicast Routing Protocol (DVMRP), Core-based Tree (CBT), Multicast Open Shortcut Path First (MOSPF), etc. [47]

The operation of PIM is based on the existing unicast routing table. The internet Engineering Task Force (IETF) defines PIM under the Request for Comment (RFC) documentation [47]. The router uses PIM to determine which other routers are multicast traffic receivers. PIM can be subdivided into the PIM-Sparse Mode (PIM-SM) and the

PIM-Dense Mode (PIM-DM) [47]. PIM-SP is defined under RFC 2117, while the IETF is still in the process of drafting the RFC for PIM-DM [48, 49]. The selection of either PIM-SM or PIM-DM is depended on the network size and topology. PIM-DM works better on a network that has only a few senders, many receivers, high volume of multicast traffic, constant traffic, and short distances between senders and receivers [47]. PIM-SM works better for a network with only a few receivers, intermittent multicast traffic, and scatter traffic between senders and receivers [47].

DVMRP is the first IP multicasting protocol that works similarly to the unicast *routing information protocol* (RIP) in the sense that it is based on the distance vector routing protocol [47]. DVMRP is defined under RFC 1075 [48, 49]. Often, DVMRP is used to create an Internet multicast backbone (Mbone), which is a virtual network for transmitting audio and video through the Internet [47]. DVMRP is an interior gateway protocol and can interface between different network topologies. However, one limitation of DVMRP is that the number of hops cannot exceed a maximum of 32 [47]. Also, DVMRP does not support the unicast datagram. Therefore, with DVMRP, two separate routing processes are required in a router that supports both multicast and unicast datagrams [47].

Like PIM, CBT is another multicast protocol that is based on the existing unicast routing table. The IETF defines CBT under RFC 2201 and RFC 2189 [48, 49]. The CBT has a center router to establish a network tree that is linked to the network path where active receivers are located. When a host intends to be a member of the group, it sends the join request to the center router either directly or through same set of

intermediate routers. Like other multicast routing protocols, any host can also send the leave message to terminate as a multicast group member.

The Open Shortest Path First (OSPF) link-state table can be used to create a distribution tree which is utilized by MOSPF [47]. MOSPF is defined under RFC 1584 and RFC 1585 [48, 49]. MOSPF establishes the routing path based on the least cost path from the existing OSPF database [49]. Although OSPF is only dependent on the destination address, MOSPF is based on both the source and destination datagrams [47].

❖ *Multicast Routing Protocols-The Operation Modes*

The multicast routing protocols may typically be classified as being either Dense-Mode (DM) or Sparse-Mode (SM) protocols. A DM protocol assumes that all of the receivers in the network are member of a multicast group. Then, traffic is delivered to all points in the network and any sub-networks through a process called flooding. After flooding occurs, the routers have to send explicit prune messages to turn the traffic back off at specific branches where they do not want to be a receiver [47]. With this flooding and pruning, the multicast traffic is sent only to subscribe receivers. However, any pruned node can request to be a member later by issuing a rejoin message. The DM protocol is dependent on the source tree algorithm [47]. DVMRP, MOSPF, and PIM-DM are examples of multicast routing protocols that are supported by DM.

The SM protocol works in a way that is opposite to the DM protocol in that SM only forwards traffic to receivers when they request it. The distribution network starts with the empty tree and adds receiver nodes as they explicitly request to join by issuing the join message [47]. The PIM-SM and CTB protocols are examples of multicast

routing protocols for SM. The SM protocol can be used with either a source tree or a shared tree distribution algorithm. Like a DM protocol, the active routers can send out the prune message when they do not want to receive multicast traffic [47-49].

2.4 Recent Technology for Open Source Video Compression

In February 2010, MPEG-LA, LLC which owns H. 264/ AVC (MPEG-4 part 10), announced that they will charge a royalty fee to use their video coding for Internet video by end of 2015 [50]. Google, Inc. which currently owns Youtube, a video sharing website, acquired the TrueMotion video codec series (VPS-VP8) from On2 Technology, Inc., and released it as an open source video compression standard under the Berkeley Software Distribution (BSD) license in May 2010. Google aimed to replace the Adobe Flash Player and H.264/AVC with the new VP8 standard (WebM) [51]. Major web browsers started to support VP8, including the free HTML5 web video standard [51]. Because VP8 was developed as an alternative to H.264/AVC, several major concepts of operation and the archetype of VP8 started to be shared with the H.264/AVC.

In 2011, MPEGLA reviewed significant technique patents to form the VP8 patent pool [53]. The United States Department of Justice (DoJ) investigated the MPEG license and found that 12 patent holders were involved [51]. While the patent lawsuit process was still on going, the IETF defined RFC 6386 for the VP8 data format and released the decoding guide at the end of 2011 [51, 52]. However, many revisions have been released to update RFC 6386. In March 2013, MPEG-LA announced an agreement to allow Google to use the essential license techniques for VP8 [53]. At the time of this

writing, Google plans to release VP9 and start transcoding video databases for the Youtube website [54]. However, Nokia claims recent patent infringement against Google and may prevent VP8 and VP9 from ultimately being royalty free [55].

2.4.1 The Open-Source VP8 Video Codec versus H.264

The majority principle concepts of VP8 are very similar to H.264. VP8 is also categorized as a block based video compression method. VP8 is primarily designed to support Internet and web-based video applications. The video pixel input for VP8 is also made up of eight bits per sample of luminance and chrominance 4:2:0 formats, which are partitioned in a variety of video block sizes [55, 56].

- *Frame Types*

VP8 has only two frame types. A key frame is a frame where only intra prediction has occurred [56]. In other block based video compression method, key frames are known as I-frames. In contrast to key frames, VP8 P-frames have inter frame prediction. VP8 does not directly define Bi-directional frames, which are commonly known as B-frames. Unlike H.264, which defines 16 arbitrary reference frames, the coded frame prior to the current one (the last reference frame), the golden reference frame, and the altref (alternative reference) frame are the three primary reference frames for VP8 [55-57]. The reference frames enhance coding efficiency in many unique ways. For example, the golden frame can be used alone to maintain the background information areas where stationary objects exist while objects are moving on the foreground locations [56]. The background sprite can be recognized from information between the golden frame and the last frame. A background sprite is created by golden

reference frame to distinguish between background and moving object in the foreground. The golden reference frame is also used for error recovery in live video conferencing [56]. Noise-reduced prediction relies on the alternative reference frame to send similar frames to the decoder so that complexity at the decoder can be reduced [56]. Because B-frames are not implemented for VP8, the combination of golden frames and alternative frames are used together to achieve bidirectional motion compensation [56].

❖ *Intra frame Prediction in VP8*

The intra frame prediction modes for VP8 are constructed from three types of blocks: 4x4 luminance, 16x16 luminance, and 8x8 chrominance [56]. Intra frame prediction can be implemented on a whole MB, supporting four different modes for luminance and chrominance, horizontal prediction, vertical prediction, DC prediction, and true motion prediction [56]. Ten other different modes can be implemented for 16 4x4 sub-blocks (SB) of the luminance component. H.264 has three different modes to predict luminance depending on the SB sizes, while the corresponding chrominance blocks can be predicted based on the same mode [57].

❖ *Inter Frame Prediction in VP8*

Like H.264, the inter frame prediction methods for VP8 handle all SBs which have sizes that are the multiple of 4x4, ranging to a size of 16x16 pixels. The luminance and chrominance components are processed at the same priority. The prediction information includes the SB size, the motion vector, and the dissimilarity measure of the reference MBs [56]. The luminance is divided into 4x4 SB elements. The predicted

results of the chrominance components are derived from the average of the motion vectors from the relative luminance MB [56]. However, H.264 supports sub pixel accuracy at a resolution of one-quarter pixel, while VP8 offers 1/8 pixel accuracy for the luminance, scaling it at the sample rate for the chrominance [57, 58]. The inter frame prediction in VP8 usually uses one of the three previous frames as a reference frame. In case of different characteristics of an object in motion within a MB, an advanced inter frame prediction mode called SPLITMV can be implemented to achieve better prediction performance [56].

❖ *Transform Coding, Quantization and Rate Control Algorithm in VP8*

Transform coding in VP8 is similar to H.264 in that the 16x16 MB is partitioned into sixteen 4x4 Discrete Cosine Transformation (DCT) blocks [58]. The most significant coefficient of the DCT is located at the top left of the matrix. This first coefficient is called the DC coefficient, while the rest of the 15 coefficients are known as the AC coefficients [58]. Unlike H.264, the DCT computation specified in VP8 requires more multiplication operations rather than just adding, subtracting, and right shifting [60]. Like H.264, the quantization matrix in VP8 contains quantization parameters that are divided into the DCT coefficients to discard low order bits and coefficient-by-coefficient basis. In VP8, the quantization matrix is controlled by the value of the QP. According to the rate control specification in Section 2.2 of the VP8 specification, rate control and control of QP are closely linked. In VP8, it is possible to apply the rate control algorithm from H.264, making it possible to reuse hardware Intellectual Property (IP) [60]. However, the QP levels in H.264 are designed for 52 QP levels with a quantization step from 0.625 to 224. VP8 prefers to support 128 QP levels

with quantization steps from 4 to 284. To reuse the hardware IP, a fine-tuned mapping process is required to match the index between H.264 and VP8 [54]. Because the rate control schemes in VP8 use the same concept as in H.264, the chicken-and-egg dilemma still occurs in VP8 and all of the proposed solutions described in Section 2.2 apply to VP8 as well.

❖ *Entropy Coding in VP8*

The entropy coding process combines all the data from the DCT, predictions, and motion vectors into the lossless binary output files. VP8 uses a Boolean arithmetic coder (non-adaptive arithmetic coder) [56]. The symbol values in VP8 are converted into a series of Booleans using a pseudo Huffman Tree scheme [56].

In summary, VP8 is very similar to H.264 in the principal concepts of the video compression design and operation. The development of VP is based on less complexity in the intra-frame and inter-frame prediction algorithms, which reduces processor time, power consumption, etc. Several existing hardware implementations of H.264 can be adapted for use with VP8, such as an advance rate control scheme, a rate distortion optimization algorithm, motion estimation, and a prediction method [56]. However, many research papers show that H.264 and X264 provide the best performance in term of PSNR, SSIM, and size of the compressed data stream [56-60]. Experimental tests in wireless networks comparing H.264/AVC, H.264/SVC, and VP8 showed that H.264/SVC provided the best performance in terms of data compression, spatial complexity of the queue in memory, and data error recovery [59]. H.264 also has better perceptual video quality than VP8 at all resolution including at 720P due to the reduction of encoding complexity in VP8 [60]. However, the VP8 decoder can process

the VP8 encoded file about 30 percent faster than in H.264 at the same bitrate and in the same network environment [56].

2.4.2 The Open-Source VP9 Video Codec versus H.264

The VP9 video codec is designed to improve upon the earlier version VP8. The goal for the new project is to reduce the bitrate up to 50 percent compared to VP8 for any given resolution [61]. VP9 also aims to improve the video encoder for high definition (HD) video [61]. There are two profiles for VP9. The video pixel input for profile 0 supports the luminance and chrominance 4:2:0 format, while profile 1 has options to select the 4:2:0 format, the 4:2:2 format, and the 4:4:4 format [61]. The superblock (SB) feature in VP9 is a new, bigger video partition introduced in VP9, where the video block size can be 32x32 or 64x64 [61].

A new feature called segmentation is defined and benefits to static background. The segmentation combines all MBs that have a similarity in information together [62]. A new two-pass prediction strategy to improve frame prediction is also introduced in VP9 [62]. The first pass encodes all possible blocks with inter frame prediction. The results from the first pass are then encoded again with intra frame prediction in the second pass. The compound prediction is another prediction algorithm that combines two predictors together. For example, two inter predictors with the same mode are able to merge to generate a new predictor. To support HD video, better DCT performance is required. VP9 supports an 8x8 DCT (+2x2 second order), and a 16x16 DCT [62]. A new set of prediction filters is also under development for VP9 [62]. Non-interpolation and interpolation are two existing options for prediction filters.

Lossless transform coding for VP9 is based on the Walsh Hadamard transform [62]. VP9 improves the entropy encoding by develop a new predictive model and context. Google currently claims that VP9 improves compression and video quality up to 44 percent for HD video and up to 26 percent for SIF (352x240) resolution [61, 62].

The community is working actively on development of VP9 at the time of this writing [62]. The effort is concentrated on enhancement of HD video and video web server applications [62]. It is still unclear how VP9 rate control, will compare to VP8. However, increased complexity due to new features in VP9 may cause more power consumption, slow down the processor, and require more hardware [62]. With the introduction of new features, reusing existing hardware IP is infeasible with VP9, which increases the final encoder product cost. However, VP9 is an open-source CODEC. It can use without payment of a royalty fee. VP9 may only be appropriated for software encoder and web applications, while hardware encoders may ultimately still rely on H.264.

2.5 Summary

Cameras are extremely useful devices in the ITS network. The main use of camera images in the Oklahoma statewide ITS is for traffic monitoring by humans. The operators prefer to view clear images on screen and they should be able to identify and classify vehicles and objects in the images. The advent of digital entertainment and image processing theory has motivated the development of several measurement techniques based on camera images, such as vehicle detectors, speed sensors, snow sensors, etc. The effectiveness of these techniques is strongly dependent on the quality

of the images. There is a strong desire to define the standard for communication in the Oklahoma statewide ITS as the NTCIP standard. But NTCIP specifies the camera control protocol only and does not address video quality or video compression. The methods for transmitting video from the cameras along the highways to ITS consoles in the Oklahoma system, or more generally to the TMC, are adopted from the digital entertainment industry where block-based video compression techniques such as MPEG-2, H.264, etc., are used.

Rate control is an important component in block-based video compression algorithms that has a dramatic impact on video quality. Although rate control theory is not a new research topic in video technology, it is an interesting problem that appeals to many researchers who continue working on new solutions to improve video quality. Examples of strategies that have been proposed for video rate control include but are not limited to linear model prediction, frame skipping, SSIM-based methods, and others [26, 38, 39]. These strategies range in complexity from simple to extremely complex.

The Oklahoma statewide ITS network is a heterogeneous network including copper cable, fiber optic cable, and wireless communications. It involves links with various different bandwidth capabilities, including 20 Mbps wireless, 10/100 Mbps Ethernet, and the 1/10 Gbps fiber backbone. Interfaces between different network components from many manufactures can cause incompatibility issues. In addition, there are many multicast routing protocols and distribution schemes that may be used in the network for video stream delivery. Although the IETF is pushing for standardization of multicast protocols through cooperation among manufactures, it is impossible to ensure that every network components from a variety manufactures will be compatible

with all multicast routing protocols. Incompatibility of the multicast routing protocols on a heterogeneous network can cause latency, jitter delay, and intermittent working.

In this dissertation, I proposed an innovative adaptive video rate control technique to support video distribution in the Oklahoma statewide ITS network. Unlike in digital entertainment applications for ITS networks, the video frame rate is not as significant of a concern; however, the spatial quality of the frames that are transmitted is critical. Video quality can be improved by adaptively changing the frame rate based on real-time monitoring of network traffic and congestion through direct measurement of packet delays. This can be accomplished in a simple way by using the well-known internet ping test [63]. To reduce video data traffic on the ITS network, video streams with same content can be sent from a video server in both multicast and unicast schemes. User consoles should subscribe to the video stream using the multicast scheme first. However, if the multicast scheme is not successfully connected, the unicast video stream is an alternative way to receive video stream.

Chapter 3: Streaming Video Transmission

Internet Protocol (IP) is the principal multimedia transmission scheme in the Oklahoma statewide ITS network. Unicast, multicast, and multi-unicast are three primary schemes to transmit the encoded video between a video server and video clients over IP. To reduce data traffic and improve bandwidth utilization, the multicast scheme is the ideal communication option for ITS applications. However, the ITS network implementation integrates several link mediums having vastly different bandwidths and throughput capabilities. The network interface from several link mediums can cause multicast scheme malfunction due to incompatibility among network devices. For example, low bandwidth wireless equipment (Motorola Canopy) in the Oklahoma statewide ITS network, is intermittent to not deliver multicast packages to the high speed backbone network (fiber optic network). The 900 MHz radio modems which are used to implement many links in the Oklahoma statewide ITS network where fiber optic are not available are incapable of handling multicast packets.

3.1 An alternative Method to Enhance Streaming Video Transmission

In this chapter, I proposed a novel, simple, and effective solution to decrease video traffic in the network, reduce video connection errors, enhance efficient video delivery, and maximize system performance. Appendix G of the H.264/MPEG-4 AVC part 10 standard specifies the scalable video coding (SVC) technique [64]. The SVC can enable the subset bit-stream video to transmit a secondary video stream and primary video stream simultaneously. Thus, both unicast and multicast schemes with the same

video content can be transmitted to the clients from the same video server simultaneously. First, the video clients will subscribe to the video stream using the multicast address. However, if the connection with multicast cannot be achieved, the video clients will try instead to connect to the streaming video using the unicast scheme.

In the following section, I will show that the proposed method can be used to solve the aforementioned problems by setting up an experimental network to simulate various video transmission methods while measuring the data traffic.

3.2 Network Simulation and Experiment

I constructed an IP network (see Figure 2.) to simulate video transmission between the video server and clients in various video transmission schemes. The amount of video data transmitted is measured using a function from a network device (router/switch) known as the port utilization monitor [65]. In this experiment, the Summit S200-24 switch by Extreme Networks is the network device that is going to be used for evaluation [65]. The video connection is also monitored by a network protocol analyzer called WireShark [67], which is an open source program. The WireShark project was started in 1998 [67].

The Hauppauge model WinTV-HVR 9500 is the main hardware encoder used for this experiment [66]. Actually, the WinTV is a hardware encoder based on the MPEG-2 video compression standard [66]. The experiments with MPEG-2 show that it is possible to create multiple streaming videos simultaneously, not only compliant with the H.264 standard, but with other video compression standards as well. The VLC open-

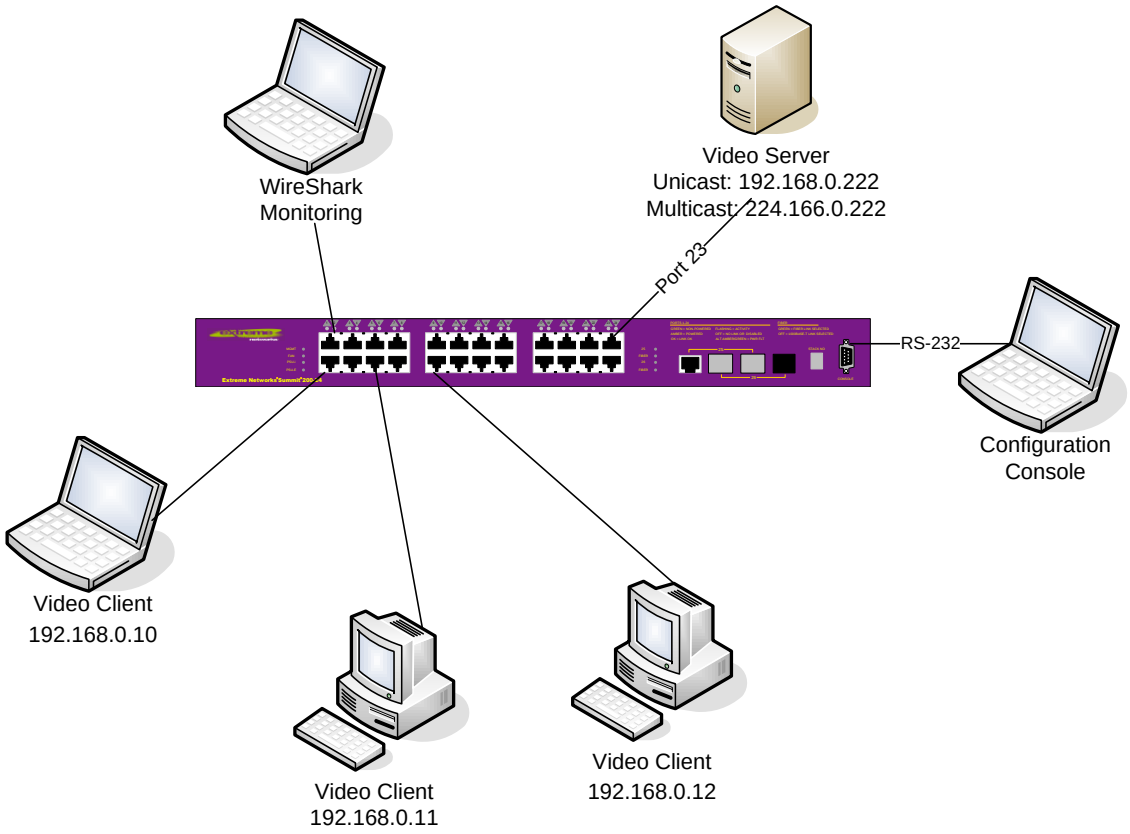


Figure 2. Network simulation setup for testing video transmission.

source media player by VideoLan is used as the video server and video client for this experiment [68].

3.2.1 Experimental Procedure

Figure 2 shows the network simulation setup for testing video transmission. The VLC video server machine is connected to a switch (Extreme Networks Summit S200-24) on port 23. The IP address for unicast is configured to 192.168.0.222, while the IP address for multicast is set to 224.168.0.222. In this experiment, the Real-time Transport Protocol (RTP) is used to stream video using the multicast scheme. RTP is

defined under the RFC 1889 by the IETF [69]. The User Datagram Protocol (UDP/IP) is a method for RTP to transmit either unicast or multicast data. However, RTP does not guarantee data delivery and has not been equipped with quality of service (QoS) functionality [69]. Unlike the multicast testing just described, the Real Time Streaming Protocol (RTSP) is used for unicast testing. The RTSP is defined under RFC 2326 by the IETF [69]. RTSP works as a network remote control to retrieve streaming video from the video server. RTSP can support either unicast or multicast schemes. Unlike RTP, RTSP is designed to work with either UDP/IP or the Transmission Control Protocol (TCP/IP) [69].

The following procedure was used to configure and set up the VLC program for streaming video:

- Run the VLC program.
- Click on the Media menu and select Stream as shown in Figure 3.
- Click on the Capture Device tab, select the appropriated device driver for the hardware encoder (WinTV HVR-9500), click on the Advanced Options and select an A/V channel on the Video input pin as shown in Figure 4.
- Click on the Stream menu.
- Activate transcoding and select the Video – MPEG-2 + MPEGA (TS) profile as shown in Figure 5.
- Specify the communication scheme in the appropriate dialog box. RTP was used for the multicast scheme while RTSP was used for both the unicast and the multi-unicast schemes. These will become something like Figures 6 to 9.

- Add the appropriate IP address, port, and video stream name as shown in Figures 6 to 9.
- Click next, put a check on all of the elementary streams, and select stream bottom as shown in Figure 10.

The configuration console shown in Figure 2 was used to configure the S200-24 switch to enable IGMP and IGMP snooping for the specific Virtual Local Area Network (VLAN). All of the working ports were configured to the default VLAN. IGMP needs to be enabled so that the video host can transmit the signal to the network where the members of multicast traffic are located. In addition, IGMP snooping is needed so that the switch can interpret the IGMP messages from the video server and forward the multicast packet to the member ports. The following essential commands were used to configure the switch for this experiment.

- IGMP was enabled using the command

  “enable igmp {vlan <vlan name>}”

  Example: Summit200-24:1#enable igmp vlan default
- IGMP snooping was enabled using the command

  “enable igmp snooping {vlan <vlan name>}”

  Example: Summit200-24:1#enable igmp snooping vlan default
- The data traffic on port 23 can be viewed, for example, by using the command

  “show ports {<portlist>} utilization”

  Example: Summit200-24:1#show port 23 utilization

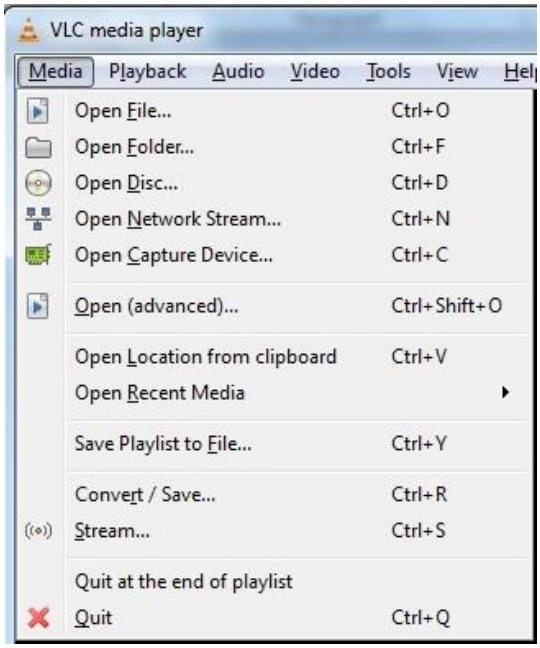


Figure 3. Show stream menu choice.

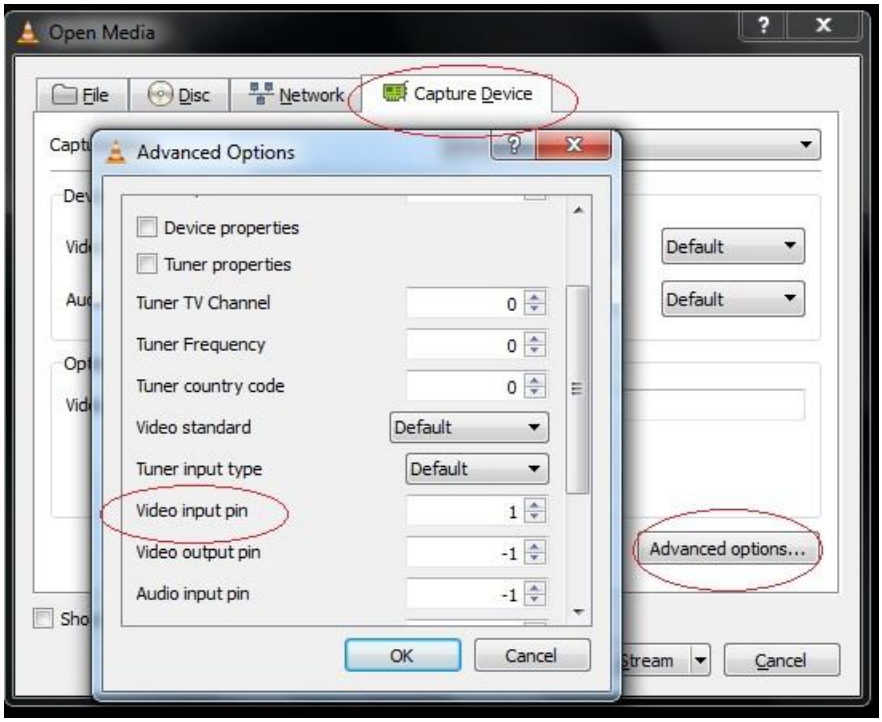


Figure 4. The device driver and video input configuration.

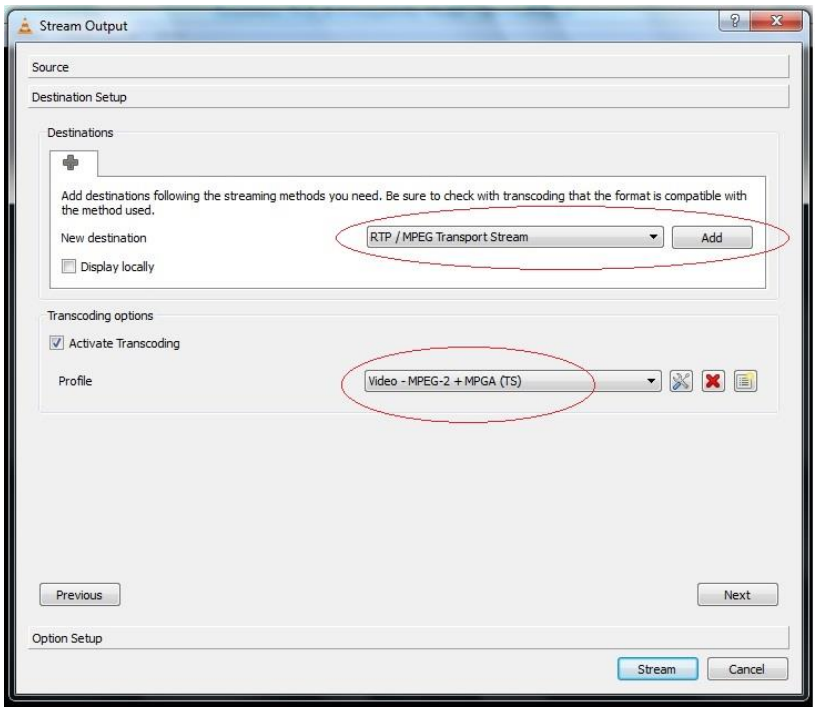


Figure 5. The transcoding activation.

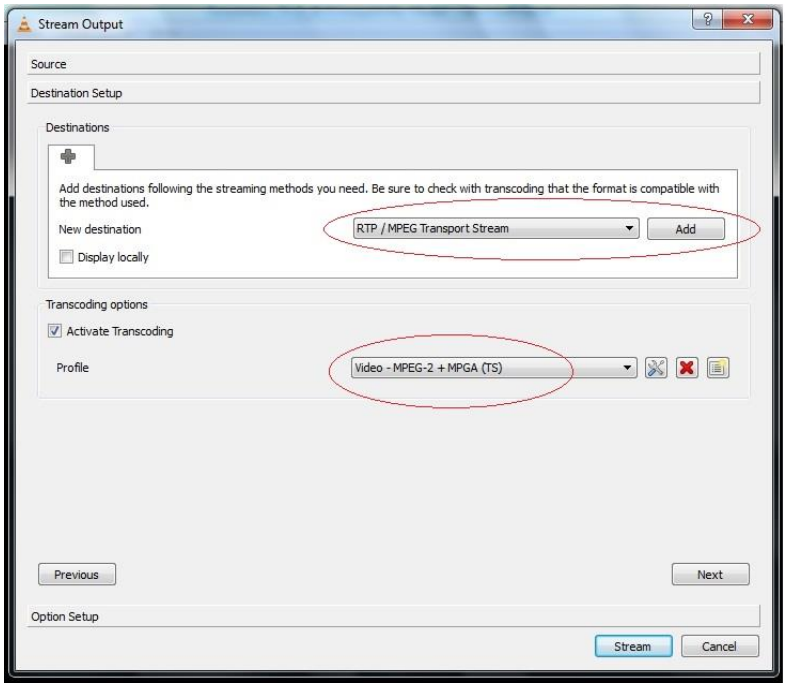


Figure 6. RTP configuration for the multicast scheme.

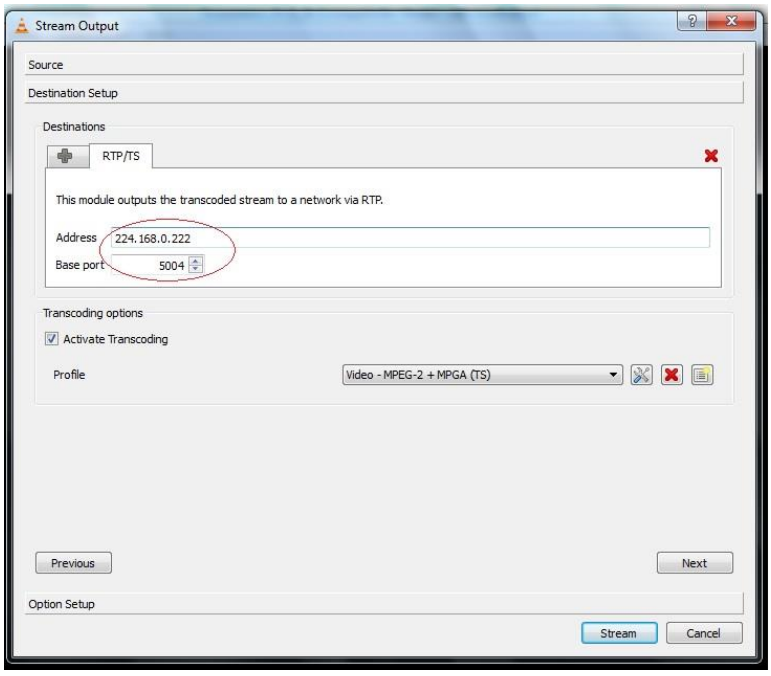


Figure 7. Multicast IP address and port configuration.

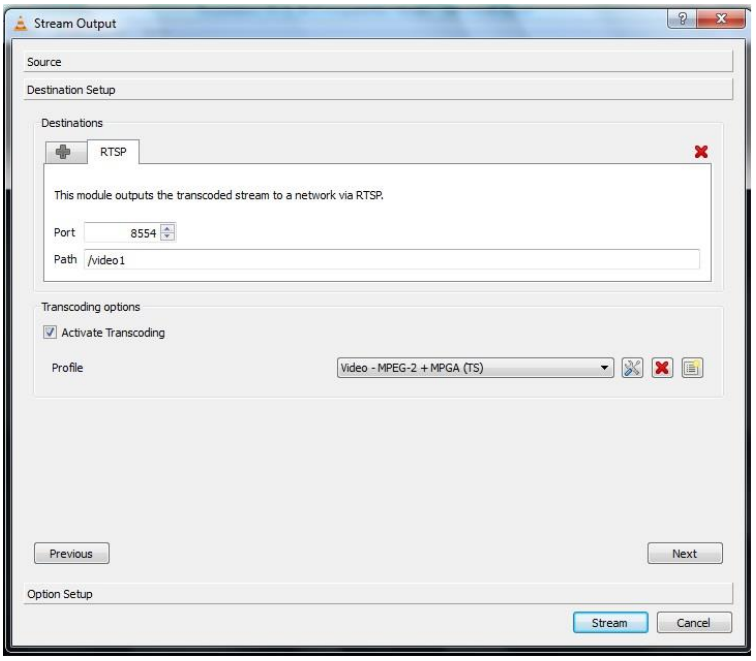


Figure 8. RTSP configuration for the unicast and multi-unicast schemes.

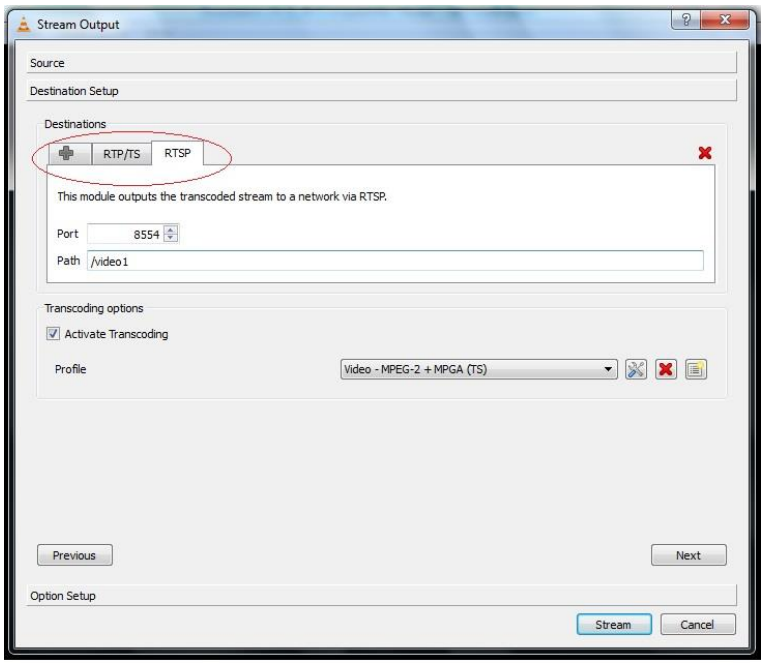


Figure 9. Both RTP and RTSP can be configured simultaneously.

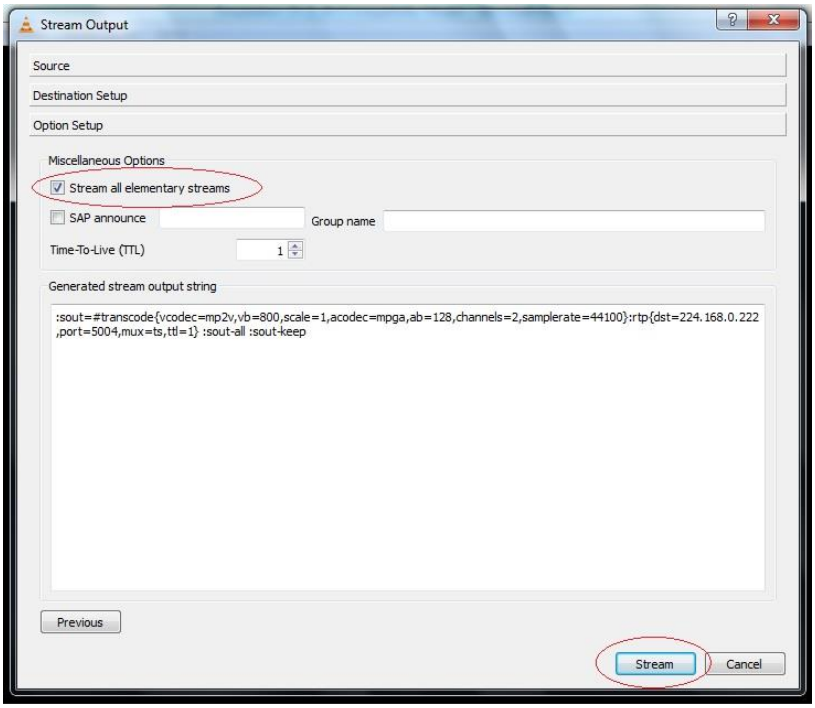


Figure 10. Enabling all elementary streams

3.2.2 Experimental Results and Discussion

I conducted the experiment by enabling streaming video for the unicast, multicast, and multi-unicast schemes as in Table 1. The multi-unicast was simulated by enabling two or more unicast video streaming. For example, the second row of Table 1, two computers with IP 192.168.0.10 and 192.168.0.11 enables streaming the unicast video. Streaming two videos with same content are sent out from the encoder. This makes the data utilization on port 23 at the last column of Table 1 increase more than the single unicast streaming video on the first row. An example of data utilization on port 23 is shown in Figure 11. For example, when the multicast scheme is activated and the unicast scheme is turned off, the data utilization is very stable at about 29 packets per second regardless of the concurrent number of clients as shown in the Figure 11(a). When multicast is turned on but the client connection is off, the data utilization is dependent on the number of concurrent unicast clients plus the multicast data stream as shown in Figure 11(b).

In conclusion, the data utilization fluctuates depending on the number of clients where unicast or multi-unicast is enabled. The multicast scheme can dramatically reduce the amount of traffic on the network. It is comparable to television broadcasting where the television station propagates the signal through the air interface to multiple viewers without limiting the number of viewers.

Examples of the network protocol analyzer (WireShark) results are shown in Figures 12 and 13. In Figure 12, the UDP video data is originated from the video server IP 192.168.0.222 and transmitted to the destination at multicast IP 224.168.0.222. Because the multicast scheme is a point to multi-points topology, explicit initialization

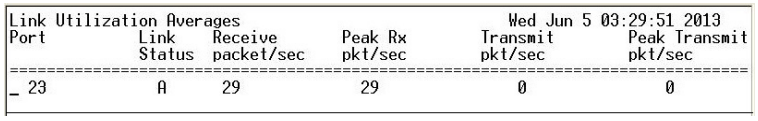
of the video connection for each individual client is not necessary. The established connection from the unicast video source 192.168.0.222 to the destination 192.168.0.11 is shown in Figure 13. Some of the RTSP messages are being sent between the video source and client while alternative multicast packets appear from time to time. This occurs with transmissions that use scalable video coding functionality.

3.3 Summary

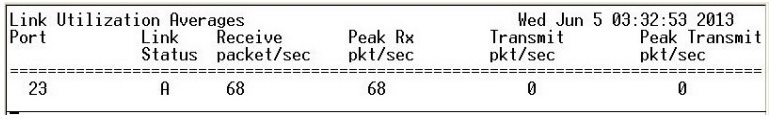
Data traffic on the IP network can be reduced by using the right video transmission scheme. The multicast scheme is the ideal video transmission method that provides the optimum bandwidth utilization regardless of the concurrent number of video clients. However, incompatible interfaces in the ITS network can prevent the multicast transmission scheme from working successfully. I prefer to enable the scalable video coding option when multiple videos of the same content are sent from the same video server simultaneously. The video client is first designed to subscribe to streaming video by using the multicast scheme. When the multicast connection scheme is not achieved, the unicast or multi-unicast scheme can be served for video transmission. Based on the experimental results, the data traffic can be reduced tremendously with my proposed method.

Table 1. Network data utilization in packets per second for several unicast/multicast combinations.

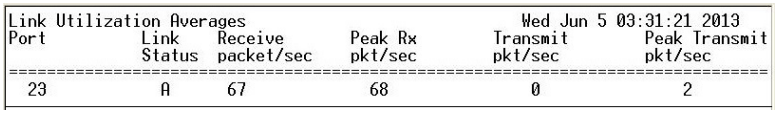
Encoder		Computer	Computer	Computer	Port 23
Unicast	**Multicast**	**192.168.0.10**	**192.168.0.11**	**192.168.0.12**	**Data Utilization (packet/Sec)**
On	Off	Unicast	Off	Off	68
On	Off	Unicast	Unicast	Off	107
On	Off	Unicast	Unicast	Unicast	164
Off	On	Multicast	Off	Off	29
Off	On	Multicast	Multicast	Off	29
Off	On	Multicast	Multicast	Multicast	29
On	On	Unicast	Unicast	Multicast	140
On	On	Unicast	Multicast	Unicast	139
On	On	Unicast	Multicast	Multicast	100
On	On	Multicast	Unicast	Unicast	140
On	On	Multicast	Multicast	Unicast	98



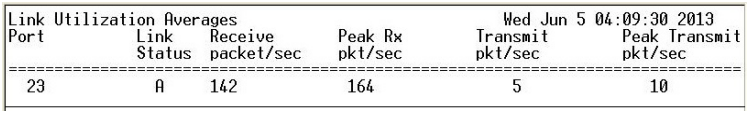
(a)



(b)



(c)



(d)

Figure 11. The example of the data utilization from port 23. (a) Multicast(on/clients)/unicast(off) (b) Unicast(on/1 client)/multicast(on/0) (c) Unicast(on/1 client)/multicast(on/client) (d) Multicast(on/0)/unicast(on/3 clients)

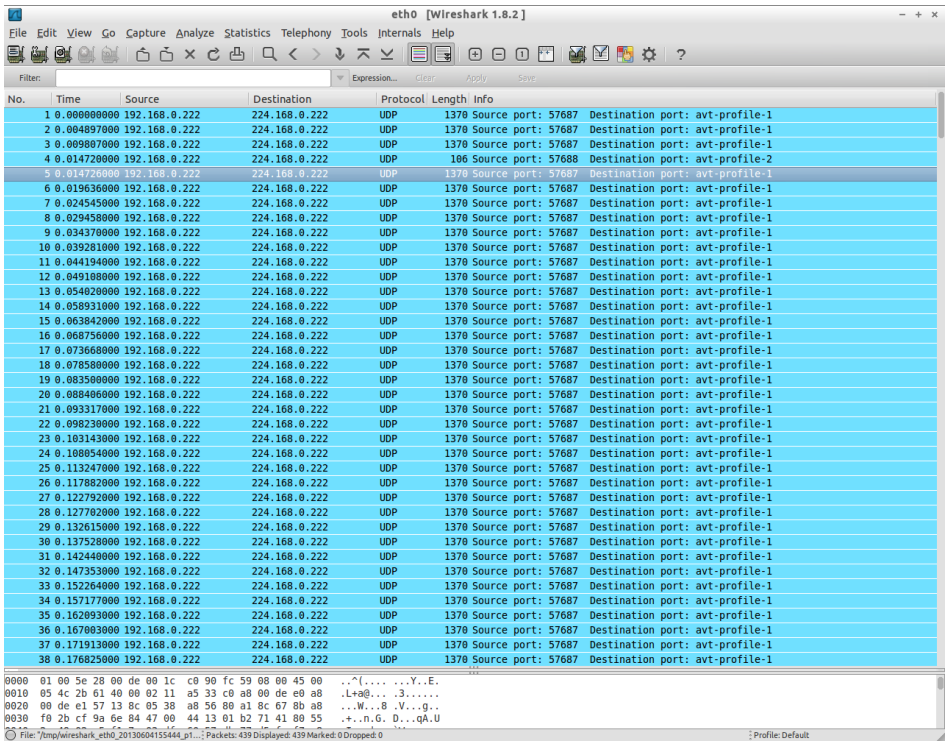


Figure 12. WireShark results with multicast.

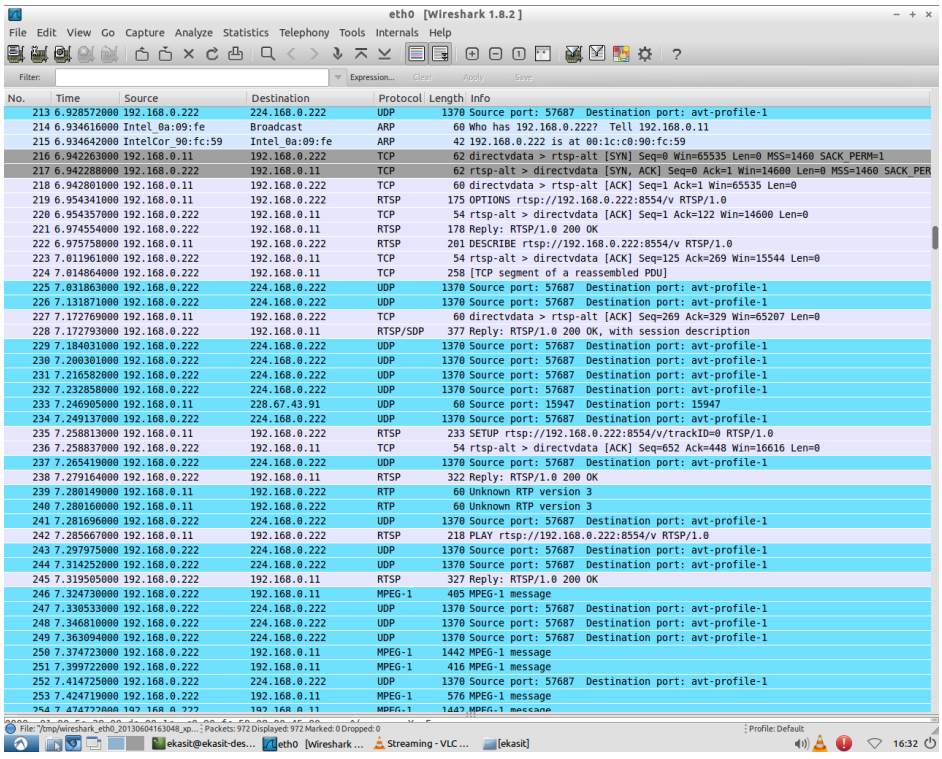


Figure 13. WireShark results with multicast and unicast.

Chapter 4: The Video Frame Rate Experiment

In this chapter, I present my proposed solution for adaptive frame rate control by skipping video input frames based on network congestion as quantified by packet delay times ensured in real time by the ping test. All of the tools used in the experiments are based on open-source software. The network simulator is the Wide Area Network Simulator (WANem) version 3.0 [70]. The video quality evaluation is done by the MSU Video Quality Measurement [74]. The ping test and the proposed video frame rate controller are two principle experiments for this chapter.

4.1 The Ping Test Network Tool Experiment

A ping is a network administrator tool to send test messages from a source machine to a specific machine across the IP network [72]. This tool can be used to measure time travel between the machines, which is known as the round-trip time, normally reported in milliseconds. The source machine operates the ping by sending an Internet Control Message Protocol (ICMP) packet with echo request to the target machine. The source machine then waits for the reply message. For some reason, if the source machine does not get any reply within the specific interval time, then a request time out will be issued. However, when the source machine receives a reply message from the destination machine, then the transmission time and the data packet received will be analyzed. The following information is then reported: data packet conditions such as data loss, data duplication, data corruption, data delay, etc.

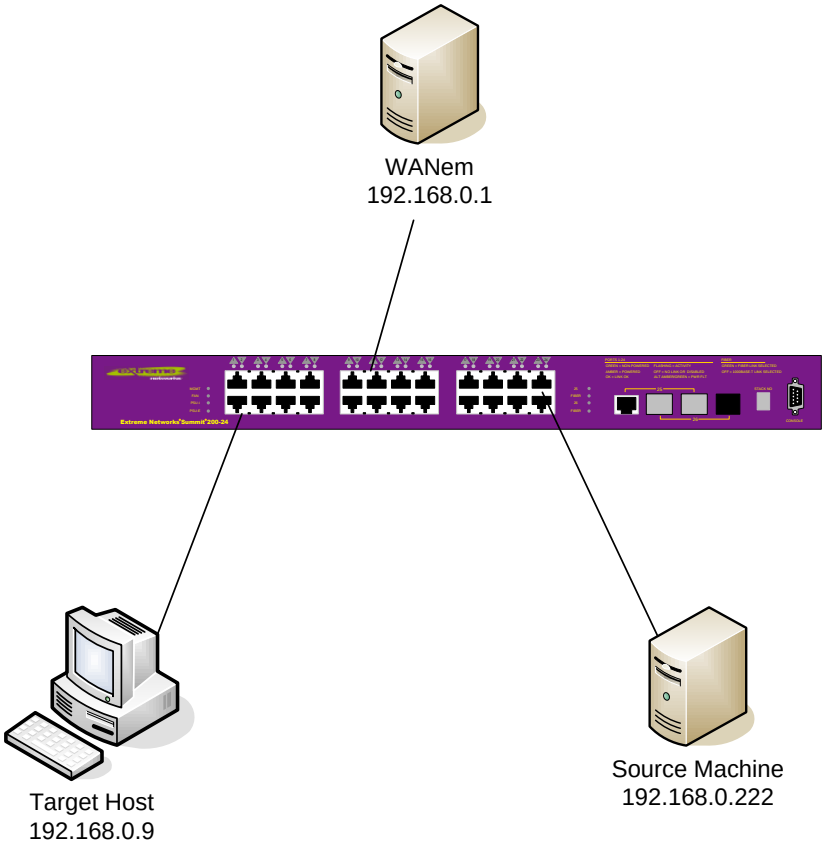


Figure 14. Network diagram for the ping test.

The WANem network simulator that I used in this experiment is a CD-ROM version and it does not require software installation or hard drive space. The network testing diagram is shown in Figure 14. The data traffic was routed from the source machine to the target by passing through the WANem computer. In this experiment, switch configuration was not required. The switch was left in the default configuration.

To control data traffic, a specific route needs to be added to the routing table. The command lines to add a new route to the routing table are based on the operating system. For example, if the operating system of the source machine is Windows, the command line is "route add 192.168.0.9 mask 255.255.255.255 192.168.0.1". The routing table at the target host needs to be configured to be the same as the routing table

at source machine. If the operating system for the target host is Windows, the command line is “route add 192.168.0.222 mask 255.255.255.255 192.168.0.1”. However, when the operating system is Linux, the command line will be a little bit different. For example, for the source machine the command line is “route add –host 192.168.0.9 netmask 0.0.0.0 gw 192.168.0.1”, while for the target host it is “route add –host 192.168.0.222 netmask 0.0.0.0 gw 192.168.0.1”.

In this experiment, several typical network problems were simulated by configuring WANem, such as data packet loss, data packet reordering, data delay, data duplication, and jitter. In the experiment, each ping test sent the ping test packet to the destination machine 100 times with buffer size 1000. Example results from these tests for various simulated network problems are shown in Appendix A.

From the test results, the ping test can be used to monitor network congestion and data errors. For example, the results of data request numbers which reply back from ping are disordered indicating that data packet reorder occurred. The network media types are able to be determined from the ping results as well. For example, fiber optic media can provide more network bandwidth and less delay so that the round trip delay measured by ping should be less than 1 millisecond. Wireless links can be identified because they have higher jitter. Jitter can be identified from the fluctuation of the round trip delays measured by ping. Data duplication is indicated by the DUP message if the Linux operating system is used. At the very end of the ping results, the summary and the ping statistic are also useful information. The average results and time can be used as a reference input for adaptive rate control. In the next section, I will explain any

proposed adaptive rate control strategy and present the results of performance tests for different frame rates in a variety of network environment.

4.2 The Adaptive Video Frame Rate Control Experiment

The rate control algorithm is one of the significant parts of any block based lossy video compression technique. As reviewed in Chapter 2, there are many different approaches to rate control ranging from simple schemes to complicated processes. I have already explained the experimental procedure and discussed the results of the ping test in the Section 4.1. In this section, I describe performance testing of the proposed adaptive rate control strategy in terms of several widely accepted perceptual video quality metrics. My proposed adaptive rate control strategy is based on dynamic adjustment of the input video frame rate.

4.2.1 Proposed Rate Control Strategy

Two of the significant factors that have an influence on perceptual video quality are moving objects and changing scenes. For example, football players exhibit a lot more movement than news reporters. With the same video encoder, the perceptual video quality at decoders when video input is football players may not perfect compared to video input of new reporters. In the ITS application, most of the pixel errors appear on the moving vehicle areas. The stationary areas such as the guard rail and median lanes usually do not suffer from pixel errors because those areas are unchanging, as shown in Figure 15. Consequently, if the input video frame rate is reduced dropping frames does not make less movement. In fact, it increases the movement that occurs between frames



Figure 15. Illustration of pixel errors in actual video from the Oklahoma statewide ITS. The errors are usually rejected to moving object such as vehicles.

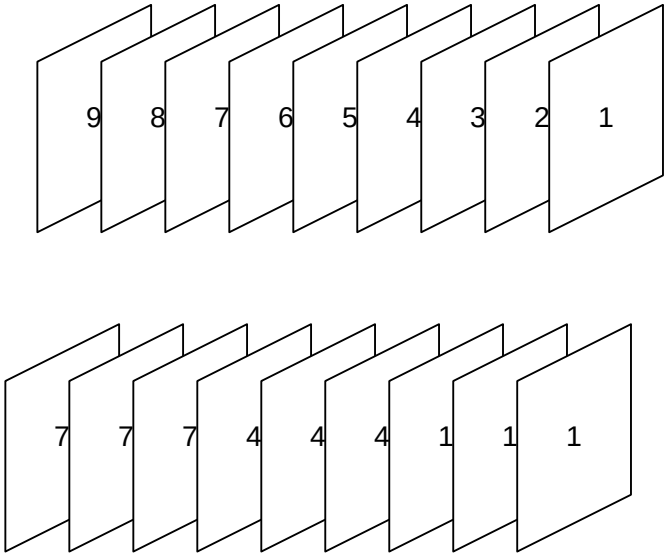


Figure 16. Rate control scheme for dropping and repeating frame. Top: original video stream at 30 fps. Bottom: the rate controller drops two out of every three frames.

(when there is any change). But it does reduce the number of different positions that must be encoded for each object. And repeating the retained frames results in many cases where there is no motion between consecutive frames, the perceptual video quality should be improved.

Typically, a camera location in the Oklahoma statewide ITS network has four fixed direction pointing along north, south, west, and east bound traffic. One or two pan-tilt-zoom (PTZ) cameras are installed on each camera location. The PTZ cameras are occasionally steered for incident surveillance but remain stationary most of the time. The video images from cameras obviously consist of moving vehicles against a mostly stationary background. Therefore, a video frame dripping scheme can be used to dramatically reduce the number of encoded information on moving vehicles without incurring any appreciable loss. For backward compatibility with hardwired equipment at 30 fps, the proposed video rate controller maintains the minimal overhead by repeating each retained frame. Because every MB in a repeated frame has a perfect match, a motion vector is zero. Examples of basic frame droppings are shown in Figure 16. Two out of every three frames are dropped and the retained frames are repeated three times. Three frame dropping modes are implemented: 30 fps (no frame dropping), 15 fps (drop 1 out of 2 frames), and 30 fps (drop 1 out of 5 frames). The simple custom circuit module which is a hybrid analog and digital circuit can be designed to handle the proposed video dropping strategy.

4.2.2 Video Quality Testing Procedure

Performance of proposed rate control strategies were evaluated using the setup network as shown in Figure 17. The input video on experiment was acquired from the real interstate highway (I-35) traffic in the Oklahoma ITS. The open-source WANem network simulator was used as a network simulator for this experiment. Experiment procedures will be shown in the following steps.

- Video Input Preparation

The analog video is recorded for 10 minutes by using a Panasonic model AG-1970 broadcast quality video cassette recorder. Because the existing analog video is based on the National Television System Committee (NTSC) specification, the recorded video has a frame rate of 30 frames per second (fps). The analog video is converted to digital video at frame rates 5, 15, and 30 fps respectively with SmartCapture, a USB based H.264 hardware encoder device [71]. The digital videos with different frame rates are used as the original reference videos for later comparison. They are also recorded back to analog videos with the same video cassette recorder for use as a video input to the encoder.

- The Benchmark Setup Diagram and Testing Procedure

The testing setup for this experiment is shown in Figure 17. The analog input that feeds to the encoder is played back from the VHS player. The USB based H.264 hardware encoder is configured for a fixed resolution of 640x480 pixels, bitrate of 1.01 Mbps, frames rate of 30 fps, and I-frame interval of 1, as shown in Figure 17. The

network simulation path is the same as in Figure 14, where the USB hardware encoder is connected to the source machine working as the video server. The target host is the software decoder and video recorder. The video traffic is routed from the video server to the WANem machine. The data bandwidths and other network resources are restricted by the WANem software before streaming video is passed through to the target host working as the video client.

Three different network media types were simulated in this experiment: the T-1 or DS-1, which has a typical bandwidth of 1.544 Mbps, the thin Ethernet or CAT-3 cable, which provides up to 10 Mbps bandwidth, and fast Ethernet which has a bandwidth of 100 Mbps. All the network simulations were capable of supporting a larger network bandwidth than the configured bitrate in the encoder. However, the packet limit is one of the significant parameters for the routers or the switches in any network. The routers or the switches will drop data packets that exceed the packet limit. To simulate realistic network conditions, the packet limit was configured to 10,000 bytes for the T-1 links, 10,240 bytes for the 10 Mbps links, and 1,000 bytes for the 100 Mbps links, as shown in Figure 19.

The WANem network simulator can simulate several network errors and traffic congestions. Data delay (or latency) is the proportional round trip time from source to destination. For example, 500 ms second delay is a common round trip time on a wireless or unstable network. The typical delay time on fiber option network is 1 ms or less. Data loss which is a common problem on the real network represents the data packet dropped due to network congestion or weak signal. Data duplication simulates

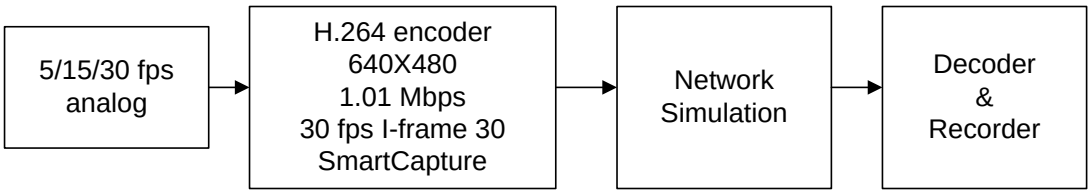


Figure 17. Benchmark diagram for the perceptual video quality experiment.

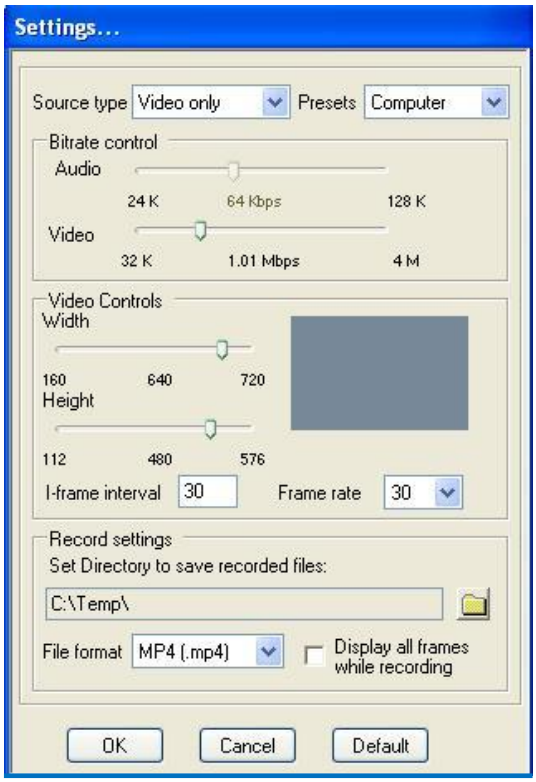
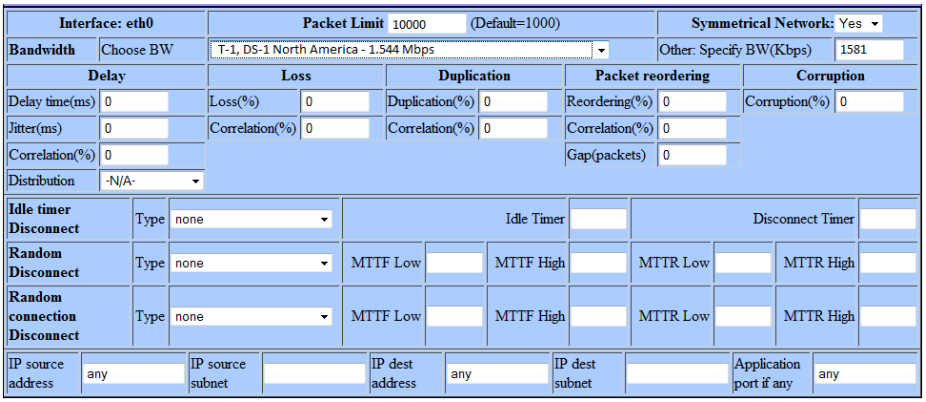
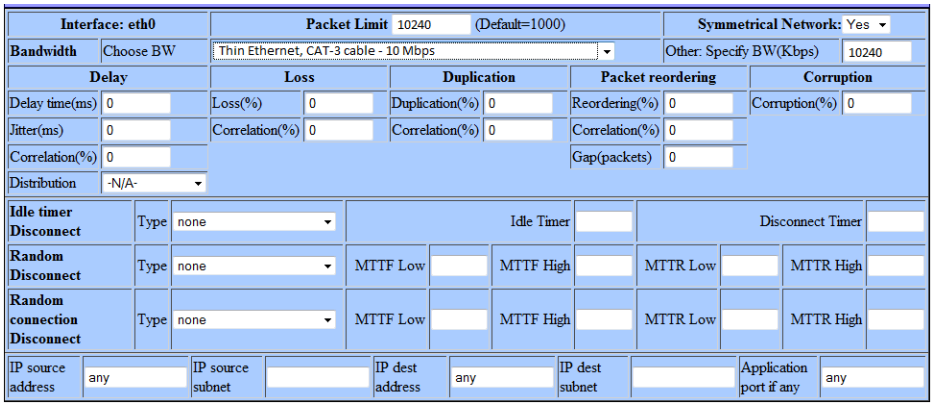


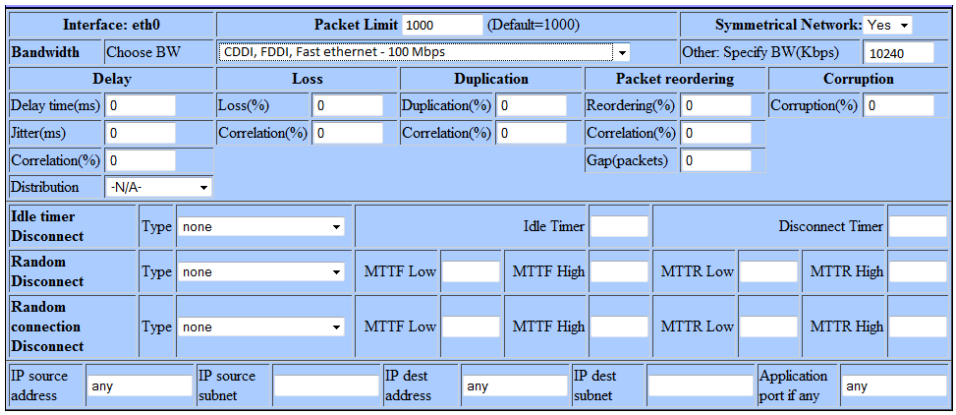
Figure 18. H.264 encoder configuration for perceptual video quality testing.



(a)



(b)



(c)

Figure 19. Packet limit configurations for the WANem network simulation. (a) The packet limit for T-1 links was configured to 10,000 bytes. (b) The packet limit for 10 Mbps links was configured to 10,240 bytes. (c) The packet limit for 100 Mbps links was configured to 1,000 bytes.

the multiple time of a single data packet sent by source to destination. Data corruption usually happens on any network due to the transmission errors, noise, signal attenuation, etc. Packet reorder indicates the unstable network transmission. Source normally sends data packet in sequence. The reordering causes data to arrive out of sequence at the destination. Jitter shows variation in the delay. The jitter is usual problems on the unstable network connection, like wireless connections. The jitter usual comes with data delay. For example, suppose the round trip time on a network is 100 ms. When the 10 ms jitter, the new round trip time can be 110 ms or 90 ms [64].

As described in Chapter 1, fiber optics is a backbone network for the Oklahoma statewide ITS. The practical cost for fiber optics is high and most of the time for network installation requires digging and trenching. It is impossible to deploy fiber optics to the entire network. Lower bandwidth alternative technologies such as point-to-point microwave links, 900 MHz radio, and CDMA modems are deployed. To simulate the Oklahoma ITS network, the WANem network simulation has to be setup to cover the list below.

- The data delay 500 ms
- The data corruption at 5, 10, 30, and 50 percent
- The data loss at 5, 10, 30, and 50 percent
- The data duplication at 5, 10, 30, and 50 percent
- The jitter 500 ms with the data delay 300 ms
- The data re-order with 300 ms delay time at 5, 10, 30, and 50 percent
- The random errors

❖ Result Editor and Analyzer

After the network simulation experiments were completed, the recorded video results and the original reference videos were edited to align the starting and stopping points. For the video editor, I used the Aimersoft Video Editor Free Trial version 3.0.0 [73]. The final video lengths after trimming at the beginning and the end of the videos are about 8.99 minutes or 16,186 frames for each video file. I would like to clarify that the number of video frame for each experiment are the same because the video frame rate will be changed at the analog video input while the configuration of the video compression does not change (retained and repeated frames).

The video results from each network condition that was tested were then analyzed using the MSU Video Quality Measurement Tool version 3.0 [74]. This tool for video quality evaluation was developed by the Computer Graphics and Multimedia Laboratory of the computing mathematics and cybernetics department at the Moscow State University located in Moscow, Russia. I evaluated the video results by comparing the video sequences received at the destination host to the original reference videos. However, the cross comparison between the video frame rates is unavailable. I evaluated the video quality using three widely accepted video quality metrics: the PSNR (256) [74], the SSIM (precise) [37], and the VQM [44].

4.2.3 The Experimental Result Analyzer

The experimental results from the MSU video quality measurement tool were saved into Comma Separated Value (CSV) format files. Figure 16 shows the configuration page of the MSU program and the CSV output file options.

The CSV result files were opened using Microsoft Excel. Statistical analysis was performed using the descriptive statistics function provided by Microsoft Excel. The data analysis procedure was begun by selecting the Data tab on the top of the Excel program and selecting Data Analysis. This opened a small dialog box showing several statistical functions, including the Descriptive Statistics. When the Descriptive Statistics function was selected, another dialog box opened. The raw data containing the 16,186 recorded video frames was then filled into the Input Range box. The Summary statistics and the Confidence Level for Mean boxes were checked. In this case, the 95 percent confidence interval was determined as shown in Figure 21. I selected the average (AVG), minimum (MIN), maximum (MAX), standard deviation (SD), and 95 percent confidence level with lower limit and upper limit.

Average values from the test results for PSNR (dB), SSIM, and VQM are reported in Table 2. Better video quality is indicated by higher PSNR and SSIM value and lower VQM value. In heavy traffic and network errors, the proposed rate control can provide better video perceptual quality at a lower frame rate of 5 fps as expected. Examples of three capture frames in case of data loss over a thin Ethernet link can be shown in Figure 22. The proposed rate control of 5 fps in Figure 22 (a) was eliminated errors by dropping five of every six frames. Some image degradation appeared in Figure 22 (b) when the proposed rate control was being obtained at 15 fps by dropping one every two frames. The video encoded without rate control was shown in Figure 22 (c) which was an example of unacceptable video images on the Oklahoma ITS. The full test results are given in Appendix B.

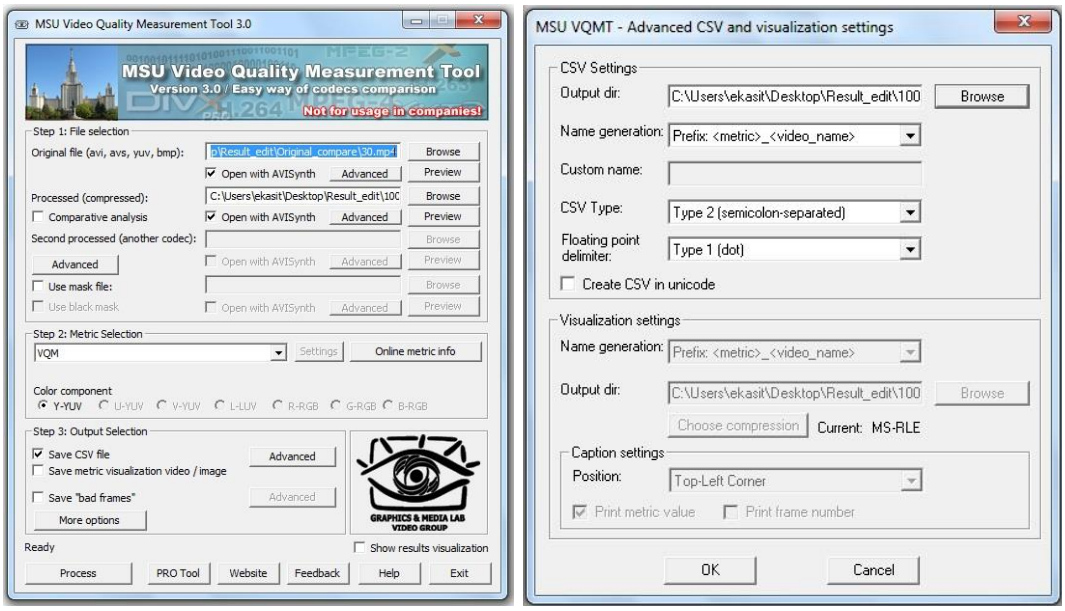


Figure 20. Configuration of the MSU Video Quality Measurement Tool and setup of the CSV output files.

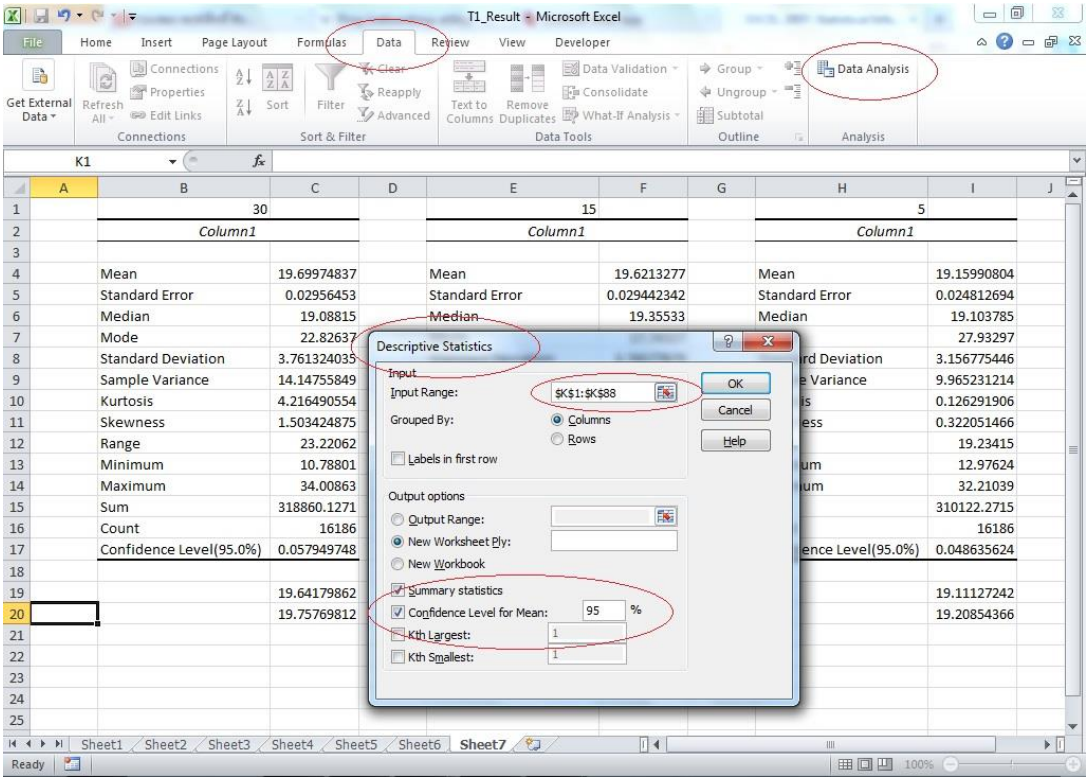


Figure 21. The descriptive statistics function to calculate the confidence interval



(a)



(b)



(c)

Figure 22. Decoded video frames received through a dropped packet network. (a) Proposed rate control method at 5 fps. (b) Proposed rate control method at 15 fps. (c) No rate control (30 fps).

Table 2. Video quality measured between original and received video streams for various network errors.

		T-1			10 Mbps			100 Mbps		
		5	15	30	5	15	30	5	15	30
Corruption	PSNR (dB)	20.3583771	19.6885439	19.3963638	20.214435	19.3155867	19.2852263	19.8712841	18.98320391	18.32901221
	SSIM	0.84739383	0.83143327	0.82119868	0.8383039	0.8300944	0.80686325	0.83220245	0.80344781	0.77765531
	VQM	7.96158294	8.62398934	8.55207002	8.09814251	8.55337608	8.98966113	8.32123665	9.098441231	9.620535124
Loss	PSNR (dB)	20.3448794	19.6872605	19.5256949	20.3701646	19.6044127	19.6677701	20.2463727	19.35706351	19.147083
	SSIM	0.85314862	0.83367946	0.81728339	0.84318494	0.82400043	0.81731185	0.84384776	0.820321841	0.806371406
	VQM	7.93809265	8.56815825	8.64941369	7.99609769	8.70089819	8.58399162	8.01182422	8.902080307	8.997401955
Duplication	PSNR (dB)	20.0584977	19.912662	19.5274986	20.3816547	16.9114504	19.2270942	20.0223645	19.46050511	18.34526295
	SSIM	0.83831854	0.83991923	0.81718249	0.84688162	0.82067391	0.81153508	0.84134722	0.823506871	0.787429255
	VQM	8.19034663	8.47135465	8.68039096	7.97844743	8.73007854	8.94946972	8.20560008	8.630197051	9.685818471
Reorder	PSNR (dB)	20.0324659	19.5379476	17.2583636	19.7235885	19.2488132	19.1553656	20.5292588	19.41980622	18.89193305
	SSIM	0.84215971	0.83362466	0.82955794	0.82584022	0.81424092	0.80197968	0.85488618	0.826527852	0.811440334
	VQM	8.2039872	8.62677717	8.43877256	8.44920124	8.8613787	8.93685229	7.86352797	8.772122286	8.916249566
Dealy 500 ms	PSNR (dB)	19.7738815	19.3599959	19.1579891	20.0239936	19.9237204	19.5489578	20.5817898	19.65116198	19.57348138
	SSIM	0.83049798	0.82167616	0.81767462	0.84514365	0.82697166	0.82425475	0.85329714	0.830279516	0.811725281
	VQM	8.77318577	8.78128753	9.03264388	8.21275797	8.69507851	8.49558027	7.82019885	8.148633669	8.457791942
Jitter 500 ms	PSNR (dB)	19.6910643	19.8306605	19.3964959	20.7948669	20.0638464	19.5568407	20.7528784	20.02604233	19.33165541
	SSIM	0.83548749	0.83718913	0.80950663	0.8663319	0.8429614	0.81186156	0.86080523	0.826146728	0.820742397
	VQM	8.37016616	8.4594316	8.70540287	7.33002748	8.230335	8.78030505	7.68003823	8.946053101	8.141791753
Random	PSNR (dB)	20.2168313	19.5580545	18.9795666	20.3853526	19.8865242	19.6843247	20.3734141	19.45149143	19.50480561
	SSIM	0.84644427	0.83018032	0.80441308	0.84228922	0.86598729	0.86815178	0.8441412	0.827705596	0.816020543
	VQM	8.29024714	8.80916661	8.98437451	8.11788515	8.30545063	9.01282643	8.0453312	8.857157999	8.59569062

4.3 Summary

The experimental results show that PSNR, SSIM, and VQM can be improved by reducing the input video frame rate. Because both the spatial and the temporal video qualities can be enhanced based on the frame rate of the video input, the ping results between the video server and the video client can be used to regulate the frame rate of the video input so that the maximum output performance at the video client can be achieved. The proposed video rate control is simple and effective to reduce the video bitrate for the ITS applications. However, this rate control strategy is not suitable for video entertainment applications.

Chapter 5: The Exploration of the Hardware Integration

In this chapter, I will develop digital hardware designs that could be used in a video CODEC to implement the adaptive frame rate control proposed in Chapter 4. These designs are based on low cost Field Programmable Gate Arrays (FPGA) and the embedded Linux platform, a free and open-source operating system [75, 78]. I use the Atlys™ Spartan-6 FPGA development board by Digilent as the FPGA development platform. The Pentium Dual-Core computer is adapted to install an embedded Linux system for this design. I provide a comparison between two hardware implementation at the end of the chapter.

5.1 The FPGA Hardware Development

In this hardware design, the proposed adaptive frame rate control strategy for the video encoder is designed and developed using the open-source hardware and software. The Atlys™ development board is integrated with High-Definition Multimedia Interface (HDMI) input and output ports. The main H.264 hardware encoder is an open IP core from Zexia Access, Ltd [76]. The Lightweight TCP/IP stack (lwIP) is the principal open source network stack to support the TCP/IP protocol in this project [77]. The web-server is developed for the user management interfaces such as the IP address configuration, the video resolution, etc. The design progress and project results are discussed in the next section.

5.1.1 Video Input Module

A progressive High-Definition Television (HDTV) signal is the main input to the development board. A custom HDMI hardware core from the HDMI demonstration project by Digilent that is called hdmi_in is reused in the design [78]. The HDMI video input resolution that syncs with the digital video display is called the Extended Display Identification Data (EDID) [79]. The demonstration input core is designed to serve the display resolution at 720p (1280x720) with a hardcoded implementation. The EDID for display resolution at 720p (1280x720) is shown in Table 3. However, ITS applications do not usually require such high resolution images. As specified in the EDID standard [80], a lower image resolution of 640x480 pixels can be defined and implemented. The EDID for a fixed resolution at 720p of 640x480 pixels is shown in Table 4.

The HDMI video input is decoded using the hardware input core in the Verilog format (dvi_decode.v) from the application note 495 (XAPP 495) by the Xilinx incorporation [81]. The decoded video data is originally in the RGB888 format which provides 8 bits for red, blue, and green [78]. The input data video is temporarily stored at a specific memory address (DDR2 SDRAM) by using the custom hardware core from the XAPP 1136 which is called the Video Frame Buffer Controller (VFBC) [82]. The VFBC is designed to support a single video frame. Data storage for multiple frames with the VFBC is possible by changing the base address for each consecutive video frame. Because the development board has only 16 address bits for the memory, the input data format is converted to the RGB565 format by neglecting the Least Significant Bit (LSB) and storing the data into the DDR2 memory.

```
for(y = 0; y < ycoFrameMax; y++) {
        for(x = 0; x < lLineStride ; x++) {
           PData = Xil_In16(pFrame + x*2 + y*lLineStride*2);
           RData = (PData >> 8)&0xF8;
           GData = (PData >> 3)&0xFC;
           BData = (PData<<3)&0xF8;
           xil_printf(" %d %d %d", RData, GData, BData);
```

Figure 23. Illustrate a RGB single frame capture using For loop.

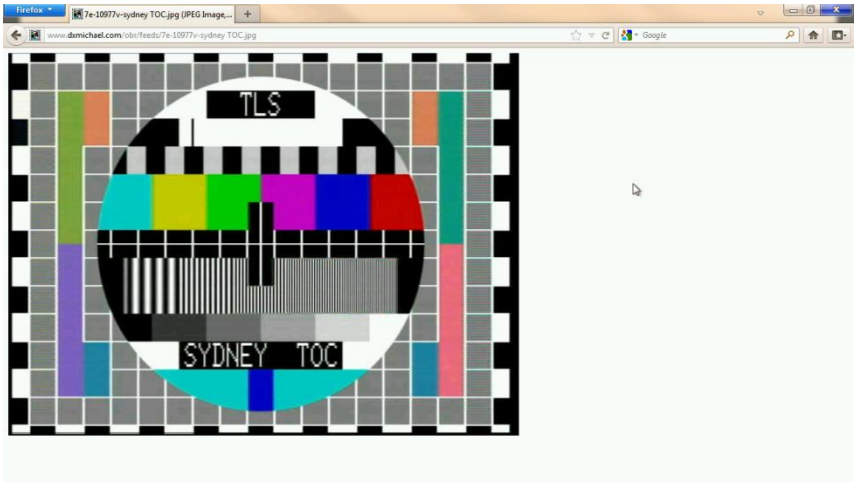


Figure 24. Sample of a single 720p video frame retrieved from memory

Table 3. Declaration for the high resolution EDID at 1280x720 pixels.

```
static const u8 rgbEdid[256] =
{0x01,0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0x3D,0xCB,0x81,0x07,0x00,0x00,0x00,0x
00,0x00,0x11,0x01,0x03,0x80,0x6E,0x3E,0x78,0x0A,0xEE,0x91,0xA3,0x54,0x4C,0x99,0x26,0
x0F,0x50,0x54,0x20,0x00,0x00,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x
01,0x01,0x01,0x01,0x01,0x01,0x1D,0x00,0x72,0x51,0xD0,0x1E,0x20,0x6E,0x28,0x55,0x00,0x
53,0x6F,0x42,0x00,0x00,0x1E,0x01,0x1D,0x80,0x18,0x71,0x1C,0x16,0x20,0x58,0x2C,0x25,0x
00,0x53,0x6F,0x42,0x00,0x00,0x9E,0x00,0x00,0x00,0xFC,0x00,0x54,0x58,0x2D,0x53,0x52,0x
38,0x30,0x35,0x0A,0x20,0x20,0x20,0x20,0x00,0x00,0x00,0xFD,0x00,0x3B,0x3D,0x1E,0x2E,0
x08,0x00,0x0A,0x20,0x20,0x20,0x20,0x20,0x20,0x01,0x5A};
```

Table 4. Declaration for the lower resolution EDID at 640x480 pixels.

```
static const u8 rgbEdid[256] =
{0x00,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x00,0x32,0xF2,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x13,0x01,0x04,0x80,0x00,0x00,0x78,0x0E,0x9E,0xCD,0x91,0x54,0x4C,0
x99,0x23,0x17,0x4D,0x50,0x00,0x00,0x00,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x0
1,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0xD8,0x09,0x80,0xA0,0x20,0xE0,0x2D,
0x10,0x08,0x60,0x22,0x01,0x80,0xE0,0x21,0x00,0x00,0x08,0x00,0x00,0x00,0x10,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0xFC,0x00,0x36,0x34,0x30,0x78,0x34,0x38,0x30,0x40,0x36,0x30,0x00,0x00,0x0
0,0x00,0x00,0x00,0xFD,0x00,0x3A,0x3C,0x1E,0x20,0x02,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x01,0x42,0x02,0x03,0x31,0x71,0x41,0x00,0x3E,0x0F,0x7F,0x07,0x
17,0x7F,0xFF,0x27,0x7F,0xFF,0x3F,0x7F,0xFF,0x4F,0x7F,0x00,0x57,0x7F,0x00,0x5
F,0x7F,0x01,0x67,0x7F,0x00,0x6F,0x7F,0x00,0x77,0x7F,0x03,0x83,0x7F,0x00,0x00,0
x67,0x03,0x0C,0x00,0x10,0x00,0xB8,0x2D,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x0
0,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0
x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x
00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x93};
```

According to the HDMI demonstration project documentation, the data at a specific pixel coordinate (x,y) can be retrieved by using the function below on the Software Development Kit (SDK) application.

```
Xil_In16(FRAME_BASE_ADDR+x*2+y*LINE_STRIDE*2)
```

The line stride is the number of pixels counted from the beginning of one video line to the beginning of the following video line. However, the retrieved data from memory are not in the RGB888 format. The data shift function with zero padding must be applied to transform the data back to the RGB888 format. With the looping code from the beginning of the video frame at the upper left hand corner (0, 0) to the end at bottom right hand corner, the video frame can be placed in memory in the raw RGB888 format. To compare the image from the memory and the input image, I have tested a single video frame by using the source code in Figure 23.

Appendix C shows the raw video data after it has been printed out to a file in the Portable Pixel Map (PPM) format. PPM is a basic format to represent color images. An example of a retrieved image is shown in Figure 24, which is at the 720p resolution.

5.1.2 The Video Data Partition and the Macro-Block

The input video in RGB888 format from Section 5.1.1 represents the pixel image information for the whole frame. It is inconvenient to process the entire data of

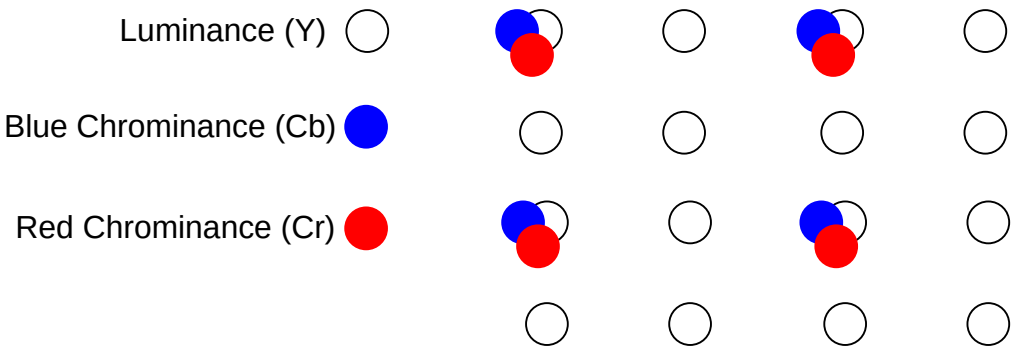


Figure 25. The 4:2:0 YC_rC_b sampling format illustrated for a 4x4 image.

the whole frame at the same time. Research on the human visual system shows that the human eye is more sensitive to luminance information than to the chrominance information [83]. The RGB color space can be converted to the luminance and chrominance (YC_bC_r) format [83]. Thus, the video input can be converted to the YC_bC_r format with resolution reduced to the 4:2:0 YC_bC_r sampling format, where each of the chrominance components is represented with only half the number of samples as the luminance component both horizontally and vertically, as shown in Figure 25.

To make the data easy to handle, the 4:2:0 YC_bC_r input data is partitioned into macro-blocks (MB) 16x16 of size pixels. For the H.264 standard, one MB of 4:2:0 YC_bC_r data is composed of one 16x16 Y block, one 8x8 C_r block, and one 8x8 C_b block. The MB data partition as described can be done by flowchart in Figure 26. The input data is imported to the 16 times loop. The luminance part is selected every time on the loop. The chrominance is processed on an event loop. In this case, I assumed that the video input data starts in the raster scan from the left scan line to the right scan line and from the top scan line to the bottom scan line.

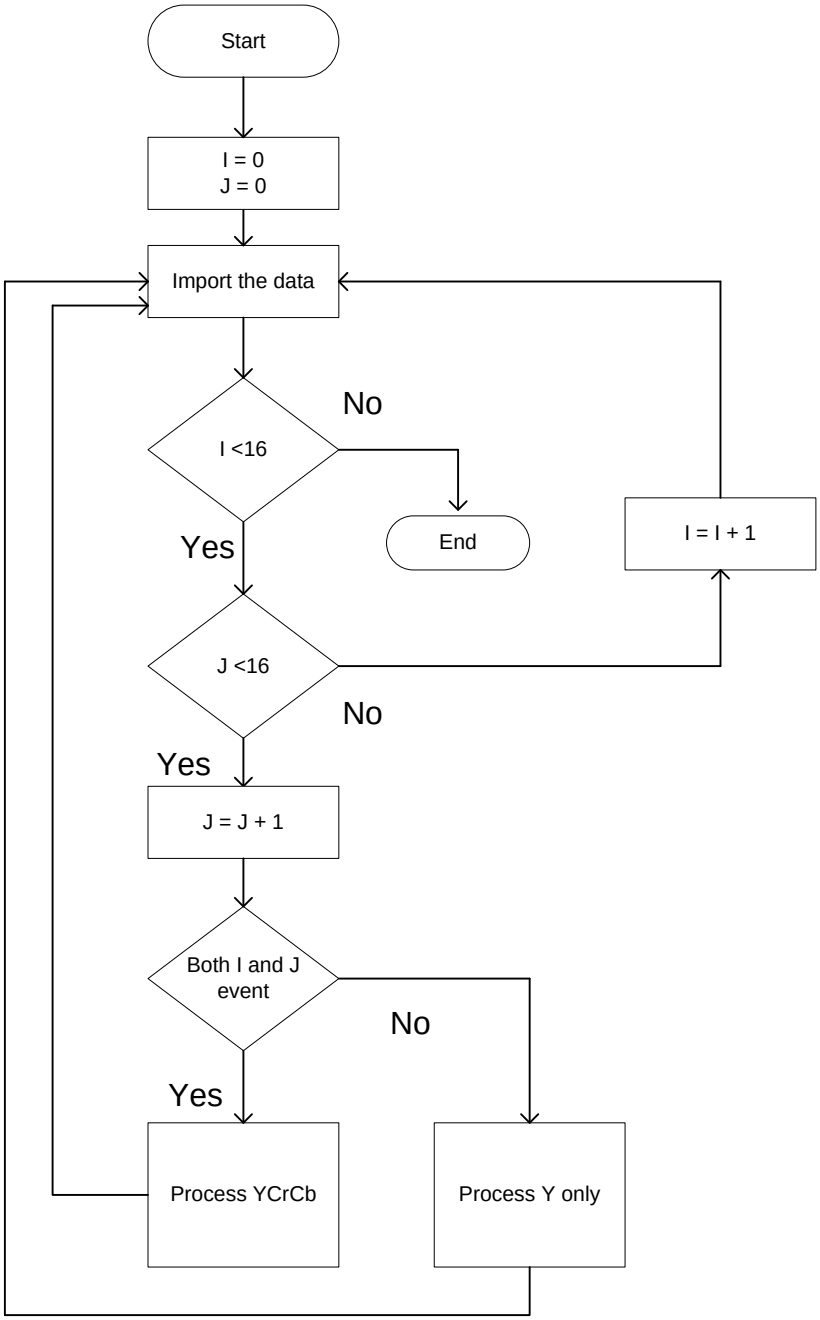


Figure 26. The flow chart for the MB data partition.

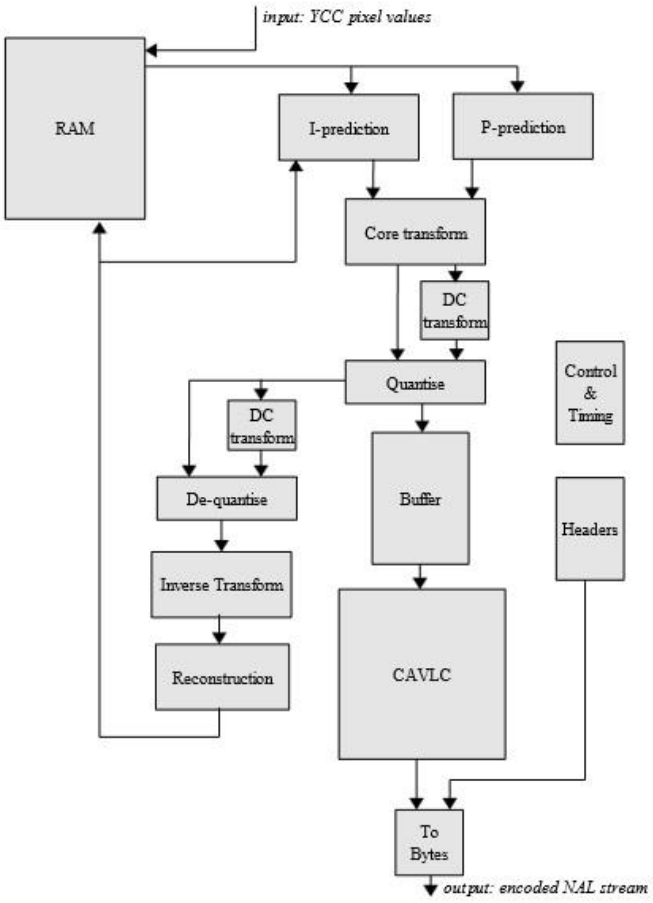


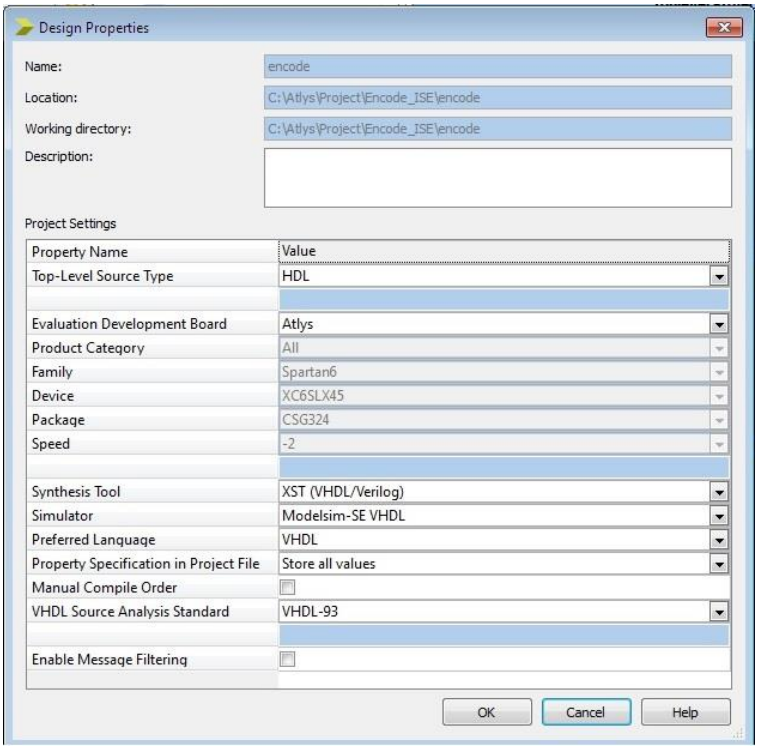
Figure 27. Block diagram of the H.264 video compression core (from [76]).

5.1.3 The Video Hardware Compression Core

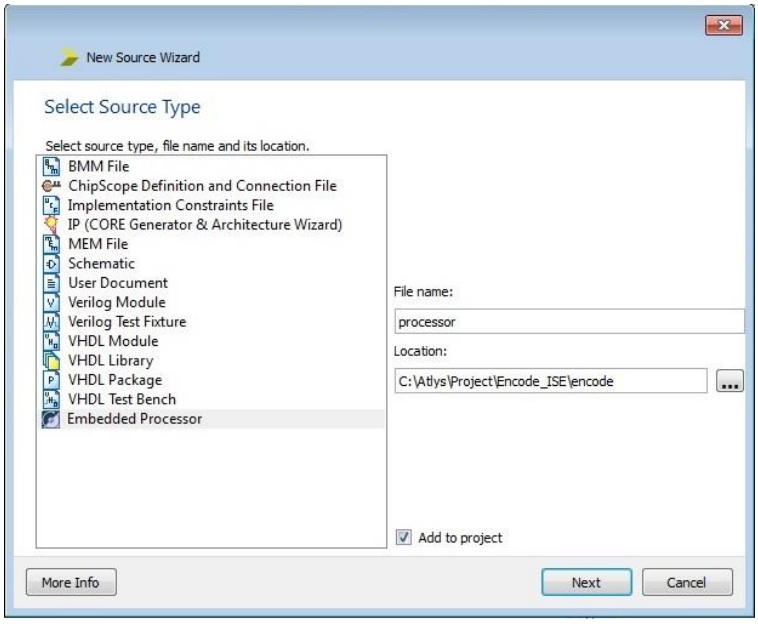
The video hardware compression core is an open-core IP, based on the VHSIC Hardware Description Language (VHDL) source code from Zexia Access, Ltd [76]. The input is imported from the video data partition and the macro-block path by using the Xilinx Platform Studio (XPS) General Purpose Input/Output (X-GPIO) interface [84]. The output stream from the hardware compression implementation is sent back to the lwIP TCP/IP stack by using another X-GPIO peripheral based at a different memory space address from where it can be distributed to the destination. A block diagram of the hardware compression core is shown in Figure 22.

5.1.4 Completed FPGA Hardware Encoder

MicroBlaze processor is a 32 bit soft processor core from Xilinx [68]. The FPGA Spartan-6 can accommodate designs with dual MicroBlaze processors [68]. With both dual processors enabled, they can share hardware peripherals and resources. The Xilinx Embedded Development Kit (EDK), a software designing tool for the embedded processing system, is the main design tool. The EDK can either work alone as the FPGA design tool or pass through the Xilinx Integrated Software Environment (ISE) as a managing FPGA design. In this case, both of the MicroBlaze processors are considered as a part of the overall system. It is better for the MicroBlaze processors to be designed under the ISE due to easy project management. The ISE software configuration for the Atlys™ board and the microprocessor subsystem are shown in Figure 28. The EDK is launched by the ISE software and allows configuration of the hardware. The Base System Builder (BSB) wizard was used to generate simple MicroBlaze processors with all peripherals. The DDR2 memory is managed by the Multi-Port Memory Controller (MPMC). In this project, four ports are dedicated for application access. The HDMI video input and output use two separate VFBC interface ports. The Xilinx Cache Link (XCL) interface and the Processor Local Bus (PLB) interface are designed for the other two interface ports to the MicroBlaze processors as shown in Figure 29. Figure 30 show the schematic diagram of the system. Figure 31 shows the completed microprocessor subsystem. An overall block diagram for the video compression hardware to implement the adaptive frame rate control strategy proposed in Chapter 4 is shown in Figure 32.



(a)



(b)

Figure 28. ISE software configuration for the Atlys™ development board. (a)The ISE software configuration. (b) Adding new microprocessor sub-system.

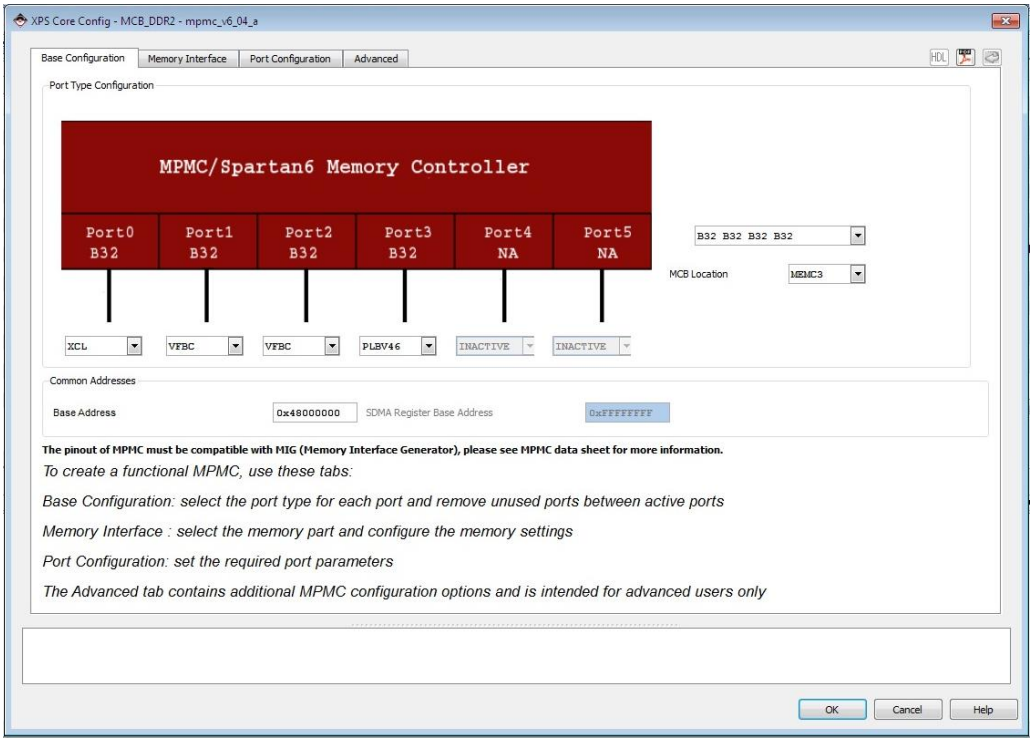


Figure 29. The MPMC interface for DDR2 memory.

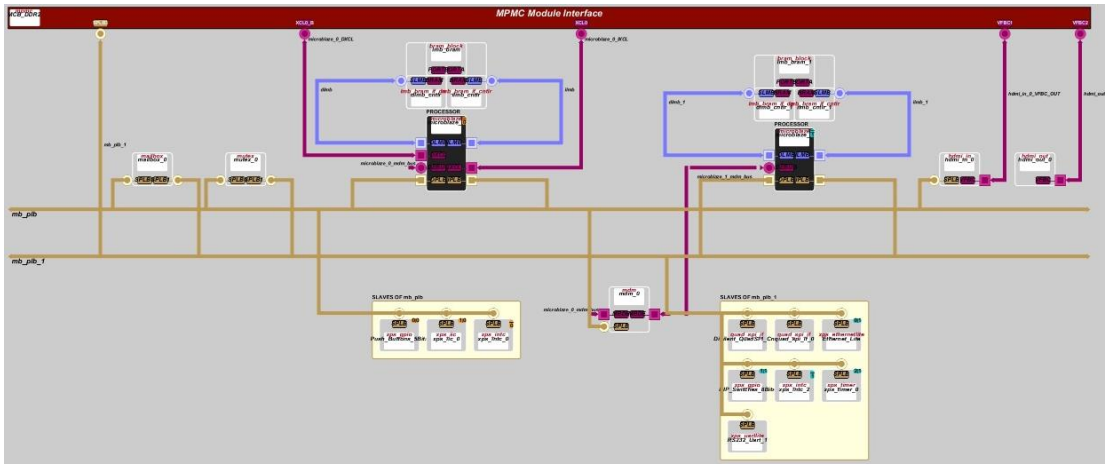
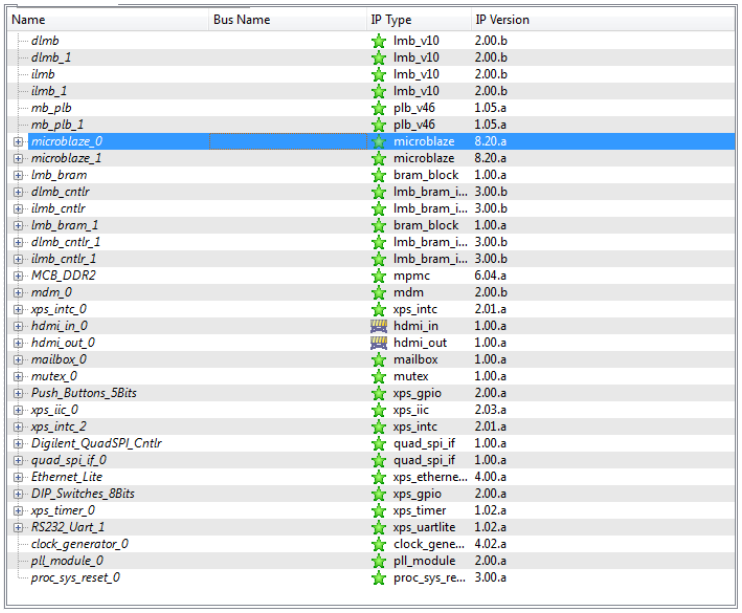
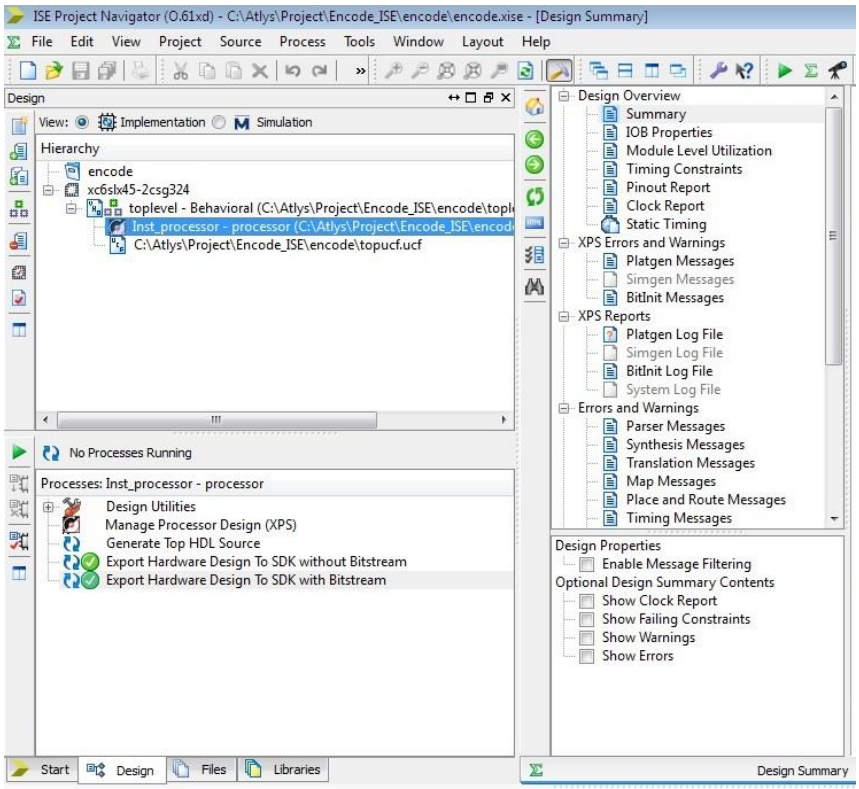


Figure 30. The schematic diagram of the completed microprocessor subsystem.



(a)



(b)

Figure 31. The completed design for the MicroBlaze subsystem. (a) The dual processors with peripherals generation by the EDK. (b) The microprocessor subsystem under the ISE.

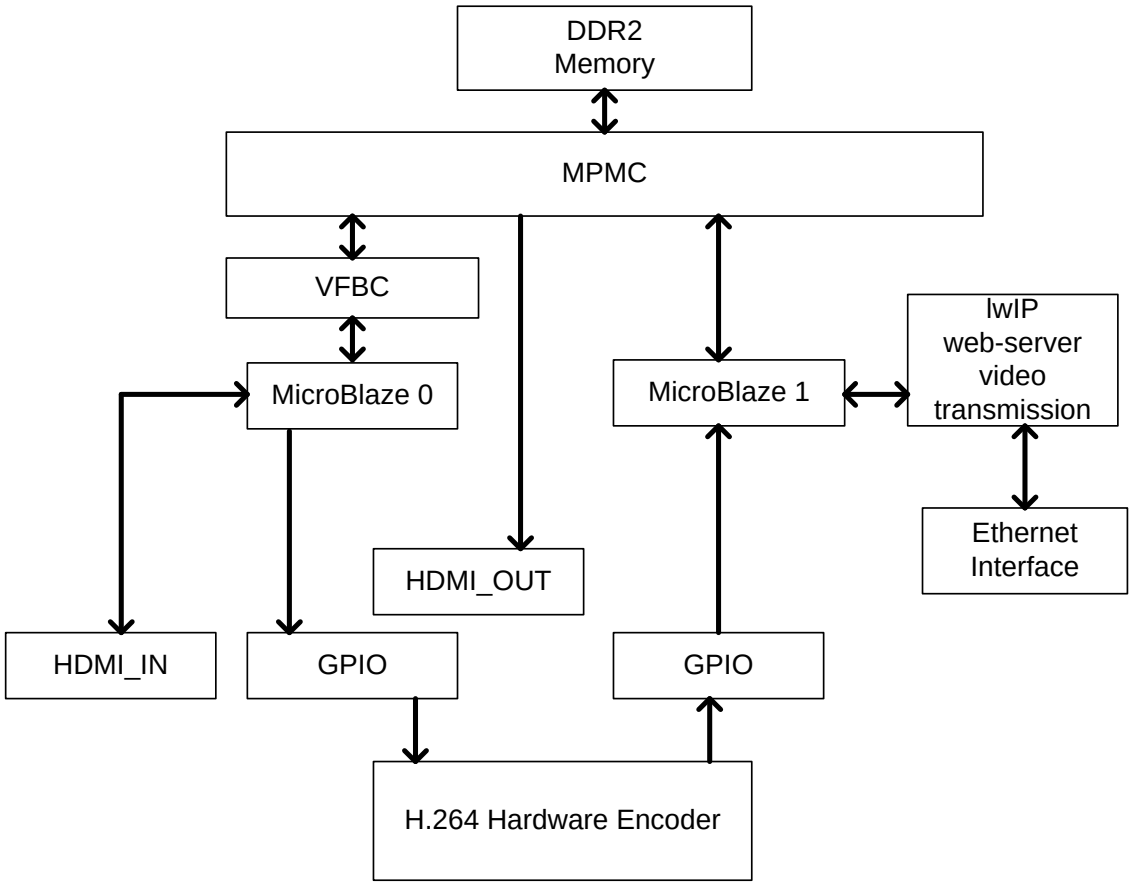


Figure 32. Block diagram of the video encoder implemented using an FPGA.

5.1.5 Results and Summary for the FPGA Video Encoder Implementation

The Atlys™ development board can design as a basic adaptive video rate controller. The board is cheaper than other FPGA development boards but has several limitations such as memory size, operation frequency, video interface, etc. The encoded video output is very sluggish and creates a lot of latency. Main video input and output interfaces for the development board are HDMI which support high resolution video format. The high resolution video means more data to process and an increased processor operating time and resource requirement. The free lwIP TCP/IP network stack cannot enable the IGMP so that the multicast scheme is impossible to implement

without using a different TCP/IP network stack implementation. However, the VHDL and C language source codes are able to be recompiled and be reused on another FPGA development board. A different analog video input interface with higher FPGA speed and more peripheral resources would be a better and more practical way to implement a video CODEC with the proposed rate control strategy.

5.2 Embedded Linux Hardware Development

Many Linux distributions have been developed. Here, I have designed and tested the video encoder with a small Linux distribution such as the Lubuntu version 13.04, the SilTaz version 4.0, etc. These tiny versions of Linux were installed on a Pentium dual core desktop computer for testing.

5.2.1 Analog Video Input and the Hardware Encoder

The H.264/AVC hardware encoder is the main designed encoder for this project. The SmartCapture Pro USB encoder by FastVDO, Inc., was selected because of its low cost and because it provides a native Linux driver. The video transmission framework is based on the open-source VLC multimedia player [68].

VLC can be configured to stream video either graphically by using the Graphical User Interface (GUI) or by using the command line. However, the command line is more advantageous than the GUI because the command line can be compiled to a script file which is able to be run as an application file. A list of the commands to configure VLC using command line is shown in Figure 33.

- Multicast with the RTP transmission:

  vlc -vvv v4l2:///dev/video0:input=1:standard=NTSC:no-audio:live-caching=3 --sout '#transcode{vcodec=mp2v,vb=800,scale=1}:rtp{mux=ts,dst= 224.168.0.222,port=8554,sdp=sap://,name="v"}'

- Unicast with the RTSP transmission:

  vlc -vvv v4l2:///dev/video0:input=1:standard=NTSC:no-audio:live-caching=3 --sout '#transcode{vcodec=mp2v,vb=800,scale=1}:rtp{sdp=rtsp://:8554/v}'

- Multiple data transmissions (unicast and multicast) for the same content:

  vlc -vvv v4l2:///dev/video0:input=1:standard=NTSC:no-audio:live-caching=3 --'sout=#transcode{vcodec=mp2v,vb=800,scale=1,}:duplicate{dst=rtp{sdp= rtsp://:8554/video},dst=rtp{dst=224.168.0.222,port=5004,mux=ts,ttl=1}}'

Figure 33. Show a list of the commands to configure VLC.

A set of daemon programs written in the C language are a main development for this project. A daemon process in Linux is similar to a “service” in Microsoft Windows, in the sense that both Linux daemons and Microsoft Windows services run behind the scenes to work on specific applications. In this case, the daemon process runs VLC and sends the multicast traffic to the multicast address. The daemon also waits for unicast connection requests to be issued by clients in case of multicast failure. The ping test process can be implemented with another daemon process. Like task scheduler in Microsoft Windows, the software utility Cron is a schedule access command that run a specific application in every interval time. Cron can be used to run the ping test process. The ping results are used to select the video transmission profile based on the setting

threshold levels. For example, decoded video with dropped packet network that is shown in Figure 22 should select the input video frame rate at 5 fps. The threshold levels are setup based on the network errors and percent of traffic congestion.

5.5.2 Results and Summary for the Linux-Based Video Encoder

The Linux-based video encoder works better than the FPGA implementation. The video latency can be adjusted by changing the cache memory size in VLC. The camera control is using the converter box to convert the TCP/IP command to the serial RS-232 to talk to the camera along highways that create some delay. From the experiment, caching memory at 300 milliseconds is adequate to move the camera without noticing the delay because it is compensated to the camera control delay. The video encoder based on the embedded Linux is also simple to deploy. Many different hardware platforms are cheap and developed to support Linux, such as the Raspberry Pi, the Beagle board, the Arm board, etc. [86-88]. The rugged embedded Linux based board is also available on the market. All developed software codes are able to be recompiled and reused for any development board.

5.3 Summary

Both the FPGA and the embedded Linux platforms are able to be designed for the proposed video encoder method. Some considerations to select the hardware platform are included but limit to the video CODEC license, number of productions, and cost. For example, MPEGLA, which owns the H.264/AVC patent, allows a manufacturer to produce video encoders without payment of a royalty fee up to 100,000

units per year as shown in Appendix C. The overall cost per unit for the FPGA is higher than the embedded Linux implementation at a small number of manufactured units because the FPGA solution needs to pay for the Printed Circuit Board (PCB) design, the assembly for the electronic components, the cost to program the FPGA, etc. In case of high demand, the FPGA solution is preferable due to the cheap cost per unit. The Application-Specific Integrated Circuit (ASIC), a specific integrated circuit (IC) that is designed for a customized application, is chosen to be used instead of the FPGA chip. The ASIC is cheaper than the FPGA but it cannot be reprogramed. If a high number of units are produced, then the royalty fee will be added to the overall cost.

The original plan for this dissertation is to make a contribution to the Oklahoma statewide ITS. This is a low demand of production. The embedded Linux system is the best choice for the proposed video rate control implementation.

Chapter 6: Conclusion and Recommendations for Future Research

6.1 Conclusion

In intelligent transportation systems everywhere, including the Oklahoma statewide ITS, the images from roadway surveillance video cameras are used for important applications including vehicle sensing, acquisition of real-time weather information, traffic congestion monitoring, and deployment of smart work zones. The accuracy of all these measurements depends on maintaining a high image quality in the spatial dimensions, but not having a high video frame rate. Both image sensor applications based using arithmetic image processing algorithms and those where the video is viewed and processed by humans require that video images of high quality be delivered to the destination nodes in the video network.

Block based video compression algorithms usually manage the video quality by using adaptive rate control where the values of the QP parameter changes based on video buffer utilization and measured network congestion. The available adaptive rate control algorithms range from simple frame skipping to the sophisticated and complicated tradeoffs between bandwidth, frame rate, and spatial image fidelity.

In this dissertation, I introduced an innovated video frame rate control strategy by using simple ping test results. The experimental results from Section 4.1 show that the network environment status, noise, error, and media can be identified by using the ping test results. The reduction of the input video frame rate can improve the video quality output in most network congestion scenarios as in the results presented in Section 4.2. By combining the ping test results as a video frame rate control and

adaptive reducing the input video frame rate, I demonstrated improvements in the overall video quality in both the temporal and spatial dimensions.

In Chapter 5, I presented practical hardware designs for implementing the proposed rate control strategy. The volume of units that need to be produced is the significant parameter that should be considered in selecting a particular hardware platform to use for implementing the proposed video compression algorithm. MPEGLA, which owns the H.264/AVC patent, grants licenses without paying a royalty fee for up to 100,000 units per year. The production cost per unit for the FPGA goes up at small manufacture number. In cases of small demand, the embedded Linux implementation is the best option.

6.2 Contribution

In this dissertation, I presented an innovative video frame rate control strategy for reducing the video frame rate based on results from simple ping tests that were used to measure network packet delay and estimate congestion. My proposed approach does not require changing or modifying any video compression standard. It is fully compatible with existing hardware deployed in the field and is free from any patent infringement issues. The control part can be done by changing the input video frame rate. Multiple video streams sent from the same video server simultaneously are defined in appendix G of the H.264/MPEG-4 AVC part 10 known as scalable video coding (SVC). However, it is not supported by any video encoder for ITS application that is available on the current market. This function needs to be implemented on both the

video server and clients. The main contributions of this dissertation include the following.

- I provided a practical solution to achieve video delivery in a video distribution network with incompatible interfaces and to optimize the data size by either sending redundant video streams in parallel using unicast and multicast schemes or by implementing the adaptive rate control scheme. To reduce the data traffic on the network, the user will first request multicast video. Unicast will be used when multicast streaming video cannot be connected. More complicated handshaking techniques will apply for the interactive traffic control.
- I presented an adaptive H.264 video encoder based on the simple ping package results. Due to the huge difference of ITS network characteristics, the network media between the video server and client are easy to identify by checking the latency, data loss, reordering, data corruption, delay, and jitter. The adaptive H.264 video encoder will dynamically adjust the video frame rate depending on the bandwidth capabilities of the links between the source and destination nodes and the real-time network traffic.
- I developed FPGA and embedded Linux implementations of the new adaptive H.264 rate control scheme and conducted experiments to characterized and compare their performance.

6.3 Recommendations for Future Research

In general, many researchers have offered solutions for frame layer rate control algorithms by changing the QP value. The ping test results should be able to be used to change the QP value as well. Experiments need to be performed to confirm the assumption.

Other more powerful FPGA development boards with better analog video interfaces than the one considered here need to be tested by recompiling the code and evaluating the operation speed and performance. A rugged Linux board is required to replace the Pentium dual core desktop machine. A hardened environment machine is preferred because the ITS traffic cabinets are located along the highway where environmental conditions can be extreme.

References

[1] U.S. Department of Transportation. (2013, August) A Summary of the Intermodal Surface Transportation Efficiency ACT of 1991 (ISETA). Visited August, 2013. [Online]. Avaliable: http://ntl.bts.gov/DOCS/ste.html

[2] U.S. Department of Transportation. (2013, August) ITS Benefits: Continuing Successes and Operational Test Results. Visited August, 2013. [Online]. Avaliable: https://www.fhwa.dot.gov/publications/research/operations/its/98002/contsuccess.pdf

[3] U.S. Department of Transportation. (2013, August) The Transportation Equity ACT for the 21st Century (TEA-21). Visited August, 2013. [Online]. Avaliable: https://www.fhwa.dot.gov/tea21/sumcov.htm

[4] U.S. Department of Transportation. (2013, August) Safe, Accountable, Flexible, Efficient Transportation, Equity ACT: A Legacy for Users (SAFETEA-LU). Visited August, 2013. [Online]. Avaliable: https://www.fhwa.dot.gov/safetealu/index.htm

[5] National Transportation Communication for ITS Protocol, "Object definitions for Electrical and Lighting Management Systems (ELMS)," AASHTO/ITE/NEMA, Washington, D.C., A rcommended standard of the joint committee on the NTCIP 1213 version v02.19, 2006.

[6] National Transportation Communication for ITS Protocol, "Object definitions for Closed Circuit Television (CCTV) Camera Control," AASHTO/ITE/NEMA, Washington, D.C., A recommended standard of the joint committee on the NTCIP 1205 version v01.08, 2004.

[7] National Transportation Communication for ITS Protocol, "Object definitions for Dynamic Message Signs (DMS)," AASHTO/ITE/NEMA, Washington, D.C., A recommended standard of the joint committee on the NTCIP 1203 version v03, 2011.

[8] National Transportation Communication for ITS Protocol, "Object definitions for Actuated Traffic Signal Controller (ASC) Units," AASHTO/ITE/NEMA, Washington, D.C., A recommended standard of the joint committee on the NTCIP 1202 version v01.07, 2005.

[9] National Transportation Communication for ITS Protocol, "The NTCIP guide," AASHTO/ITE/NEMA, Washington, D.C., A recommended standard of the joint committee on the NTCIP 9001 version v04, 2009.

[10] S. Kanumuri, S.G. Subramanian, P.C. Cosman, and A.R. Reibman, "Predicting H.264 packet loss visibility using a generalized linear model," in *IEEE Int. Conf. on Image Process.*, Atlanta, GA, Oct. 2006, pp. 2245 - 2248.

[11] Ting-Lan Lin and P.C. Cosman, "Network-based packet loss visibility model for SDTV and HDTV for H.264 videos," in *IEEE Int. Conf. on Acoust. Speech and Signal Process.*, Dallas, TX, Mar. 2010, pp. 906 - 909.

[12] Ting-Lan Lin, S. Kanumuri, Y. Zhi, D. Poole, P.C. Cosman and A.R. Reibman, "A versatile model for packet loss visibility and its application to packet prioritization," in *IEEE Trans. Image Process.*, vol. 19, no. 3, pp. 722-735, Mar. 2010.

[13] Ting-Lan Lin, Y.L. Chang, and P.C. Cosman, "Subjective experiment and modeling of whole frame packet loss visibility for H.264," *presented at the Int. Workshop on Packet Video*, Dec. 2010.

[14] L. Jinhui, and Z. Meng, "A new vehicle detection algorithm for real time image processing system," in *IEEE Int. Conf. on Comput. Appl. and Syst. Modeling*, Taiyuan, OCT. 2010, pp. v10-1 – v10-4.

[15] H.S. Moon, Y.V. Alkihimov, Y.D. Kim, "Design of a cost effective real time vehicle detector system architecture," in *IEEE Int. Sci. and Practical Conf. on Modern Technique and Technol.*, Tomsk, Mar. 2005, pp. 49-51.

[16] R.K. Gilbert, and Q. Holmes, "Dynamic traffic information from remote video monitors," in *IEEE Int. Conf. on Veh. Navigation and Inform. Syst.*, Dearborn, Michigan, Oct. 1991, pp. 213- 243.

[17] N.C. Mithun, N.U. Rashid, and S.M.M. Rahman, "Detection and classification of vehicles from video using multiple-time spatial images," in *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1215-1225, Sept. 2012.

[18] J. Patrik, "Classification of road conditions from camera images and weather data," in *IEEE Int. Conf. on Computational Intell. for Measurement Syst.*, Ottawa, Canada, Sept. 2011, pp. 1-6.

[19] J.S. Hong, L. Xin, C. Yang-Zhou, and G. Yuan-Yuan, "Weather identifying system based on vehicle video image," in *IEEE World Congr. On Intell. Control and Automation*, Taipei, Jun. 2011, pp. 172-175.

[20] A. Sharma, D. Bullock, S. Peeta, and J. Krogmeier, "Detection of inclement weather conditions at a signalized intersection using a video image processing algorithm", *presented at the Signal Process. Workshop on Digital Signal Process.*, Sept. 2006.

[21] The National Institute for Occupational Safety and Health. (2013, August) Highway Work Zone Safety. Visited August, 2013. [Online]. Avaliable: http://www.cdc.gov/niosh/topics/highwayworkzones/

[22] M. Cady, "Portable traffic management system Smart work zone application operational test evaluation report," SRF consulting group Inc., SRP No. 0942089.7/11, May 1997.

[23] A. Kamyab, T.H. Maze, M.Nelson, and S.D. Schrock, "Using simulation to evaluate the performance of smart work zone technologies and other strategies to reduce congestion," in *ITS World Congr.*, Toronto, Canada, Sept. 1999.

[24] M. Thomas, and K. Ali, "Work zone simulation model," Central for transportation research and education, Iowa state university, report for traffic management strategies for merge areas in rural interstate work zones, CTRE management project 97-12 1999.

[25] J.M. Sullivan, C.B. Winkler, and M.R. Hagan, "Work zone safety ITS smart barrel for an adaptive queue-warning system," Transportation research institute, The university of Michigan, UMTRI-2005-3, Feb 2005.

[26] Z.G. Li, F. Pan, K.P. Lim, X. Lin, and S. Rahardja, " Adaptive rate control for H.264," in *IEEE Int. Conf. on Inage Process.*, May 2004, pp. 745-748.

[27] L. Hung-Ju, C. Tihao, and Z. Ya-Qin, "Scalable rate control for MPEG-4 video," in *IEEE Trans. Circuits. Syst. Video Technol.*, vol. 10, no. 6, pp. 878-894, Sept. 2000.

[28] Z. G. Li, F. Pan, K. P. Lim, G. N. Feng, X. Lin, S. Rahardja, and D. J. Wu, "Adaptive rate control for H.264," in *IEEE Int. Conf. on Multimedia and Expo*, May 2003, pp. 581-584.

[29] W. Li-Feng, X. Chen, N. Jian-Wei, and L. Er-Hong, "An optimized rate control algorithm for H.264 under low delay constraint," in *IEEE Int. Conf. on Wireless Commu. Network and Mobile Commu.*, Wuhan, Sept. 2006, pp. 1-4.

[30] Y. Jie, Z. Qian, and Z Lei, "The study of frame complexity prediction and rate control in H.264 encoder," in *IEEE Int. Conf. on Inage Anal. and Signal Process.*, Taizhou, Apr. 2009, pp. 187-191.

[31] C. Xiao, and L. Feifei, "A reformative frame layer rate control algorithm for H.264," in *IEEE Trans. Consum. Electron.*, vol. 56, no. 4, pp. 2806-2811, Nov. 2010.

[32] J. Minqiang, Y. Xiaoquan, and L. Nam, "Improved frame-layer rate control for H.264 using MAD ratio," in *IEEE Int. Conf. on Circuit and Syst.*, May 2004, pp. 813-816.

[33] X. Jianfeng, and H. Yun, "A novel rate control for H.264," in *IEEE Int. Conf. Circuit and Syst.*, May 2004, pp. 809-812.

[34] F. Pan, X. Lin, S. Rahardja, K. P. Lim, Z. G. Li, D. J. Wu, and S. Wu, "Proactive frame-skipping decision scheme for variable frame rate video coding," in *IEEE Int. Conf. Mutimedia and Expo*, Jun. 2004, pp. 1903-1906.

[35] P. Usach, J. Sastra, and J. M. Lopez, "Variable frame rate and GOP size H.264 rate control for mobile communication," in *IEEE Int. Conf. Mutimedia and Expo*,Jun 2009, pp. 1772-1775.

[36] Y. Chia-Hung, F. J. Shu-Jhen, L. Chih-Yang, and C. Mei-Juan, "Temporal video transcoding based on frame complexity analysis for mobile video communication," in *IEEE Trans. Broadcast.*, vol. 59, no. 1, pp. 38-46, Mar. 2013.

[37] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," in *IEEE Trans. Image Process..*, vol. 13, no. 4, pp. 600-612, Apr. 2004.

[38] C. Ziguan, and Z. Xiuchang, "SSIM-based content adaptive frame skipping for low bit rate H.264 video coding," in *IEEE Int. Conf. on Commun. Technol.*, Nov. 2010, pp. 484-487.

[39] O. Tao-Sheng, H. Yi-Hsin, and C.H. Homer, "SSIM-based perceptual rate control for video coding," in *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 5, pp. 682-691, May 2001.

[40] Z. Cui, Z. Gan, and X. Zhu, "Structural similarity optimal MB layer rate control for H.264," in *IEEE Int. Conf. Wiress Commu. and Signal Process.*, Nanjing, Nov. 2011, pp. 1-5.

[41] Live image quality assessment database release 2. Visited September, 2013. [Online]. Avaliable: http://live.ece.utexas.edu/research/quality

[42] N. Ponomarenko, M. Carli, V. Lukin, K. Egiazarian, J. Astola, and F. Battisti, “Color image database for evaluation of image quality metrics,” in *IEEE Int. Conf. Multimedia Signal Process.*, Cairns, AU, Oct. 2008, pp. 403-408.

[43] L. Xu, S. Li, K.N. Ngan, and L. Ma, “Consistent visual quality control in video coding," in *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 6, pp. 975-989, Jun. 2013.

[44] L. Xu, K.N. Ngan, S.N. Li, and L. Ma, “Video quality metric for consistent visual quality control in video coding,” in *IEEE Asia-Pacific Signal and Inform. Process. Conf.*, California, Dec. 2012, pp. 1-7.

[45] R. Feghali, F. Speranza, D. Wang, and A. Vincent, “Video quality metric for bit rate control via joint adjustment of quantization and frame rate,” in *IEEE Trans. Broadcast.*, vol. 53, no. 1, pp. 441-446, Mar. 2007.

[46] S. Deering, “Multicast routing in a datagram network,” *Ph.D. dissertation, Dept. Comput. Sci. Stanford University*, Stanford, CA, 1991.

[47] Extreme Networks, Inc., “ExtremeXOS accelerated extreme network specialist certification technical training”. Rev. 12.1, Vol. 2 of 2.

[48] D. Minoli, “IP multicast with applications to IPTV and mobile DVB-H,", A John Wiley & Sons, Inc., 2008. Stanford, CA, 1991.

[49] M. Goncalves, and K. Niles, “IP multicasting concepts and applications”, in *Networking series*, Texas, McGraw-Hill, 1999.

[50] MPEG LA's AVC license will continue not to charge royalties for internet video that is free to end users. Visited September, 2013. [Online]. Avaliable: http://www.mpegla.com/Lists/MPEG%20LA%20News%20List/Attachments/226/n-10-02-02.pdf

[51] VP8. Visited September, 2013. [Online]. Avaliable: http://en.wikipedia.org/wiki/VP8

[52] DoJ looking into possible anti-WebM moves by MPEGLA. Visited October, 2013. [Online]. Avaliable: http://arstechnica.com/tech-policy/2011/03/report-doj-looking-into-possible-anti-webm-moves-by-mpeg-la/

[53] WebM patent fight ahead for Google?. Visited October, 2013. [Online]. Avaliable: http://www.streamingmedia.com/Articles/News/Featured-News/WebM-Patent-Fight-Ahead-for-Google-76781.aspx

[54] Google and MPEG LA announce covering VP8 video format. Visited October, 2013. [Online]. Avaliable: http://www.mpegla.com/Lists/MPEG%20LA%20News%20List/Attachments/88/n-13-03-07.pdf

[55] VP9 is almost here, but a Nokia patent fight might have it DOA. Visited October, 2013. [Online]. Avaliable: http://www.streamingmedia.com/Articles/Editorial/Featured-Articles/VP9-Is-Almost-Here-But-a-Nokia-Patent-Fight-Might-Have-it-DOA-89603.aspx

[56] J. Bankoski, P. Wilkins, and X. Yaowu, "Technical overview of VP8, an open source video codec for the web," in *IEEE Int. Conf. Multimedia and Expo,* Barcelona, Jul. 2011, pp. 1-6.

[57] C. Feller, J. Wuenschmann, T. Roll, and A. Rothermel, "The VP8 video codec-overview and comparison to H.264/AVC," in *IEEE Int. Conf. Consum. Electro..*, Berlin, Sept. 2011, pp. 1-6.

[58] P.K. Bansal, V. Bansal, M.N. Shukla, and A.S. Motra, "VP8 encoder-cost effective implementation," in *IEEE Int. Conf. on Software, Commun. and Comput. Network*, Sept. 2012, pp. 1-6.

[59] Y. Yohaan, K. Myungchul, L. Sooyong, B. Lee, J.H. Soon, and L. Kyunghee, "Performance analysis of H.264/AVC, H.264/SVC, and VP8 over IEEE 802.11 wireless network," in *IEEE Int. Conf. Comput. and Commu.,* Cappadocia, Jul. 2012, pp. 151-156.

[60] Y.O. Sharrab, and N.J. Sarhan, "Detailed comparative analysis of VP8 and H.264," in *IEEE Int. Conf. Multimedia.,* Irvine, CA, Dec. 2012, pp. 133-140.

[61] Overview of VP-next a next generation open video codec. Visited October, 2013. [Online]. Avaliable: http://www.ietf.org/proceedings/85/slides/slides-85-videocodec-4.pdf

[62] VP-next overview and progress update. Visited October, 2013. [Online]. Avaliable: http://downloads.webmproject.org/ngov2012/pdf/04-ngov-project-update.pdf

[63] E. Vorakitolan, J. P. Havlicek, R. D. Barnes, and A. R. Stevenson, "Simple, effective rate control for video distribution in heterogenous intelligent transportation system applications," in *Proc. IEEE Southwest. Symp. Image Anal. & Interp.*, San Diego, CA, Apr. 2014, pp. 37-40.

[64] Tutorial: the H.264 Scalable Video Codec (SVC). Visited October, 2013. [Online]. Avaliable: http://www.eetimes.com/document.asp?doc_id=1275532

[65] Extreme Networks, Inc., "Summit 200 series switch installation and user guide," California, Software version 6.2e.2, Jun. 2003.

[66] WinTV-HVR-950Q Hybrid TV stick. Visited December, 2013. [Online]. Avaliable: http://www.hauppauge.com/site/products/data_hvr950q.html/

[67] A network protocol analyzer. Visited December, 2013. [Online]. http://www.wireshark.org/

[68] VLC media player. Visited January, 2014. [Online]. Avaliable: http://www.videolan.org/vlc/index.html

[69] Network Working Group, "Real Time Streaming Protocol (RTSP) and Real Time Transport Protocol (RTP)," California, Request for comments: 2326, Apr. 1998.

[70] The The Wide Area Network Simulator (WANem). Visited January, 2014. [Online]. Avaliable: http://wanem.sourceforge.net/

[71] SmartCapture PRO: USB based H.264/AAC encoder device. Visited January, 2014. [Online]. Avaliable: http://www.fastvdo.com/smartcapture/

[72] Ping (networking utility). Visited January, 2014.[Online]. Avaliable:http://en.wikipedia.org/wiki/Ping_(networking_utility)

[73] Aimersoft video editor. Visited January, 2014.[Online]. Avaliable: http://www.aimersoft.com/video-studio-express.html

[74] MSU video quality measurement tool. Visited January, 2014.[Online]. Avaliable:http://compression.ru/video/quality_measure/video_measurement_tool_en.html

[75] Embedded Linux wiki. Visited February, 2014.[Online]. Avaliable:http://elinux.org/Main_Page

[76] H.264 hardware encoder in VHDL. Visited February, 2014. [Online]. Avaliable:http://opencores.org/websvn,filedetails?repname=video_systems&path=%2Fvideo_systems%2Ftrunk%2Fdoc%2FH264-encoder-manual.pdf

[77] lwIP – a lightweight TCP/IP stack. Visited February, 2014. [Online]. Avaliable: http://savannah.nongnu.org/projects/lwip/

[78] Atlys HDMI demonstration project. Visited February, 2014. [Online]. Avaliable:http://www.digilentinc.com

[79] Extended display identification data. Visited February, 2014. [Online]. Avaliable: http://en.m.wikipedia.org/wiki/Extended_display_identification_data

[80] VESA enhanced extended display identification data standard. Visited February, 2014. [Online]. Avaliable: http://vesa.org

[81] Implementing a TMDS video interface in the Spartan-6 FPGA. Visited February, 2014. [Online]. Avaliable: http://www.xilinx.com/support/documentation/application_notes/xapp495_S6TMDS_Video_Interface.pdf

[82] Integrating a Video Frame Buffer Controller (VFBC) in system generator. Visited March, 2014. [Online]. Avaliable: http://www.xilinx.com/support/documentation/application_notes/xapp1136.pdf

[83] Understanding luminance and chrominance. Visited March, 2014. [Online]. Avaliable: http://wolfcrow.com/blog/understanding-luminance-and-chrominance/

[84] Xilinx processor IP library software drivers. Visited March, 2014. [Online]. http://www.cs.indiana.edu/hmg/le/project-home/xilinx/ise_13.2/ISE_DS/EDK/sw/XilinxProcessorIPLib/drivers/gpio_v3_00_a/doc/html/api/xgpio_8h.html

[85] Spartan-6 family. Visited March, 2014. [Online]. http://www.xilinx.com/products/silicon-devices/fpga/spartan-6/

[86] Raspberry Pi development board. Visited March, 2014. [Online]. http://www.raspberrypi.org/

[87] BeagleBoard development board. Visited March, 2014. [Online]. http://beagleboard.org/

[88] ARM development boards. Visited March, 2014. [Online]. http://arm.com/products/tools/development-boards/

Appendix A: The Ping Test Result

- The 50 percent data corruption

```
ekasit@ekasit-desktop:~$ ping -c 100 -s 1000 192.168.0.9
PING 192.168.0.9 (192.168.0.9) 1000(1028) bytes of data.
1008 bytes from 192.168.0.9: icmp_req=3 ttl=128 time=0.774 ms
1008 bytes from 192.168.0.9: icmp_req=10 ttl=128 time=0.846 ms
1008 bytes from 192.168.0.9: icmp_req=11 ttl=128 time=0.788 ms
1008 bytes from 192.168.0.9: icmp_req=12 ttl=128 time=0.562 ms
                                    .
                                    .
                                    .
1008 bytes from 192.168.0.9: icmp_req=83 ttl=128 time=0.497 ms
1008 bytes from 192.168.0.9: icmp_req=89 ttl=128 time=0.568 ms
1008 bytes from 192.168.0.9: icmp_req=96 ttl=128 time=0.841 ms
1008 bytes from 192.168.0.9: icmp_req=97 ttl=128 time=0.858 ms
1008 bytes from 192.168.0.9: icmp_req=99 ttl=128 time=0.772 ms
1008 bytes from 192.168.0.9: icmp_req=100 ttl=128 time=0.560 ms
--- 192.168.0.9 ping statistics ---
100 packets transmitted, 43 received, 57% packet loss, time 99060ms
rtt min/avg/max/mdev = 0.497/0.773/0.858/0.125 ms
```

- The 50 percent data duplication

```
ekasit@ekasit-desktop:~$ ping -c 100 -s 1000 192.168.0.9

PING 192.168.0.9 (192.168.0.9) 1000(1028) bytes of data.

1008 bytes from 192.168.0.9: icmp_req=1 ttl=128 time=0.630 ms

1008 bytes from 192.168.0.9: icmp_req=1 ttl=128 time=0.945 ms (DUP!)

1008 bytes from 192.168.0.9: icmp_req=2 ttl=128 time=0.846 ms

1008 bytes from 192.168.0.9: icmp_req=3 ttl=128 time=0.786 ms

                                    .

                                    .

                                    .

1008 bytes from 192.168.0.9: icmp_req=94 ttl=128 time=0.895 ms (DUP!)

1008 bytes from 192.168.0.9: icmp_req=95 ttl=128 time=0.842 ms

1008 bytes from 192.168.0.9: icmp_req=95 ttl=128 time=0.849 ms (DUP!)

1008 bytes from 192.168.0.9: icmp_req=96 ttl=128 time=0.843 ms

1008 bytes from 192.168.0.9: icmp_req=96 ttl=128 time=0.848 ms (DUP!)

1008 bytes from 192.168.0.9: icmp_req=97 ttl=128 time=0.844 ms

1008 bytes from 192.168.0.9: icmp_req=98 ttl=128 time=0.771 ms

1008 bytes from 192.168.0.9: icmp_req=99 ttl=128 time=0.842 ms

1008 bytes from 192.168.0.9: icmp_req=100 ttl=128 time=0.562 ms

--- 192.168.0.9 ping statistics ---

100 packets transmitted, 100 received, +46 duplicates, 0% packet loss, time 98998ms

rtt min/avg/max/mdev = 0.488/0.803/0.946/0.119 ms
```

❖ The 1000 millisecond data delay

```
ekasit@ekasit-desktop:~$ ping -c 100 -s 1000 192.168.0.9

PING 192.168.0.9 (192.168.0.9) 1000(1028) bytes of data.

1008 bytes from 192.168.0.9: icmp_req=1 ttl=128 time=1000 ms

1008 bytes from 192.168.0.9: icmp_req=2 ttl=128 time=1000 ms

1008 bytes from 192.168.0.9: icmp_req=3 ttl=128 time=1000 ms

1008 bytes from 192.168.0.9: icmp_req=4 ttl=128 time=1000 ms

                                    .

                                    .

                                    .

1008 bytes from 192.168.0.9: icmp_req=92 ttl=128 time=1000 ms

1008 bytes from 192.168.0.9: icmp_req=93 ttl=128 time=1000 ms

1008 bytes from 192.168.0.9: icmp_req=94 ttl=128 time=1000 ms

1008 bytes from 192.168.0.9: icmp_req=95 ttl=128 time=1000 ms

1008 bytes from 192.168.0.9: icmp_req=96 ttl=128 time=1000 ms

1008 bytes from 192.168.0.9: icmp_req=97 ttl=128 time=1000 ms

1008 bytes from 192.168.0.9: icmp_req=98 ttl=128 time=1000 ms

1008 bytes from 192.168.0.9: icmp_req=99 ttl=128 time=1000 ms

1008 bytes from 192.168.0.9: icmp_req=100 ttl=128 time=1000 ms

--- 192.168.0.9 ping statistics ---

100 packets transmitted, 100 received, 0% packet loss, time 99002ms

rtt min/avg/max/mdev = 1000.495/1000.755/1000.861/0.945 ms, pipe 2
```

- The 50 percent data packet loss

```
ekasit@ekasit-desktop:~$ ping -c 100 -s 1000 192.168.0.9
PING 192.168.0.9 (192.168.0.9) 1000(1028) bytes of data.
1008 bytes from 192.168.0.9: icmp_req=2 ttl=128 time=1.28 ms
1008 bytes from 192.168.0.9: icmp_req=3 ttl=128 time=0.855 ms
1008 bytes from 192.168.0.9: icmp_req=6 ttl=128 time=0.648 ms
1008 bytes from 192.168.0.9: icmp_req=8 ttl=128 time=0.642 ms
1008 bytes from 192.168.0.9: icmp_req=13 ttl=128 time=0.844 ms
                                    .
                                    .
                                    .
1008 bytes from 192.168.0.9: icmp_req=83 ttl=128 time=0.795 ms
1008 bytes from 192.168.0.9: icmp_req=87 ttl=128 time=0.771 ms
1008 bytes from 192.168.0.9: icmp_req=89 ttl=128 time=0.632 ms
1008 bytes from 192.168.0.9: icmp_req=90 ttl=128 time=0.844 ms
1008 bytes from 192.168.0.9: icmp_req=91 ttl=128 time=0.856 ms
1008 bytes from 192.168.0.9: icmp_req=92 ttl=128 time=0.840 ms
1008 bytes from 192.168.0.9: icmp_req=96 ttl=128 time=0.769 ms
1008 bytes from 192.168.0.9: icmp_req=99 ttl=128 time=0.668 ms
1008 bytes from 192.168.0.9: icmp_req=100 ttl=128 time=0.566 ms
--- 192.168.0.9 ping statistics ---
100 packets transmitted, 54 received, 46% packet loss, time 99010ms
rtt min/avg/max/mdev = 0.503/0.755/1.289/0.140 ms
```

- The data reorder

```
ekasit@ekasit-desktop:~$ ping -c 100 -s 1000 192.168.0.9
PING 192.168.0.9 (192.168.0.9) 1000(1028) bytes of data.
1008 bytes from 192.168.0.9: icmp_req=2 ttl=128 time=1000 ms
1008 bytes from 192.168.0.9: icmp_req=1 ttl=128 time=2000 ms
1008 bytes from 192.168.0.9: icmp_req=3 ttl=128 time=0.612 ms
1008 bytes from 192.168.0.9: icmp_req=4 ttl=128 time=0.852 ms
1008 bytes from 192.168.0.9: icmp_req=5 ttl=128 time=0.845 ms
                                    .
                                    .
                                    .
1008 bytes from 192.168.0.9: icmp_req=92 ttl=128 time=2000 ms
1008 bytes from 192.168.0.9: icmp_req=93 ttl=128 time=2000 ms
1008 bytes from 192.168.0.9: icmp_req=95 ttl=128 time=0.895 ms
1008 bytes from 192.168.0.9: icmp_req=98 ttl=128 time=0.553 ms
1008 bytes from 192.168.0.9: icmp_req=97 ttl=128 time=1000 ms
1008 bytes from 192.168.0.9: icmp_req=96 ttl=128 time=2001 ms
1008 bytes from 192.168.0.9: icmp_req=99 ttl=128 time=0.859 ms
1008 bytes from 192.168.0.9: icmp_req=100 ttl=128 time=0.560 ms
--- 192.168.0.9 ping statistics ---
100 packets transmitted, 100 received, 0% packet loss, time 99008ms
rtt min/avg/max/mdev = 0.523/980.738/2001.057/916.316 ms, pipe 3
```

❖ The jitter 500 ms with the delay time 300 ms

C:\Documents and Settings\Pathfinder>ping -n 100 -l 1000 192.168.0.9

Pinging 192.168.0.9 with 1000 bytes of data:

Reply from 192.168.0.9: bytes=1000 time=553ms TTL=128

Reply from 192.168.0.9: bytes=1000 time<1ms TTL=128

Reply from 192.168.0.9: bytes=1000 time=363ms TTL=128

Reply from 192.168.0.9: bytes=1000 time=118ms TTL=128

Reply from 192.168.0.9: bytes=1000 time=262ms TTL=128

Reply from 192.168.0.9: bytes=1000 time<1ms TTL=128

.

.

.

Reply from 192.168.0.9: bytes=1000 time=534ms TTL=128

Reply from 192.168.0.9: bytes=1000 time=340ms TTL=128

Reply from 192.168.0.9: bytes=1000 time=319ms TTL=128

Reply from 192.168.0.9: bytes=1000 time=59ms TTL=128

Reply from 192.168.0.9: bytes=1000 time<1ms TTL=128

Reply from 192.168.0.9: bytes=1000 time=374ms TTL=128

Reply from 192.168.0.9: bytes=1000 time=976ms TTL=128

Ping statistics for 192.168.0.9:

Packets: Sent = 100, Received = 100, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 1643ms, Average = 416ms

Appendix B: The Video Quality Measurement and Analysis Results

❖ The T-1 Network Simulation (The Data Corruption)

	5 percent			10 percent			30 percent			50 percent		
	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps
PSNR (AVG)	20.5519	19.7384	19.6287	19.6997	19.6213	19.1599	21.0342	19.7099	19.5861	20.1476	19.6845	19.2107
PSNR (MIN)	14.1829	14.5811	13.0351	10.788	12.9679	12.9762	8.7378	13.2547	13.1441	12.6272	12.2607	9.91355
PSNR (MAX)	31.7872	30.3255	36.4615	34.0086	34.8955	32.2104	34.0249	30.906	33.9948	34.551	30.1417	30.8043
PSNR (SD)	2.5316	2.81036	3.97271	3.76132	3.74578	3.15678	3.84002	2.44798	3.74811	3.63015	3.30458	3.32158
PSNR (95% Confidence Level)	0.039	0.0633	0.06121	0.05795	0.05771	0.04864	0.05916	0.03772	0.05775	0.05593	0.05469	0.05117
Lower limit	20.5129	19.6751	19.5675	19.6418	19.5636	19.1113	20.9751	19.6722	19.5284	20.0917	19.6298	19.1595
Upper limit	20.5909	19.8017	19.69	19.7577	19.679	19.2085	21.0934	19.7476	19.6439	20.2036	19.7392	19.2619
PSNR gain/loss	0	0.81347	0.92315	0	0.07842	0.53984	0	1.32432	1.44809	0	0.46312	0.93697
SSIM (AVG)	0.85282	0.82665	0.8353	0.83364	0.8313	0.81488	0.86243	0.82841	0.82943	0.84068	0.83937	0.80518
SSIM (MIN)	0.65037	0.64875	0.64892	0.5957	0.55308	0.61577	0.50978	0.5444	0.54535	0.61422	0.64003	0.54344
SSIM (MAX)	0.974	0.98344	0.93876	0.98083	0.97817	0.97455	0.97807	0.98046	0.97111	0.97869	0.96404	0.97122
SSIM (SD)	0.07182	0.08899	0.08215	0.08451	0.08697	0.09638	0.07249	0.08894	0.0781	0.07937	0.0831	0.10099
SSIM (95% Confidence Level)	0.00111	0.00137	0.00165	0.0013	0.00134	0.00148	0.00112	0.00137	0.0012	0.00122	0.00182	0.00156
Lower limit	0.85171	0.82528	0.83366	0.83234	0.82996	0.8134	0.86132	0.82704	0.82823	0.83946	0.83755	0.80362
Upper limit	0.85393	0.82802	0.83695	0.83494	0.83264	0.81637	0.86355	0.82978	0.83064	0.8419	0.84119	0.80673
SSIM gain/loss	0	0.02617	0.01752	0	0.00234	0.01876	0	0.03403	0.033	0	0.00131	0.0355
VQM (AVG)	7.19951	8.63897	7.89816	8.61567	8.64184	8.99887	7.54528	8.68815	8.53694	8.48587	8.52699	8.7743
VQM (MIN)	1.09567	1.03326	1.3701	2.32641	1.83216	1.4337	2.69194	2.11642	3.44084	2.27417	2.79776	3.13959
VQM (MAX)	12.3013	13.5486	12.7175	15.3241	13.6655	13.7995	16.7039	13.2942	13.994	14.1028	15.7903	16.3571
VQM (SD)	2.10574	2.53567	1.99442	2.38583	2.4676	2.47586	1.96905	2.53173	1.99844	2.2777	2.85546	2.5065
VQM (95% Confidence Level)	0.03087	0.03907	0.03073	0.03676	0.03802	0.03814	0.03034	0.03901	0.03079	0.03509	0.03318	0.03862
Lower limit	7.16864	8.59991	7.86743	8.57891	8.60382	8.96073	7.51494	8.64915	8.50615	8.45078	8.49381	8.73569
Upper limit	7.23039	8.67804	7.92889	8.65243	8.67986	9.03702	7.57562	8.72716	8.56773	8.52096	8.56017	8.81292
VQM gain/loss	0	1.43946	0.69865	0	0.02617	0.3832	0	1.14287	0.99166	0	0.04112	0.28843

❖ The T-1 Network Simulation (The Data Loss)												
	5 percent			**30 percent**			**30 percent**			**50 percent**		
	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps
PSNR (AVG)	19.77	20.2049	19.0202	20.1055	19.1816	19.8584	21.3993	19.5566	19.5317	20.1047	19.806	19.6925
PSNR (MIN)	10.9274	14.4879	10.7076	11.515	10.6125	13.7055	15.645	10.6753	13.8921	11.767	11.1321	14.322
PSNR (MAX)	34.8238	35.0361	31.0605	34.2523	34.6189	31.4414	35.0347	35.277	29.1831	34.3404	35.2692	31.1482
PSNR (SD)	3.88398	3.57029	3.40206	3.81541	4.19067	3.04071	3.34149	3.601	2.76706	3.55573	3.54181	2.78182
PSNR (95% Confidence Level)	0.05984	0.05501	0.05241	0.05878	0.06456	0.04685	0.05148	0.05548	0.04263	0.05478	0.05457	0.04286
Lower limit	19.7101	20.1499	18.9678	20.0467	19.117	19.8115	21.3478	19.5011	19.4891	20.05	19.7514	19.6496
Upper limit	19.8298	20.2599	19.0727	20.1643	19.2461	19.9052	21.4508	19.6121	19.5743	20.1595	19.8606	19.7353
PSNR gain/loss	0	-0.4349	0.74974	0	0.92396	0.24714	0	1.84269	1.86758	0	0.29875	0.41228
SSIM (AVG)	0.83649	0.84995	0.80441	0.85057	0.8167	0.82652	0.87782	0.82794	0.81933	0.84771	0.84014	0.81887
SSIM (MIN)	0.59747	0.66232	0.51703	0.6292	0.54503	0.61914	0.7004	0.562	0.63145	0.56216	0.62316	0.56893
SSIM (MAX)	0.97892	0.98127	0.97149	0.97839	0.98085	0.97313	0.97963	0.98159	0.96452	0.97836	0.98179	0.97231
SSIM (SD)	0.08254	0.06995	0.10268	0.07189	0.09717	0.08392	0.05045	0.08673	0.08826	0.07004	0.08322	0.094
SSIM (95% Confidence Level)	0.00127	0.00108	0.00158	0.00111	0.0015	0.00129	0.00078	0.00134	0.00136	0.00108	0.00128	0.00145
Lower limit	0.83522	0.84887	0.80283	0.84946	0.8152	0.82523	0.87704	0.8266	0.81797	0.84664	0.83885	0.81742
Upper limit	0.83776	0.85102	0.80599	0.85167	0.81819	0.82781	0.8786	0.82928	0.82069	0.84879	0.84142	0.82032
SSIM gain/loss	0	-0.0135	0.03208	0	0.03387	0.02405	0	0.04988	0.05849	0	0.00758	0.02885
VQM (AVG)	8.35227	8.19075	9.01826	7.95492	8.9718	8.50425	7.39705	8.79763	8.60154	8.04813	8.31246	8.47359
VQM (MIN)	1.09553	1.19329	1.47773	2.33393	1.79254	1.41505	1.08044	1.0565	2.9336	2.02379	1.33178	1.45729
VQM (MAX)	14.9827	12.9852	15.5899	14.4456	15.0744	13.4075	11.6529	14.9602	13.2214	14.1341	14.4128	13.7855
VQM (SD)	2.41431	2.22814	2.59656	2.05709	2.78508	2.3961	1.98191	2.48732	2.23468	1.9906	2.31657	2.19498
VQM (95% Confidence Level)	0.0372	0.03433	0.04	0.03169	0.04291	0.03692	0.03053	0.03832	0.03443	0.03067	0.03569	0.03382
Lower limit	8.31508	8.15642	8.97826	7.92323	8.92889	8.46734	7.36652	8.75931	8.56711	8.01746	8.27677	8.43978
Upper limit	8.38947	8.22508	9.05827	7.98661	9.01471	8.54117	7.42759	8.83595	8.63597	8.0788	8.34815	8.50741
VQM gain/loss	0	-0.1615	0.66599	0	1.01688	0.54933	0	1.40058	1.20449	0	0.26433	0.42546

The T-1 Network Simulation (The Data Duplication)												
	5 percent			**10 percent**			**30 percent**			**50 percent**		
	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps
PSNR (AVG)	19.9057	20.153	19.0793	19.978	19.6932	19.8598	20.4747	19.938	19.416	19.8756	19.8665	19.7549
PSNR (MIN)	14.0178	13.2968	10.4906	11.2278	13.185	12.8811	15.5577	13.8428	12.7038	13.9763	11.7365	12.5007
PSNR (MAX)	35.5489	36.4585	31.5556	36.0099	36.4236	31.7865	35.4642	36.1089	31.3596	35.3679	35.6727	30.9867
PSNR (SD)	3.83776	3.81055	3.71884	3.85542	3.97592	2.98222	3.74324	3.79398	3.36348	4.05721	3.73323	2.95736
PSNR (95% Confidence Level)	0.05913	0.05871	0.0573	0.0594	0.06126	0.04595	0.05767	0.05845	0.05182	0.06251	0.05752	0.04556
Lower limit	19.8466	20.0943	19.0221	19.9186	19.632	19.8138	20.417	19.8795	19.3641	19.8131	19.809	19.7094
Upper limit	19.9648	20.2117	19.1366	20.0374	19.7545	19.9057	20.5324	19.9964	19.4678	19.9381	19.924	19.8005
PSNR gain/loss	0	-0.2472	0.82637	0	0.28472	0.11821	0	0.53674	1.05873	0	0.00912	0.12068
SSIM (AVG)	0.83671	0.85407	0.80491	0.83395	0.83432	0.82907	0.8478	0.84097	0.81723	0.83482	0.83031	0.81751
SSIM (MIN)	0.65861	0.62287	0.57784	0.61118	0.4826	0.50877	0.69504	0.6628	0.56322	0.65438	0.65762	0.57071
SSIM (MAX)	0.97983	0.98331	0.97289	0.97316	0.98358	0.98054	0.97981	0.98334	0.97206	0.98204	0.97123	0.97966
SSIM (SD)	0.0825	0.07655	0.10032	0.08058	0.08301	0.08359	0.07295	0.07558	0.09404	0.07981	0.08405	0.11091
SSIM (95% Confidence Level)	0.00127	0.00118	0.00155	0.00124	0.00128	0.00129	0.00112	0.00116	0.00145	0.00123	0.00129	0.00171
Lower limit	0.83544	0.85289	0.80337	0.83271	0.83304	0.82778	0.84667	0.83981	0.81579	0.83359	0.82902	0.8158
Upper limit	0.83798	0.85525	0.80646	0.83519	0.8356	0.83036	0.84892	0.84213	0.81868	0.83605	0.83161	0.81922
SSIM gain/loss	0	-0.0174	0.03179	0	-0.0004	0.00488	0	0.00683	0.03056	0	0.00451	0.01731
VQM (AVG)	8.25173	8.1822	9.0635	8.32346	8.68653	8.34922	7.9456	8.50885	8.77621	8.2406	8.50784	8.53264
VQM (MIN)	1.10152	0.97615	1.59192	1.60199	0.91827	1.06304	1.04778	0.94688	1.72543	1.09723	1.0576	1.57988
VQM (MAX)	13.1426	13.3169	15.6282	13.2866	13.6977	15.1509	12.2776	13.095	14.3976	13.545	13.8939	13.1587
VQM (SD)	2.38808	2.39013	2.7397	2.21279	2.52459	2.41481	2.24634	2.42813	2.62106	2.59328	2.37459	2.26877
VQM (95% Confidence Level)	0.03679	0.03682	0.04221	0.03409	0.0389	0.0372	0.03461	0.03741	0.04038	0.03995	0.03658	0.03495
Lower limit	8.21493	8.14538	9.02129	8.28937	8.64763	8.31202	7.91099	8.47144	8.73583	8.20065	8.47126	8.49768
Upper limit	8.28852	8.21902	9.10571	8.35755	8.72543	8.38643	7.98021	8.54625	8.81659	8.28056	8.54443	8.56759
VQM gain/loss	0	-0.0695	0.81177	0	0.36307	0.02576	0	0.56325	0.83061	0	0.26724	0.29203

❖ The T-1 Network Simulation (The Data Reorder)												
	5 percent			**10 percent**			**30 percent**			**50 percent**		
	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps
PSNR (AVG)	20.2512	19.7957	19.7742	19.9935	19.7201	19.9617	19.9883	19.9126	9.82155	19.8968	18.7235	19.476
PSNR (MIN)	12.11	14.0774	8.72279	13.1876	12.0769	9.54059	10.0448	13.1314	9.13029	13.2325	7.75121	12.8573
PSNR (MAX)	30.9348	25.7933	34.532	31.4924	35.5071	35.1912	34.6902	34.2351	10.1559	34.5911	35.1576	31.2719
PSNR (SD)	2.94647	2.15166	3.87537	2.88841	3.82417	4.06869	3.83132	3.74776	0.15769	3.77199	3.78668	3.10551
PSNR (95% Confidence Level)	0.0454	0.03315	0.05971	0.0445	0.05892	0.06269	0.05903	0.05774	0.00243	0.05811	0.05834	0.04785
Lower limit	20.2058	19.7625	19.7145	19.949	19.6611	19.899	19.9293	19.8548	9.81912	19.8387	18.6651	19.4282
Upper limit	20.2966	19.8288	19.8339	20.038	19.779	20.0244	20.0473	19.9703	9.82398	19.9549	18.7818	19.5238
PSNR gain/loss	0	0.45557	0.47705	0	0.27346	0.0318	0	0.0757	10.1667	0	1.17335	0.42083
SSIM (AVG)	0.8464	0.84619	0.83526	0.84262	0.83607	0.82843	0.84273	0.83856	0.83303	0.8369	0.81368	0.8215
SSIM (MIN)	0.58687	0.60818	0.42078	0.45985	0.63462	0.6352	0.49805	0.56133	0.56558	0.61752	0.50856	0.58664
SSIM (MAX)	0.947	0.97102	0.98081	0.97951	0.98212	0.97262	0.98126	0.97846	0.96852	0.97905	0.98142	0.97198
SSIM (SD)	0.06455	0.07894	0.09056	0.09111	0.08079	0.08232	0.08703	0.08638	0.08979	0.082	0.09489	0.09633
SSIM (95% Confidence Level)	0.00099	0.00122	0.0014	0.0014	0.00124	0.00127	0.00134	0.00133	0.00177	0.00126	0.00146	0.00148
Lower limit	0.84541	0.84498	0.83387	0.84121	0.83483	0.82717	0.84139	0.83723	0.83127	0.83563	0.81222	0.82002
Upper limit	0.84739	0.84741	0.83666	0.84402	0.83732	0.8297	0.84407	0.83989	0.8348	0.83816	0.81514	0.82298
SSIM gain/loss	0	0.00021	0.01114	0	0.00654	0.01418	0	0.00417	0.00969	0	0.02322	0.0154
VQM (AVG)	8.10199	8.55796	8.1178	8.25124	8.48716	8.64474	8.18838	8.26867	8.34093	8.27435	9.19332	8.65162
VQM (MIN)	1.88399	1.75813	4.97894	1.48009	1.77482	1.42093	1.57938	1.72645	2.24817	1.67784	1.53027	1.94725
VQM (MAX)	14.4326	18.1355	13.2563	12.9895	17.6361	13.2366	16.8112	16.5385	13.9065	13.4054	17.7482	13.9815
VQM (SD)	2.13455	2.55746	1.76833	2.11122	2.54793	2.45569	0.6548	2.45324	2.30665	2.33931	2.651	2.37009
VQM (95% Confidence Level)	0.03289	0.0394	0.02724	0.03253	0.03926	0.03783	0.03009	0.0378	0.03554	0.03604	0.04084	0.03652
Lower limit	8.0691	8.51855	8.09056	8.21871	8.4479	8.60691	8.15829	8.23087	8.30539	8.2383	9.15248	8.6151
Upper limit	8.13487	8.59736	8.14505	8.28376	8.52642	8.68258	8.21847	8.30646	8.37647	8.31039	9.23417	8.68813
VQM gain/loss	0	0.45597	0.01581	0	0.23592	0.39351	0	0.08029	0.15255	0	0.91898	0.37727

❖ The T-1 Network Simulation (The Data Delay and Jitter)						
	The Data Delay 500 ms			**The Jitter 500 ms, Delay 300 ms**		
	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps
PSNR (AVG)	19.7739	19.36	19.158	19.6911	19.8307	19.3965
PSNR (MIN)	9.80826	10.9982	10.6773	12.5408	13.1016	11.0015
PSNR (MAX)	35.4602	31.7072	35.6991	34.5277	35.3567	30.4696
PSNR (SD)	4.43692	3.38663	3.21708	3.87738	3.66002	3.31344
PSNR (95% Confidence Level)	0.06836	0.05218	0.04956	0.05974	0.05639	0.05105
Lower limit	19.7055	19.3078	19.1084	19.6313	19.7743	19.3454
Upper limit	19.8422	19.4122	19.2076	19.7508	19.887	19.4475
PSNR gain/loss	0	0.41389	0.61589	0	-0.1396	0.29457
SSIM (AVG)	0.8305	0.82168	0.81767	0.83549	0.83719	0.80951
SSIM (MIN)	0.48601	0.5648	0.5045	0.6545	0.61985	0.50782
SSIM (MAX)	0.98012	0.98268	0.97331	0.97877	0.98229	0.97038
SSIM (SD)	0.10151	0.08641	0.09951	0.08394	0.08637	0.10365
SSIM (95% Confidence Level)	0.00156	0.00133	0.00153	0.00129	0.00133	0.0016
Lower limit	0.82893	0.82034	0.81614	0.83419	0.83586	0.80791
Upper limit	0.83206	0.82301	0.81921	0.83678	0.83852	0.8111
SSIM gain/loss	0	0.00882	0.01282	0	-0.0017	0.02598
VQM (AVG)	8.77319	8.78129	9.03264	8.37017	8.45943	8.7054
VQM (MIN)	1.88719	1.85749	1.56544	2.27866	1.73717	3.27864
VQM (MAX)	15.479	16.7509	15.4812	13.6443	13.5499	15.5059
VQM (SD)	2.42402	2.71492	2.46472	2.28866	2.41552	2.40697
VQM (95% Confidence Level)	0.03735	0.04183	0.03797	0.03526	0.03722	0.03708
Lower limit	8.73584	8.73946	8.99467	8.33491	8.42222	8.66832
Upper limit	8.81053	8.82312	9.07062	8.40543	8.49665	8.74249
VQM gain/loss	0	0.0081	0.25946	0	0.08927	0.33524

❖ The T-1 Network Simulation (The Random Error)

	Random		
	5 fps	15 fps	30 fps
PSNR (AVG)	20.2168	19.5581	18.9796
PSNR (MIN)	12.0338	13.6589	10.696
PSNR (MAX)	33.7808	34.4866	32.3242
PSNR (SD)	3.77986	3.61949	3.87907
PSNR (95% Confidence Level)	0.05824	0.05576	0.05976
Lower limit	20.1586	19.5023	18.9198
Upper limit	20.2751	19.6138	19.0393
PSNR gain/loss	0	0.65878	1.23726
SSIM (AVG)	0.84644	0.83018	0.80441
SSIM (MIN)	0.61407	0.53795	0.60405
SSIM (MAX)	0.97812	0.98104	0.97538
SSIM (SD)	0.07923	0.08238	0.1015
SSIM (95% Confidence Level)	0.00122	0.00127	0.00156
Lower limit	0.84522	0.82891	0.80285
Upper limit	0.84766	0.83145	0.80598
SSIM gain/loss	0	0.01626	0.04203
VQM (AVG)	8.29025	8.80917	8.98437
VQM (MIN)	2.6386	2.20389	1.37041
VQM (MAX)	13.9696	13.7704	15.2496
VQM (SD)	2.38366	2.48231	2.74697
VQM (95% Confidence Level)	0.03672	0.03824	0.04232
Lower limit	8.25352	8.77092	8.94205
Upper limit	8.32697	8.84741	9.0267
VQM gain/loss	0	0.51892	0.69413

❖ The 10 Mbps Network Simulation (The Data Corruption)												
	5 percent			**10 percent**			**30 percent**			**50 percent**		
	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps
PSNR (AVG)	20.9037	19.576	19.2864	19.833	19.6378	20.0444	19.9821	18.9183	18.6462	20.1389	19.1303	19.1639
PSNR (MIN)	14.8147	13.6868	12.2659	11.27	13.2529	14.5839	10.4658	11.8742	10.9936	14.4372	11.0272	12.4119
PSNR (MAX)	35.6005	36.1578	37.1317	33.236	36.0778	37.5445	35.0794	33.8187	30.4116	32.8886	34.7162	34.2087
PSNR (SD)	3.56012	3.44092	3.22122	3.51856	3.52524	3.38571	3.52899	3.88137	3.60446	3.06389	4.03409	3.59075
PSNR (95% Confidence Level)	0.05485	0.05679	0.04963	0.05421	0.05431	0.05216	0.05437	0.0598	0.05553	0.0472	0.06215	0.05532
Lower limit	20.8489	19.5192	19.2368	19.7788	19.5834	19.9923	19.9277	18.8585	18.5906	20.0917	19.0682	19.1086
Upper limit	20.9586	19.6328	19.336	19.8873	19.6921	20.0966	20.0365	18.9781	18.7017	20.1861	19.1925	19.2192
PSNR gain/loss	0	1.32772	1.61731	0	0.19529	-0.2114	0	1.0638	1.33591	0	1.00859	0.97499
SSIM (AVG)	0.8647	0.86289	0.80398	0.83091	0.83216	0.8262	0.82231	0.80847	0.80212	0.83529	0.81685	0.79516
SSIM (MIN)	0.62557	0.63235	0.57013	0.57377	0.54153	0.60029	0.59249	0.61068	0.51207	0.63926	0.5746	0.58207
SSIM (MAX)	0.98052	0.95435	0.98351	0.97566	0.98238	0.98291	0.98113	0.97824	0.97281	0.97667	0.98007	0.97833
SSIM (SD)	0.06168	0.07555	0.10074	0.08058	0.08782	0.08793	0.09309	0.09699	0.10192	0.07417	0.09843	0.10427
SSIM (95% Confidence Level)	0.00095	0.00096	0.00155	0.00124	0.00135	0.00135	0.00143	0.00149	0.00157	0.00114	0.00152	0.00161
Lower limit	0.86375	0.86193	0.80242	0.82967	0.83081	0.82485	0.82088	0.80697	0.80055	0.83415	0.81534	0.79355
Upper limit	0.86565	0.86385	0.80553	0.83215	0.83352	0.82756	0.82375	0.80996	0.80369	0.83644	0.81837	0.79676
SSIM gain/loss	0	0.00181	0.06072	0	-0.0013	0.00471	0	0.01384	0.0202	0	0.01844	0.04014
VQM (AVG)	7.49291	7.63692	8.92691	8.33378	8.378	8.67729	8.40243	9.19478	9.29107	8.16346	9.0038	9.06337
VQM (MIN)	1.05384	1.09046	1.02823	0.94406	2.79683	1.02127	2.72269	2.01896	3.09981	2.849	1.6676	2.80859
VQM (MAX)	12.6996	12.6916	14.8435	13.3264	14.8779	13.8947	15.5852	13.8383	15.0353	12.5948	15.2142	14.3755
VQM (SD)	2.02833	2.03195	2.46469	2.27886	2.22411	2.50015	2.31896	2.7242	2.59213	1.87026	2.78576	2.52976
VQM (95% Confidence Level)	0.03125	0.0359	0.03797	0.03511	0.03427	0.03852	0.03573	0.04197	0.03994	0.02881	0.04292	0.03898
Lower limit	7.46166	7.60102	8.88894	8.29867	8.34374	8.63877	8.3667	9.15281	9.25114	8.13464	8.96088	9.0244
Upper limit	7.52416	7.67282	8.96488	8.36889	8.41227	8.71581	8.43816	9.23675	9.33101	8.19227	9.04672	9.10235
VQM gain/loss	0	0.14401	1.434	0	0.04423	0.34351	0	0.79235	0.88864	0	0.84034	0.89991

❖ The 10 Mbps Network Simulation (The Data Loss)												
	5 percent			**10 percent**			**30 percent**			**50 percent**		
	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps
PSNR (AVG)	20.5269	19.5423	19.5529	20.5645	19.6891	19.8544	20.6199	19.9019	19.5517	19.7694	19.2844	19.712
PSNR (MIN)	15.3026	10.9162	10.4879	14.8971	13.2283	9.73908	12.3713	13.043	11.8737	12.8877	12.4316	10.2276
PSNR (MAX)	35.5546	36.244	31.9921	37.7724	37.5314	32.6749	39.0495	38.3983	34.3964	29.5325	30.3105	32.153
PSNR (SD)	3.56933	4.23805	3.52174	3.70699	4.27224	3.3584	3.98341	4.21949	3.44998	2.5484	2.73493	3.58985
PSNR (95% Confidence Level)	0.05499	0.06529	0.05426	0.05711	0.06582	0.05174	0.06137	0.06501	0.05315	0.03926	0.04214	0.05531
Lower limit	20.4719	19.477	19.4987	20.5074	19.6233	19.8027	20.5585	19.8369	19.4985	19.7301	19.2422	19.6567
Upper limit	20.5819	19.6076	19.6072	20.6216	19.7549	19.9062	20.6813	19.9669	19.6049	19.8087	19.3265	19.7673
PSNR gain/loss	0	0.98458	0.97394	0	0.8754	0.71008	0	0.71798	1.0682	0	0.48504	0.05735
SSIM (AVG)	0.8561	0.82314	0.8172	0.85076	0.82277	0.82844	0.84581	0.8327	0.8129	0.82007	0.81739	0.81072
SSIM (MIN)	0.6904	0.58491	0.54764	0.66229	0.62187	0.45411	0.58486	0.65344	0.62348	0.54712	0.61304	0.55517
SSIM (MAX)	0.98007	0.98307	0.97413	0.98149	0.98473	0.97603	0.98285	0.98515	0.97955	0.97489	0.96232	0.96393
SSIM (SD)	0.0679	0.09712	0.09668	0.07554	0.09867	0.09313	0.07904	0.08441	0.09357	0.09671	0.07074	0.08141
SSIM (95% Confidence Level)	0.00105	0.0015	0.00149	0.00116	0.00152	0.00143	0.00122	0.0013	0.00144	0.00149	0.00109	0.00125
Lower limit	0.85505	0.82165	0.81571	0.84959	0.82125	0.827	0.8446	0.8314	0.81145	0.81858	0.8163	0.80947
Upper limit	0.85714	0.82464	0.81869	0.85192	0.82429	0.82987	0.84703	0.834	0.81434	0.82156	0.81848	0.81198
SSIM gain/loss	0	0.03295	0.0389	0	0.02799	0.02232	0	0.01312	0.03292	0	0.00268	0.00935
VQM (AVG)	7.94775	8.76489	8.7682	7.99957	8.61666	8.38644	7.65861	8.55391	8.67767	8.37846	8.86813	8.50366
VQM (MIN)	1.88928	1.1418	2.06084	0.96907	0.95782	1.34534	0.76771	0.83213	1.14123	3.15848	2.33765	3.02358
VQM (MAX)	12.2117	15.448	15.9615	12.5677	13.389	17.3998	13.7362	13.4556	13.0628	13.9827	14.058	16.4309
VQM (SD)	2.06104	2.73624	2.60887	2.22651	2.72849	2.44467	2.14645	2.55703	2.47414	1.88055	2.23182	2.56648
VQM (95% Confidence Level)	0.03175	0.04216	0.04019	0.0343	0.04204	0.03766	0.03307	0.0394	0.03812	0.02897	0.03439	0.03954
Lower limit	7.916	8.72273	8.72801	7.96527	8.57463	8.34877	7.62554	8.51451	8.63955	8.34949	8.83374	8.46412
Upper limit	7.9795	8.80705	8.80839	8.03388	8.6587	8.4241	7.69168	8.5933	8.71578	8.40743	8.90251	8.5432
VQM gain/loss	0	0.81714	0.82045	0	0.61709	0.38687	0	0.8953	1.01906	0	0.48967	0.1252

❖ The 10 Mbps Network Simulation (The Data Duplication)												
	5 percent			**10 percent**			**30 percent**			**50 percent**		
	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps
PSNR (AVG)	19.8789	18.9282	18.7922	20.0084	9.61876	19.2263	20.7712	20.1761	19.6463	20.8681	18.9227	19.2436
PSNR (MIN)	11.1052	11.4054	12.5609	13.7437	7.82844	12.5414	13.7155	10.0104	14.5157	14.1713	12.9796	13.6187
PSNR (MAX)	35.0269	33.1756	30.882	38.1927	10.1714	35.0417	38.626	35.4917	38.1523	36.4887	37.3327	36.5434
PSNR (SD)	3.66081	2.85188	3.54085	4.25752	0.38296	3.95726	4.02301	3.335	3.49842	3.5337	4.38251	4.00432
PSNR (95% Confidence Level)	0.0564	0.04394	0.05455	0.06559	0.0059	0.06097	0.06198	0.05138	0.0539	0.05444	0.06752	0.06169
Lower limit	19.8225	18.8842	18.7377	19.9428	9.61286	19.1653	20.7092	20.1247	19.5924	20.8137	18.8552	19.1819
Upper limit	19.9353	18.9721	18.8468	20.074	9.62466	19.2873	20.8332	20.2275	19.7002	20.9226	18.9903	19.3052
PSNR gain/loss	0	0.95072	1.08666	0	10.3896	0.78211	0	0.59505	1.12489	0	1.94541	1.62459
SSIM (AVG)	0.83632	0.80554	0.79089	0.83956	0.82728	0.80636	0.85844	0.84109	0.83457	0.85321	0.80878	0.81433
SSIM (MIN)	0.60711	0.60863	0.458	0.66601	0.62098	0.6021	0.68099	0.43165	0.66558	0.66962	0.59049	0.62713
SSIM (MAX)	0.97873	0.97087	0.97594	0.9819	0.95972	0.98017	0.98239	0.98149	0.98462	0.9806	0.98239	0.98387
SSIM (SD)	0.07357	0.0893	0.10813	0.07728	0.08967	0.10824	0.06694	0.10242	0.07968	0.06959	0.10156	0.09158
SSIM (95% Confidence Level)	0.00113	0.00138	0.00167	0.00119	0.00152	0.00167	0.00103	0.00158	0.00123	0.00107	0.00156	0.00141
Lower limit	0.83519	0.80417	0.78922	0.83837	0.82576	0.80469	0.85741	0.83952	0.83334	0.85213	0.80722	0.81292
Upper limit	0.83746	0.80692	0.79255	0.84075	0.8288	0.80803	0.85947	0.84267	0.83579	0.85428	0.81035	0.81574
SSIM gain/loss	0	0.03078	0.04544	0	0.01228	0.0332	0	0.01734	0.02387	0	0.04442	0.03888
VQM (AVG)	8.45881	9.29073	9.00385	8.29323	8.72921	8.96587	7.51557	7.95928	8.64998	7.64618	8.94109	9.17818
VQM (MIN)	2.56171	2.12787	3.1739	1.07294	1.08226	1.12952	0.81896	1.13092	0.80665	1.00981	0.95674	0.87019
VQM (MAX)	14.9371	13.7634	14.7404	13.1343	13.8924	14.0457	12.8005	17.0832	12.6548	12.7475	13.4463	13.9162
VQM (SD)	2.10919	2.59824	2.27647	2.36235	2.98833	2.80049	2.0613	2.50597	2.35177	2.04248	2.73843	2.84002
VQM (95% Confidence Level)	0.0325	0.04003	0.03507	0.0364	0.03523	0.04315	0.03176	0.03861	0.03623	0.03147	0.04219	0.04376
Lower limit	8.42632	9.2507	8.96878	8.25683	8.69398	8.92272	7.48381	7.92067	8.61375	7.61471	8.8989	9.13442
Upper limit	8.49131	9.33076	9.03893	8.32962	8.76443	9.00901	7.54733	7.99789	8.68621	7.67765	8.98328	9.22193
VQM gain/loss	0	0.83192	0.54504	0	0.43598	0.67264	0	0.44371	1.13441	0	1.29491	1.532

❖ The 10 Mbps Network Simulation (The Data Reorder)												
	5 percent			**10 percent**			**30 percent**			**50 percent**		
	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps
PSNR (AVG)	19.6235	19.5911	19.2854	19.9603	19.8553	19.5768	19.2624	17.9228	18.1799	20.0481	19.6261	19.5794
PSNR (MIN)	12.8458	12.4885	12.7107	10.2998	12.544	12.6615	13.1398	10.5353	12.6446	12.512	12.5107	13.8793
PSNR (MAX)	33.4991	32.1747	30.897	35.8108	37.0964	35.1602	30.3686	30.0355	32.1991	30.4032	30.1744	34.6829
PSNR (SD)	2.71861	2.56308	3.4955	4.45173	3.71625	4.41162	3.28572	3.30845	3.28546	3.0512	3.28469	3.14168
PSNR (95% Confidence Level)	0.04188	0.03395	0.05385	0.06859	0.05726	0.06634	0.05062	0.05097	0.05062	0.04701	0.04386	0.0484
Lower limit	19.5816	19.5572	19.2315	19.8917	19.798	19.5104	19.2118	17.8718	18.1293	20.0011	19.5822	19.531
Upper limit	19.6654	19.6251	19.3392	20.0289	19.9125	19.6431	19.313	17.9738	18.2306	20.0951	19.6699	19.6278
PSNR gain/loss	0	0.03239	0.33812	0	0.10504	0.38357	0	1.33963	1.08247	0	0.42204	0.46873
SSIM (AVG)	0.81685	0.80626	0.80665	0.83532	0.83161	0.81542	0.81603	0.78707	0.76866	0.83516	0.83203	0.81718
SSIM (MIN)	0.58796	0.4406	0.46486	0.48175	0.45297	0.52355	0.59736	0.50753	0.49068	0.55214	0.54257	0.63139
SSIM (MAX)	0.97427	0.94778	0.97749	0.98007	0.95647	0.98225	0.97386	0.96853	0.97414	0.97507	0.95083	0.97952
SSIM (SD)	0.09301	0.09348	0.09222	0.10595	0.0533	0.09855	0.09191	0.10487	0.11219	0.07994	0.0744	0.09134
SSIM (95% Confidence Level)	0.00143	0.00182	0.00142	0.00163	0.00182	0.00152	0.00142	0.00162	0.00173	0.00123	0.00184	0.00141
Lower limit	0.81542	0.80444	0.80523	0.83368	0.82979	0.81391	0.81462	0.78545	0.76693	0.83393	0.83019	0.81578
Upper limit	0.81828	0.80808	0.80807	0.83695	0.83343	0.81694	0.81745	0.78868	0.77039	0.83639	0.83387	0.81859
SSIM gain/loss	0	0.01059	0.0102	0	0.00371	0.01989	0	0.02897	0.04738	0	0.00313	0.01798
VQM (AVG)	8.5235	8.56333	8.63277	8.38315	8.46464	8.49964	8.79151	9.87908	9.94039	8.09864	8.53846	8.67461
VQM (MIN)	2.29194	2.04389	2.30687	1.4701	1.58917	0.94268	1.75827	1.93078	1.58559	1.64507	1.59901	1.16575
VQM (MAX)	14.0519	14.7308	14.5314	16.5246	15.1971	14.0654	14.2062	16.3431	14.7158	14.2214	14.6159	13.0791
VQM (SD)	2.06323	2.83429	2.3227	2.89194	2.49899	2.38696	2.48741	2.85029	2.54589	2.09789	2.86377	2.24921
VQM (95% Confidence Level)	0.03179	0.01329	0.03579	0.04456	0.03539	0.03678	0.03832	0.04391	0.03922	0.03232	0.03331	0.03465
Lower limit	8.49171	8.55005	8.59698	8.3386	8.42925	8.46287	8.75319	9.83516	9.90117	8.06632	8.50515	8.63996
Upper limit	8.55528	8.57662	8.66855	8.42771	8.50003	8.53642	8.82984	9.92299	9.97961	8.13097	8.57177	8.70926
VQM gain/loss	0	0.03984	0.10927	0	0.08149	0.11649	0	1.08757	1.14888	0	0.43982	0.57597

❖ The 10 Mbps Network Simulation (The Data Delay and Jitter)						
	The Data Delay 500 ms			**The Jitter 500 ms, Delay 300 ms**		
	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps
PSNR (AVG)	20.024	19.9237	19.549	20.7949	20.0638	19.5568
PSNR (MIN)	13.7404	14.1321	11.5336	11.4681	9.95591	12.5458
PSNR (MAX)	34.6655	35.3593	36.2661	35.9936	36.2296	37.4624
PSNR (SD)	3.28174	4.02894	4.4657	3.49104	3.51562	3.80471
PSNR (95% Confidence Level)	0.05056	0.06207	0.0688	0.05379	0.05416	0.05862
Lower limit	19.9734	19.8616	19.4802	20.7411	20.0097	19.4982
Upper limit	20.0746	19.9858	19.6178	20.8487	20.118	19.6155
PSNR gain/loss	0	0.10027	0.47504	0	0.73102	1.23803
SSIM (AVG)	0.84514	0.82697	0.82425	0.86633	0.84296	0.81186
SSIM (MIN)	0.65213	0.47364	0.63279	0.55975	0.57379	0.61651
SSIM (MAX)	0.97996	0.98361	0.98027	0.9805	0.98217	0.98348
SSIM (SD)	0.07399	0.09515	0.09312	0.05781	0.07921	0.09961
SSIM (95% Confidence Level)	0.00114	0.00147	0.00143	0.00089	0.00122	0.00153
Lower limit	0.844	0.82551	0.82282	0.86544	0.84174	0.81033
Upper limit	0.84628	0.82844	0.82569	0.86722	0.84418	0.8134
SSIM gain/loss	0	0.01817	0.02089	0	0.02337	0.05447
VQM (AVG)	8.21276	8.69508	8.49558	7.33003	8.23034	8.78031
VQM (MIN)	1.16869	1.36252	2.19667	1.0299	0.97171	0.87518
VQM (MAX)	13.1957	14.5607	13.1741	14.692	15.2303	14.1144
VQM (SD)	2.26302	2.82001	2.36197	1.80995	2.32132	2.57456
VQM (95% Confidence Level)	0.03487	0.04345	0.03639	0.02789	0.03576	0.03967
Lower limit	8.17789	8.65163	8.45919	7.30214	8.19457	8.74064
Upper limit	8.24762	8.73853	8.53197	7.35791	8.2661	8.81997
VQM gain/loss	0	0.48232	0.28282	0	0.90031	1.45028

❖ The 10 Mbps Network Simulation (The Random Error)

	Random		
	5 fps	15 fps	30 fps
PSNR (AVG)	20.3854	19.8865	19.6843
PSNR (MIN)	11.8798	12.0556	13.7898
PSNR (MAX)	37.8855	35.2556	34.302
PSNR (SD)	3.48551	3.15774	3.27014
PSNR (95% Confidence Level)	0.0537	0.05243	0.04416
Lower limit	20.3317	19.8341	19.6402
Upper limit	20.4391	19.939	19.7285
PSNR gain/loss	0	0.49883	0.70103
SSIM (AVG)	0.84229	0.86599	0.86815
SSIM (MIN)	0.62499	0.6518	0.56755
SSIM (MAX)	0.98392	0.9557	0.92813
SSIM (SD)	0.08428	0.08333	0.08134
SSIM (95% Confidence Level)	0.0013	0.00182	0.00148
Lower limit	0.84099	0.86417	0.86667
Upper limit	0.84359	0.86781	0.86963
SSIM gain/loss	0	-0.0237	-0.0259
VQM (AVG)	8.11789	8.30545	9.01283
VQM (MIN)	0.85097	0.90324	0.82064
VQM (MAX)	13.805	13.4538	14.7662
VQM (SD)	2.38684	2.85418	2.46934
VQM (95% Confidence Level)	0.03677	0.03316	0.03723
Lower limit	8.08111	8.27229	8.9756
Upper limit	8.15466	8.33861	9.05006
VQM gain/loss	0	0.18757	0.89494

❖ The 100 Mbps Network Simulation (The Data Loss)												
	5 percent			**10 percent**			**30 percent**			**50 percent**		
	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps
PSNR (AVG)	19.8064	19.4248	18.2698	20.8521	19.1963	19.1507	19.963	19.3458	19.4026	20.3641	19.4613	19.7653
PSNR (MIN)	12.9205	13.0177	9.7825	13.0974	12.4332	12.1725	11.0418	13.5692	12.926	13.5012	10.2693	12.5818
PSNR (MAX)	33.938	32.919	33.1278	36.0673	33.5101	33.3722	33.9903	34.6989	34.8801	33.5457	34.5854	37.8667
PSNR (SD)	3.63851	3.11433	4.79818	3.35418	3.29848	3.26456	3.50867	3.72956	3.27648	3.4026	4.00553	3.57811
PSNR (95% Confidence Level)	0.05606	0.04798	0.07392	0.05168	0.05082	0.05408	0.05406	0.05746	0.05048	0.05242	0.06171	0.05513
Lower limit	19.7503	19.3769	18.1959	20.8004	19.1455	19.0966	19.9089	19.2884	19.3521	20.3117	19.3996	19.7101
Upper limit	19.8624	19.4728	18.3437	20.9037	19.2471	19.2048	20.017	19.4033	19.453	20.4165	19.523	19.8204
PSNR gain/loss	0	0.38154	1.53656	0	1.65574	1.70135	0	0.61715	0.56041	0	0.90281	0.59884
SSIM (AVG)	0.82638	0.81745	0.77643	0.8657	0.8208	0.8262	0.83321	0.8204	0.8123	0.8501	0.82263	0.81055
SSIM (MIN)	0.5616	0.58707	0.47002	0.6684	0.61484	0.65324	0.52602	0.62269	0.60329	0.66131	0.53115	0.60564
SSIM (MAX)	0.97893	0.97582	0.97631	0.98011	0.97712	0.96093	0.9781	0.9807	0.98038	0.97743	0.98029	0.98373
SSIM (SD)	0.09729	0.08972	0.13831	0.06008	0.08732	0.08336	0.08203	0.08701	0.10059	0.07158	0.09276	0.09555
SSIM (95% Confidence Level)	0.0015	0.00138	0.00213	0.00093	0.00135	0.00182	0.00126	0.00134	0.00155	0.0011	0.00143	0.00147
Lower limit	0.82488	0.81607	0.7743	0.86477	0.81945	0.82438	0.83195	0.81906	0.81075	0.849	0.8212	0.80908
Upper limit	0.82788	0.81884	0.77856	0.86662	0.82214	0.82802	0.83448	0.82174	0.81385	0.8512	0.82406	0.81203
SSIM gain/loss	0	0.00893	0.04995	0	0.0449	0.0395	0	0.01281	0.02091	0	0.02747	0.03955
VQM (AVG)	8.26821	8.83268	9.61378	7.47384	8.96471	8.92136	8.38764	9.04413	8.77055	7.91761	8.76681	8.68391
VQM (MIN)	1.16325	2.16498	2.94988	2.68165	1.90546	1.98085	2.55131	1.67623	2.81067	2.19565	1.40171	1.02954
VQM (MAX)	14.0993	13.5719	17.0042	12.7861	13.8897	13.0071	14.7599	13.5803	13.8854	13.0649	15.9553	13.6399
VQM (SD)	2.5595	2.39365	3.47158	1.74775	2.44477	2.82262	2.15324	2.6169	2.48824	2.00659	2.6847	2.389
VQM (95% Confidence Level)	0.03943	0.03688	0.05349	0.02693	0.03767	0.03267	0.03317	0.04032	0.03834	0.03092	0.04136	0.03681
Lower limit	8.22878	8.7958	9.56029	7.44692	8.92704	8.88869	8.35446	9.00381	8.73222	7.88669	8.72544	8.64711
Upper limit	8.30764	8.86956	9.66726	7.50077	9.00237	8.95404	8.42081	9.08445	8.80889	7.94852	8.80817	8.72072
VQM gain/loss	0	0.56447	1.34557	0	1.49086	1.44752	0	0.65649	0.38291	0	0.8492	0.76631

❖ The 100 Mbps Network Simulation (The Data Duplication)												
	5 percent			**10 percent**			**30 percent**			**50 percent**		
	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps
PSNR (AVG)	20.4755	19.8494	18.9342	19.5611	18.7041	18.3017	19.2437	19.3981	19.211	20.8092	19.8904	16.9341
PSNR (MIN)	14.2186	13.3241	10.84	11.2961	10.5503	11.1129	13.2701	13.3616	11.8666	14.7017	10.7296	9.8457
PSNR (MAX)	32.2262	37.748	33.4585	37.1199	30.0588	31.7938	34.7863	30.8235	26.935	39.2924	38.7622	27.1802
PSNR (SD)	2.88469	2.96318	3.49672	3.69265	3.79261	3.85308	3.61084	2.95153	2.3696	3.77366	4.00542	3.35648
PSNR (95% Confidence Level)	0.04444	0.04565	0.05387	0.05689	0.05843	0.05936	0.05563	0.04547	0.03651	0.05814	0.06171	0.05171
Lower limit	20.431	19.8037	18.8804	19.5042	18.6457	18.2423	19.1881	19.3527	19.1745	20.7511	19.8287	16.8824
Upper limit	20.5199	19.895	18.9881	19.618	18.7626	18.361	19.2993	19.4436	19.2475	20.8673	19.9521	16.9858
PSNR gain/loss	0	0.62611	1.54123	0	0.85694	1.25943	0	-0.1544	0.03268	0	0.91884	3.87507
SSIM (AVG)	0.85675	0.81245	0.80886	0.80672	0.80638	0.80062	0.83728	0.83284	0.80378	0.86464	0.84236	0.73645
SSIM (MIN)	0.65219	0.58701	0.46196	0.58304	0.51051	0.61114	0.60065	0.63392	0.57847	0.66975	0.5606	0.44028
SSIM (MAX)	0.97676	0.98346	0.9781	0.98319	0.97395	0.97377	0.96821	0.97121	0.97968	0.98267	0.98528	0.95766
SSIM (SD)	0.0643	0.09049	0.10002	0.09777	0.1123	0.09479	0.07297	0.0805	0.10305	0.06128	0.08414	0.13141
SSIM (95% Confidence Level)	0.00099	0.00139	0.00154	0.00151	0.00173	0.00146	0.00112	0.00124	0.00159	0.00094	0.0013	0.00202
Lower limit	0.85576	0.81106	0.80732	0.80521	0.80465	0.79916	0.83616	0.8316	0.8022	0.86369	0.84106	0.73442
Upper limit	0.85774	0.81384	0.8104	0.80822	0.80811	0.80208	0.83841	0.83408	0.80537	0.86558	0.84366	0.73847
SSIM gain/loss	0	0.0443	0.04789	0	0.00034	0.0061	0	0.00444	0.0335	0	0.02228	0.12819
VQM (AVG)	7.98792	8.56969	9.22338	8.69146	9.22396	9.63323	8.53477	8.63726	9.09934	7.60826	8.08988	10.7873
VQM (MIN)	1.4622	0.82934	1.33641	0.92845	1.7797	1.57343	2.34632	1.72788	1.15834	0.73386	0.75663	2.68987
VQM (MAX)	12.9522	13.61	16.1097	14.8134	15.9666	14.4975	13.7714	13.2872	14.0699	12.7705	15.5611	17.0828
VQM (SD)	1.93509	2.14381	2.57297	2.4058	2.91545	2.83635	1.96673	2.26964	2.63154	2.10767	2.43923	2.95713
VQM (95% Confidence Level)	0.02981	0.03303	0.03964	0.03707	0.04492	0.0437	0.0303	0.03497	0.04054	0.03247	0.03758	0.04556
Lower limit	7.9581	8.53666	9.18374	8.65439	9.17904	9.58953	8.50447	8.60229	9.0588	7.57579	8.0523	10.7418
Upper limit	8.01773	8.60272	9.26302	8.72852	9.26887	9.67693	8.56507	8.67223	9.13988	7.64073	8.12746	10.8329
VQM gain/loss	0	0.58178	1.23547	0	0.5325	0.94178	0	0.10249	0.56457	0	0.48162	3.17906

❖ The 100 Mbps Network Simulation (The Data Reorder)												
	5 percent			**10 percent**			**30 percent**			**50 percent**		
	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps
PSNR (AVG)	20.4526	18.9877	18.6611	20.1314	19.502	18.7811	20.401	19.5584	18.7719	21.1321	19.6311	19.3535
PSNR (MIN)	14.7509	11.3008	10.7841	13.8951	11.8904	12.7921	12.9456	12.5866	12.5896	15.1054	13.5083	10.3484
PSNR (MAX)	36.7356	36.9497	32.3348	36.6986	37.2172	35.2251	37.4027	38.0686	35.1305	37.9651	38.3024	29.3749
PSNR (SD)	3.38638	4.43171	3.69714	4.00402	3.8415	3.16968	3.85692	3.98812	3.25357	3.79717	3.68632	2.6465
PSNR (95% Confidence Level)	0.05217	0.06828	0.05696	0.06169	0.05918	0.05261	0.05942	0.06144	0.06391	0.0585	0.05679	0.04077
Lower limit	20.4004	18.9195	18.6042	20.0697	19.4428	18.7285	20.3416	19.497	18.708	21.0736	19.5743	19.3128
Upper limit	20.5047	19.056	18.7181	20.1931	19.5612	18.8337	20.4604	19.6199	18.8359	21.1906	19.6878	19.3943
PSNR gain/loss	0	1.46482	1.79144	0	0.62937	1.35025	0	0.84257	1.62904	0	1.50105	1.77856
SSIM (AVG)	0.85673	0.81022	0.77823	0.84334	0.82928	0.84726	0.85282	0.83187	0.80776	0.86666	0.83473	0.81251
SSIM (MIN)	0.69923	0.52217	0.41057	0.6358	0.58857	0.54962	0.68395	0.61888	0.6635	0.68134	0.55525	0.54696
SSIM (MAX)	0.98084	0.98421	0.97467	0.98058	0.98416	0.96453	0.98133	0.98489	0.96514	0.98196	0.98542	0.9669
SSIM (SD)	0.06063	0.10539	0.12675	0.07649	0.08071	0.08518	0.0674	0.08909	0.08067	0.06031	0.07815	0.09225
SSIM (95% Confidence Level)	0.00093	0.00162	0.00195	0.00118	0.00124	0.0018	0.00104	0.00137	0.00178	0.00093	0.0012	0.00142
Lower limit	0.8558	0.8086	0.77627	0.84216	0.82804	0.84547	0.85178	0.8305	0.80598	0.86573	0.83353	0.81109
Upper limit	0.85766	0.81184	0.78018	0.84451	0.83053	0.84906	0.85386	0.83325	0.80954	0.86759	0.83594	0.81393
SSIM gain/loss	0	0.04651	0.0785	0	0.01405	-0.0039	0	0.02095	0.04506	0	0.03192	0.05415
VQM (AVG)	7.98412	9.18871	9.35475	8.2666	8.65545	8.99177	7.78137	8.64986	8.61985	7.42202	8.59447	8.69863
VQM (MIN)	1.35097	1.07034	1.45827	1.30849	0.89751	0.84996	1.02721	0.8016	0.83944	0.89251	0.78073	3.42916
VQM (MAX)	12.3176	14.9826	15.1451	12.9567	14.6192	14.2551	12.5635	13.9572	13.5789	11.8151	13.5125	15.753
VQM (SD)	1.94945	2.85087	2.84921	2.30726	2.40044	2.67636	2.00256	2.66179	2.79694	1.91675	2.36857	2.09993
VQM (95% Confidence Level)	0.03003	0.04392	0.0439	0.03555	0.03698	0.03042	0.03085	0.04101	0.04228	0.02953	0.03649	0.03235
Lower limit	7.95408	9.14479	9.31085	8.23105	8.61847	8.96135	7.75052	8.60885	8.57757	7.39249	8.55798	8.66628
Upper limit	8.01415	9.23263	9.39865	8.30214	8.69244	9.02219	7.81223	8.69087	8.66213	7.45155	8.63096	8.73098
VQM gain/loss	0	1.20459	1.37063	0	0.38885	0.72517	0	0.86848	0.83848	0	1.17244	1.27661

❖ The 100 Mbps Network Simulation (The Data Delay and Jitter)						
	The Data Delay 500 ms			**The Jitter 500 ms, Delay 300 ms**		
	5 fps	15 fps	30 fps	5 fps	15 fps	30 fps
PSNR (AVG)	20.5818	19.6512	19.5735	20.7529	20.026	19.3317
PSNR (MIN)	13.3521	12.4117	11.1785	11.3126	10.9248	9.99211
PSNR (MAX)	38.5949	35.1689	35.335	38.4999	33.4666	38.4177
PSNR (SD)	3.99691	3.44638	3.66938	3.93612	3.23354	4.10325
PSNR (95% Confidence Level)	0.06158	0.05688	0.05653	0.06064	0.04982	0.06322
Lower limit	20.5202	19.5943	19.5169	20.6922	19.9762	19.2684
Upper limit	20.6434	19.708	19.63	20.8135	20.0759	19.3949
PSNR gain/loss	0	0.93063	1.00831	0	0.72684	1.42122
SSIM (AVG)	0.8533	0.83028	0.81173	0.86081	0.82615	0.82074
SSIM (MIN)	0.63483	0.62326	0.55951	0.56478	0.51695	0.51677
SSIM (MAX)	0.98238	0.96005	0.98052	0.98208	0.98535	0.97758
SSIM (SD)	0.07427	0.08134	0.10376	0.06487	0.08437	0.09407
SSIM (95% Confidence Level)	0.00114	0.00179	0.0016	0.001	0.0013	0.00145
Lower limit	0.85215	0.82849	0.81013	0.85981	0.82485	0.81929
Upper limit	0.85444	0.83207	0.81332	0.8618	0.82745	0.82219
SSIM gain/loss	0	0.02302	0.04157	0	0.03466	0.04006
VQM (AVG)	7.8202	8.14863	8.45779	7.68004	8.94605	8.14179
VQM (MIN)	0.81182	0.82706	1.14541	0.79784	0.7914	1.23949
VQM (MAX)	13.0897	14.2284	15.3964	14.2098	16.4674	14.7557
VQM (SD)	2.24711	2.98162	2.56535	2.12828	2.61301	2.24796
VQM (95% Confidence Level)	0.03462	0.03512	0.03952	0.03279	0.04026	0.03463
Lower limit	7.78558	8.11351	8.41827	7.64725	8.9058	8.10716
Upper limit	7.85482	8.18376	8.49732	7.71283	8.98631	8.17643
VQM gain/loss	0	0.32843	0.63759	0	1.26601	0.46175

❖ The 100 Mbps Network Simulation (The Random Error)

	Random		
	5 fps	15 fps	30 fps
PSNR (AVG)	20.3734	19.4515	19.5048
PSNR (MIN)	14.0827	13.574	12.1795
PSNR (MAX)	39.0329	38.738	30.8975
PSNR (SD)	4.27086	3.56611	3.05753
PSNR (95% Confidence Level)	0.0658	0.05494	0.04711
Lower limit	20.3076	19.3965	19.4577
Upper limit	20.4392	19.5064	19.5519
PSNR gain/loss	0	0.92192	0.86861
SSIM (AVG)	0.84414	0.82771	0.81602
SSIM (MIN)	0.64617	0.63231	0.5923
SSIM (MAX)	0.98259	0.9855	0.97143
SSIM (SD)	0.07688	0.08356	0.09343
SSIM (95% Confidence Level)	0.00118	0.00129	0.00144
Lower limit	0.84296	0.82642	0.81458
Upper limit	0.84533	0.82899	0.81746
SSIM gain/loss	0	0.01644	0.02812
VQM (AVG)	8.04533	8.85716	8.59569
VQM (MIN)	0.74901	0.79491	2.54279
VQM (MAX)	12.9777	13.0404	14.8242
VQM (SD)	2.37295	2.49662	2.29516
VQM (95% Confidence Level)	0.03656	0.03846	0.03536
Lower limit	8.00877	8.81869	8.56033
Upper limit	8.08189	8.89562	8.63105
VQM gain/loss	0	0.81183	0.55036

Appendix C: The Sample Raw Data in the PPM Format

P3

rgb2.ppm

1280 720

255

48 52 48 48 52 48 48 52 48

248 248 252 248 248 252 248 248 252 248 248 252 248 248 252 248 248 252 248 248 252 248 248 252 248 248 252 248 248 252

240 248 244 240 248 244 240 248 244 240 248 244 240 248 244 240 248 244 240 248 244 240 248 244 240 248 244 240 248 244