İSTANBUL KÜLTÜR UNIVERSITY
INSTITUTE OF SCIENCES

NETWORK CODING FOR MULTI-STATE VIDEO TRANSMISSION

Master of Science Thesis by

Öznur ŞENGEL

Department : Computer Engineering

Program : Computer Engineering

Supervisor : Assis.Prof.Dr. Sıla EKMEKÇİ FLIERL

SEPTEMBER 2013

İSTANBUL KÜLTÜR UNIVERSITY
INSTITUTE OF SCIENCES

NETWORK CODING FOR MULTI-STATE VIDEO TRANSMISSION

M.Sc. Thesis by

Öznur ŞENGEL

1009051004

Date of submission : 17 September 2013
Date of defence examination : 23 September 2013

Supervisor and Chairperson : Assis.Prof.Dr. Sıla EKMEKÇİ FLIERL
Members of Examining Committee : Assis.Prof.Dr. Akhan AKBULUT
Assis.Prof.Dr. Ertuğrul SAATÇİ

SEPTEMBER 2013

CONTENT

ABBREVIATIONS

FEC	: Forward Error Correction
LC	: Layered Coding
MDC	: Multiple Description Coding
MSVC	: Multi-State Video Coding
TCP	: Transmission Control Protocol
XOR-ed	: eXclusive OR-ed
CD	: Compact Disc
MPEG	: Moving Pictures Experts Group
AVC	: Advanced Video Coding
GPL	: General Public License
DSL	: Digital Subscriber Line
DVD	: Digital Versatile Disc
ISDN	: Integrated Services Digital Network
LAN	: Local Area Network
MMS	: Multimedia Messaging Services
VBSMC	: Variable Block-Size Motion Compensation
HD	: High Definition
CABAC	: Context-Adaptive Binary Arithmetic Coding
CAVLC	: Context-Adaptive Binary-Length coding
VCL	: Variable Length Coding
DCT	: Discrete Cosine Transform
HTML5	: Hyper Text Markup Language 5
CPU	: Central Processing Unit
DVB	: Digital Video Broadcasting
ASF	: Advanced Systems Format
wma	: Windows Media Audio
wmv	: Windows Media Video
UDP	: User Datagram Protocol
WAN	: Wide Area Network
DNS	: Domain Name System
TFTP	: Trivial File Transfer Protocol
SNMP	: Simple Network Management Protocol

IP	: Internet Protocol
DHCP	: Dynamic Host Configuration Protocol
IPTV	: Internet Protocol Television
IPv4	: Internet Protocol Version 4
IPv6	: Internet Protocol Version 6
RTP	: Real-Time Transport Protocol
RTCP	: Real-Time Transport Control Protocol
Qos	: Quality of services
PT	: Payload Type
HTTP	: Hypertext Transfer Protocol
HTTPS	: Secure Hypertext Transfer Protocol
POP3	: Post Office Protocol 3
FTP	: File Transfer Protocol
ACK	: Acknowledge
SYN	: Synchronize
NS	: Nonce Sum
ECN	: Electronic Communication Network
CWR	: Congestion Window Reduced
URG	: Urgent Pointer
PSH	: Push Function
RST	: Reset the Connection
FIN	: Final
TV	: Television
ARQ	: Automatic Repeat request
MD	: Multiple Description
FIFO	: First In First Out
Snr	: Signal-to-noise ratio

TABLE LIST

FIGURE LIST

SYMBOL LIST

t-1, t	:	Time
Computer A, Computer B, …	:	The name of computer
Node A, Node B, Node C, …	:	The node of the network
Subsequence1, Subsequence2, …	:	Subsquences of the video
Frame1, Frame2, ..., FrameK	:	The frame of the video
y, y_a, y_b, …	:	Vertical position of the pixels
x, x_a, x_b, …	:	Horizontal position of the pixels
Path #1, Path #2, …	:	Transmission channels
Description #1, Description #2, …	:	Subsquences of the video
T_x	:	Transmission (output) queue
R_x	:	Receiving queue
n_1, n_2, ..., n_N	:	Network nodes
ϕ_n	:	Set of packets in Tx queue of the intermediate node
p_1,p_2, ..., p_{ϕ_n}	:	Packets
$t(p_i)$	:	Primary packets of the target node
$\Psi_{t(p_i)}$	:	Set of the overhead packets
$v_1, \dots, v_{\Psi_{t(p_i)}}$	:	Overhead packets
c_k^i	:	Candidate codes
$S_k^{t(p_i)}$	:	The k^{th} subset of overhead packets
P_i	:	Primary packet

$I_k^i(n_\eta)$	:	The improvement video quality
d_l^k, g_l^k, and $d(\pi_n(j))$	:	Indicator functions
$\Delta(l)$	:	snr
$\gamma(l)$	:	Priority of the packet
$P(l)$	:	Probability of the loss packet l
A1, A2, …, B1, B2, …, C1, C2, …	:	Packets
c_1, c_2, c_3, ...	:	Candidate network packets (codes)
Input A, Input B	:	Input values of XOR gate
Output A⊕B	:	Output value of XOR gate

Enstitüsü	**:**	**Fen Bilimleri**
Anabilim Dalı	**:**	**Bilgisayar Mühendisliği**
Programı	**:**	**Bilgisayar Mühendisliği**
Tez Danışmanı	**:**	**Yrd.Doç.Dr. Sıla EKMEKÇİ FLIERL**
Tez Türü ve Tarihi	**:**	**Yükseklisans - Eylül 2013**

KISA ÖZET

NETWORK CODING FOR MULTI-STATE VIDEO TRANSMISSION

Öznur ŞENGEL

Bu çalışmanın konusu dayanıklı video paketlerinin gönderiminde kullanılan Multi-State Video Coding (MSVC) Tekniği ile Ağ Kodlama tekniklerini kullanarak daha fazla ağ yayılımı ve video kalitesi ile paketleri tüm ağdaki düğümlere göndermektir. Böylece düğümler kendisine ait olan paketlere daha hızlı bir şekilde erişebilecektir.

Sistemin iki ana kısmı bulunmaktadır 1) Multi-State Video Coding ve 2) Ağ Kodlama. Çalışmanın amacı hem video kalitesini hem de ağ yayılımını arttırmaktır. Multi-State Video Coding tekniğini dayanıklılığı sağlamak ve Ağ Kodlamayı ise ağ üzerindeki yayılımı artırmak için kullanıyoruz. Bu mantıkla öncelikle var olan video paketlerini MSVC tekniği ile iki ayrı alt dizine ayırıp ağ kodlama ile paketlerin gönderimini sağlıyoruz. Böylece, ağ üzerinde paketlerin dayanıklılığı gibi yayılımını ve videonun kalitesinide artırmayı amaçlıyoruz.

Multi-State Video Coding (MSVC) bir video streaming tekniği olan Multiple Description Coding temeline dayanan Video Kodlama Teknolojisidir. MSVC ile video iki veya daha fazla altdizine ayrılmaktadır. Biz çalışmamızda videoyu iki altdizine ayırıyoruz: birinci altdizinde sadece çift numaralı çerçeveler, ikinci altdizinde tek numaralı çerçeveler bulunmaktadır. Oluşan bu altdizinler kodlanarak paketler haline getirilmektedir. Herbir paket sırasıyla ağda farklı kanallar üzerinden alıcılara gönderilmektedir. Eğer alıcı hem tek hem de çift numaralı çerçeveleri almış

ise alıcı çözümlemeyi yapıp çerçeveleri oluşturur. Eger çerçevelerden biri kayıp ise, alıcı kayıp çerçeveyi farklı altdizinlerde bulunan bir önceki ve sonraki çerçeveler ile MSVC tekniğinin durum iyileştirme yapısını kullanarak oluşturabilmektedir.

MSVC tekniği ile oluşan altdizinlerdeki paketlerin gönderimi için ağ kodlama yapılmaktadır. Çalışmamızda iki farklı ağ kodlama algoritması ile paketlerin gönderimi yapılmaktadır. İlk algoritma "Network Coding for Video (NCV)" daha iyi video kalitesi ile videoların alıcıya ulaşmasını sağlamaktadır. Bunun için altdizinlerdeki ilk aktif paketi alıp belirli kriterler ile diğer paketleri birleştirip aday paketleri oluşturmaktadır. Daha sonra en iyi ve yüksek kaliteyi sağlayan paketi ağdaki tüm nodelara göndermektedir. İkinci algoritma "Network Coding for Video: looking into the queue in depth (NCVD)" NCV'nin yaptığı gibi paketleri oluşturmaktadır, bu sefer ilk aktif paket ile oluşanlar dışında, kuyruktaki diğer aktif paketler ile de aday paketleri oluşturmaktadır. Oluşan tüm aday kodlar arasından en iyi paketi seçmektedir.

Ağ kodlama algoritmaları ile gönderilen paketler ağdaki tüm düğümlere ulaşmaktadır. Ulaşan paketler alıcı düğümler tarafından çözümlenmektedir. Çözümlenen paketler içinde alıcıya ait paketler var ise gönderilen paket yerine ulaşmış demektir. Ana paketin dışında alıcıya ulaşan kendisine ait olmayan başka paketler var ise alıcı bu paketleri de saklamaktadır. Ağ kodlama esnasında ara düğüm kendi ağındaki düğümler ile sürekli bilgi alışverişi içinde olduğundan ara düğüm paket gönderimi yapmadan önce ağdaki tüm düğümlerdeki bu paketlerin bilgisini almaktadır. Ara düğüm bu paketleri aday paketlerden en yüksek verimi sağlayanı bulmak için kullanmaktadır. Ara düğüm gönderilecek paketi en fazla düğüm tarafından çözümlenerek kendi çerçevesini elde etmesini sağlayacak şekilde belirlemektedir.

Bu kapsam için geliştirdiğimiz sistemi farklı ağ senaryolarında denemelerini gerçekleştirdik. Öncelikle, sistemin çalışacağı ağı oluşturduk. Ağı ara düğüm ve ara düğüm ile etkileşim içinde olan birden fazla düğümden oluşturduk. Birbirleri ile haberleşmelerini sağlamak için ara düğüm ağdaki tüm düğümleri dinlemeye başlıyor. Düğümler kendilerine ait videonun transferinin başlamasından önce ara düğüme bir önceki transfere ait buffer bilgilerini göndermektedir. Buffer bilgilerini alan ara düğüm ilgili algoritmanın yapısına göre paketleri oluşturup bilgi alışverişi içinde olduğu tüm düğümlere göndermektedir.

Oluşturduğumuz farklı senaryolarda, ara düğümün gönderdiği ilk aktif paketin her iki algoritmada da (1) Network Coding for Video (NCV), (2) Network Coding for Video: looking into the queue in depth (NCVD) alıcıya başarılı bir şekilde ulaştığını gördük. Ağda ara düğüm ile düğüm arasında haberleşmenin kesildiği ve bir önceki paketin alımı bitmeden diğer paketin gönderiminden kaynaklanan paket kayıplarında alıcı düğüm kendisine gönderilen çerçeveye ulaşamadığı durumlar ile karşılaştık. Böyle çerçevenin ulaşmadığı durumlarda MSVC tekniğinin durum iyileştirme özelliği ile kayıp frame oluşturulmakta ve videoda oluşabilecek kesinti alıcıya yansımamaktır.

Anahtar Sözcükler: Multi-State Video Coding (MSVC), Network Coding, Network Coding for Video (NCV), Network Coding for Video: looking into the queue in depth (NCVD), Video Streaming, Video Transmission

Institute	**:**	**Institute of Sciences**
Department	**:**	**Computer Engineering**
Program	**:**	**Computer Engineering**
Supervisor	**:**	**Assis.Prof.Dr. Sıla EKMEKÇİ FLIERL**
Degree Awarded and Date	**:**	**M.Sc. – September 2013**

ABSTRACT

NETWORK CODING FOR MULTI-STATE VIDEO TRANSMISSION

Öznur ŞENGEL

The goal of this work is to send video packets to all nodes in the network by enveloping Multi-State Video Coding (MSVC) at the same time network coding to maximize the throughput and video quality.

This work has two main parts: 1) Multi-State Video Coding and 2) Network Coding. The main purpose of this work is to maximize not only the video quality but also the network throughput. We will use Multi-State Video Coding to achieve robustness and we will use network coding to increase throughput over the network. After generating the two subsequences using MSVC, we apply network coding to support transmission of packets. In this manner, we aim to increase the throughput as well as robustness and quality of the video transmission.

Multi-State Video Coding (MSVC) is based on Multiple Description Coding which is a kind of video coding technique. Video is split into two or more subsequences with MSVC. In our work, we have two subsequences; the first subsequence includes even numbered frames, the second subsequence includes odd numbered frames. These subsequences are encoded and distributed into packets. Each packet is transferred using the same channel or different channels to the receiver. If the receiver takes both even and odd frames correctly, it decodes packet to create frames. If any frame is lost, the receiver may reconstruct lost frame from previous and next

frames of the different subsequences. This situation is referred to as state recovery of MSVC.

We use network coding to send each packet in substreams generated via MSVC. In our work, we apply two network coding algorithms to send packets. The first algorithm is “Network Coding for Video (NCV)”. It provides delivered video to recipient with best and high video quality. For this, it chooses the first active packet from the queue and combine active packet with other packet to create candidate network code according to some criteria. Before encoding, algorithm calculates the improvement for each candidate network code. Then, the maximum improvement packet is delivered to all nodes over the network. The second algorithm is “Network Coding for Video: looking into the queue in depth (NCVD)” is similar to NCV in creating packets. But it not only looks the first active packet of the queue but also looks the other active packet in the queue to create candidate code. NCVD chooses the video packet which is satisfies the high video quality.

Packets are delivered to all nodes in the network via network coding algorithm. Received packets are decoded by target nodes. If there is any packet for that receiver in decoded packet, the primary packet is received successfully. If there are other packets which are not belongs to receiver, the receiver also stores this packet. Before sending any packets, the intermediate node takes this packets information from all nodes in network through the intermediate node is always in information exchange via all nodes in network. The intermediate node uses this packet to find which candidate code provides the highest video quality. The intermediate node determines the packet that decoded more node to obtain own frame.

We tested our system in different network scenarios. First of all, we create the network that the system will work on it. There are intermediate node and more than one node that interact with intermediate node in network. Intermediate node starts listening to ensure all nodes in the network communicate with each other. Nodes send frames information of the previous transfer to intermediate node before starting video transfer. After taking all buffer information of the nodes, intermediate node sends packet that is generated according to the structure of the algorithm to all nodes on network.

Created in different scenarios, the first active packet that is sending by intermediate node, is received successfully in both algorithm (1) Network Coding for Video (NCV), (2) Network Coding for Video: looking into the queue in depth (NCVD). We encountered with the communication cuts off between intermediate node and other node and the receiver does not reach the own frame because the packet is sent before the end the previous packet. In such case frame is not received to target, the lost frame generated with state recovery of MSVC technique and the interruption is not recognized by receiver.

KeyWords: Multi-State Video Coding (MSVC), Network Coding, Network Coding for Video (NCV), Network Coding for Video: looking into the queue in depth (NCVD), Video Streaming, Video Transmission

1. INTRODUCTION

Nowadays, there are different types of communication. People can communicate each other via different kind of communication devices such as mobile phone, computer, tablet etc. Social networking over internet is the most popular communication platform. Users can share all information about their interesting on their profile, chat with their friends, do video conferences, and find their friends and relatives.

While the internet is more important, the communication quality must be perfect. There are a lot of conditions to improve communication quality such as wired/wireless internet quality, hardware quality of computer, modem quality to interrupt the communication over the internet. But this hardware quality is not enough to communicate over the internet.

Think about a web page which includes pictures, animations, and video files. The pictures are the most important part which showed in a picture gallery. When you enter this web page to see some pictures, you have the latest technology modem and computer. Also your wired/wireless connection is perfect and the communication with base station is strong. Although these perfect hardware quality equipment, you can see some artifacts when you display pictures. The reason is that the encoder/decoder software of the picture. When the picture is encoded, some bits are loss so sometimes we see blurred some part of the picture. As we see that, the high hardware quality just affect the reliability, throughput and speed download of the web page. If there is software to encode the picture, the hardware quality cannot improve the picture quality when it is decoded.

New technology devices also supports communication on video from phone, computer etc. In addition to communication, there are a lot of video streaming web pages to satisfy different king of video to their users. Video streaming compresses content of the video to transmit from one node to another. When the video content is received, content decoded and displayed. The most important feature for video transmission is to send video content without any lost. If any frame is lost the video quality is decrease. There are a lot of methods such as Forward Error Correction

(FEC), Layered Coding (LC), Multiple Description Coding (MDC), Robust Source Coding, etc. for video streaming over the internet to improve video quality. All of these methods have different tricks to improve video quality.

Multi-State Video Coding (MSVC) (Apostolopoulos, 2001) is a Multiple Description scheme in which the video is split into two subsequences. One of these subsequences includes only odd numbered frames the other one the even numbered frames. Each subsequence is encoded into separate bitstreams. The bitstreams from the even and odd frames are divided into packets and the sequence of packets for each subsequence is transmitted each over a different channel to the receiver. The receiver decodes the subsequences. If both the even and odd streams are received correctly, they are decoded to produce the even and odd frames. If one of the frames is lost, the other frames may still be decoded and displayed. In this situation, the lost frames are reconstructed by using the past and future frames from the other subsequences. This property of MSVC is referred to as state recovery.

After this step, we have multiple packet streams to send to the destination address (receiver). Computer drops packets on the network with destination IP (Internet Protocol) address and the packet are delivered to that destination address. The sending and receiving computers have no control over how the packets get from sender to receiver. If we can control this, we increase the throughput. To do this, we consider applying network coding. There are some intermediate nodes on the network. These nodes can perform simple network coding operation and combine packets from several incoming streams into a single outgoing packets. This packet is broadcasted to the entire neighborhood, thus reaching several nodes at the same time (Seferoglu & Markopoulou, 2009). There are some network coding algorithms for video based on this idea. The first algorithm, “Network Coding for Video (NCV)”, achieves the same throughput gains as in (S. Katti et al., 2006) but also intelligently chooses the network codes that maximize video quality. The second algorithm, “Network Coding for Video: looking into the queue in depth (NCVD)”, uses NCV as a building block but considers more coding options thus further improving video quality and throughput.

The main idea of the NCV algorithm is to select the best network code to improve video quality. Each client has virtual buffer and intermediate node maintains a queue that stores packets. We select the first active packet as primary packet from the queue. According to this information, we generate network codes which are candidate codes. Then, we choose the best code according to video quality improvement.

NCVD algorithm is looking into the queue in depth. The second algorithm improves over NCV by also optimizing the selection of the primary packet. We can select not only the head of line packet (e.a NCV) but also one of the active packets on queue as the primary packet. According to this primary packet, we generate candidate codes and then choose the best one to improve both throughput and video quality.

The main purpose of this work is to maximize not only the video quality but also the network throughput. We use Multi-State Video Coding to achieve robustness and we use Network Coding to increase throughput over a broadcast. After generating the two descriptions using MSVC, we will apply Network Coding. In this manner, we aim to increase the throughput as well as robustness and quality of the video transmission.

1.1. PROBLEM STATEMENT

The aims of this thesis improve the video quality when video frames transmit from destination to target. To achieve this aim, we use MSVC to split video in two subsequences first one includes even frames the other includes odd frames. Then the system applies network algorithm before sending packet. According to network algorithm, system prepares a XOR-ed packet. Packet transmits with TCP protocol to all nodes on network. Each node is decoded packet to get their packets. If the main packet delivered the target node correctly, our aim is achieved.

2. RELEATED WORK

In our work, we deal with two main ideas: video streaming technology that is Multi-State Video Coding and Network Coding. There are a lot of works in each topic separately.

Basically, MSVC based on Multiple Description Coding (MD) schema. MSVC generates more than two descriptions either balanced or unbalanced (Apostolopoulos & Wee, 2001). Unbalanced achieves to adapt quantization (Flierl, Sikora, & Frossard, 2006), spatial resolution or frame rate. MSVC is improved by applying multiple state recovery (Apostolopoulos, 2001) and error concealment methods on a MD basis (Liao & Gibson, 2008; Liao & Gibson, 2009) uses MSVC with refined error concealment method and it also compare four different approaches SDC_ROPE, MSVC, EROPE, MSVC_OMS that is the optimal mode selection approach for MSVC.

Network coding part has different work either with video packet or with different communication packets. General principle of the network coding to receive any kind of packet to target with minimum lost. There are different works on not only wired network coding but also wireless network coding. Media streaming especially video streaming becomes more popular and authors start to deal with relationship between network coding and media streaming such as in paper (Thomos & Frossard, 2009).

The most important part is transmission that can be either over single path or multiple paths. (Wang, Reibman, & Lin, 2005) deals with various MD coding techniques on quantization transform coding, also it compare multiple path transmission with single path transmission on MDC. (Ekmekci & Sikora, 2003) paper compares MSVC with Single Layer Coding (SDC) at different channel condition and coding option to show that the motion vector are always received. A lot of different network coding strategies and algorithms are developed to transmit video (Montpetit & Medard, 2010). This algorithm combined either with MD (Ramasubramonian & Woods, 2010) or with MSVC. Path diversity has more important place to send packet from different channels to improve reliable video communication over lossy packet networks (Apostolopoulos, 2001).

The aim of the using network coding is to improve throughput. When two clients (ClientA and ClientB) exchange the couple of packet, ClientA sends its packet to router which forwards it to ClientB and ClientB sends its packet to router which forwards it to ClientA. In (S. Katti et al., 2005; S. Katti et al., 2006) ClientA and ClientB sends packet to router which XORs and broadcasts resulting XOR-ed packet as a result of this scenario all process takes three transmission instead of four transmission. This architecture is called COPE (S. Katti et al., 2008; Katabi et al., 2006). In (Fragouli et al., 2007) discuss the opportunities and challenges how network coding improve throughput and reliability of wireless network.

3. VIDEO

Nowadays, the most popular communication way is Internet. The number of users shares all information about themselves over the internet. When the internet is most common over the world, the data must be transmitting safely without losing any off them. A lot of people share their images and videos via internet. When another people want to see this images or videos, the received picture is important. When we open any picture, sometimes we are waiting to download all images. This speed affects to see the image immediately. Also this happens while watching video from internet. We have to wait to see the next frame while watching. This duration can be littler so nobody can recognize that when they are watching video.

Technology especially internet technology is changing and the user expect more quality than previous from the internet. Formerly, images and text are placed on their web pages but now they want to see video. They want to see videos fast and with high quality as well as on their television. There are a lot of issues such as bandwidth, users are away to transmit high quality video over the internet. Video streaming are becoming more and more popular by the day so the users is rapidly expanding bandwidth services. User can broadcast lectures, deliver seminars, make announcements, or show exactly how something is supposed to work with video streaming.

When we communicate over the internet, heterogeneity and congestion cause three main problems unpredictable throughput, losses and delays. We can solve these problems with providing quality, reliability, and interactivity. We can ensure quality with low bitrates, reliability with independent of loss pattern and interactivity with low latency when many users watch same video.

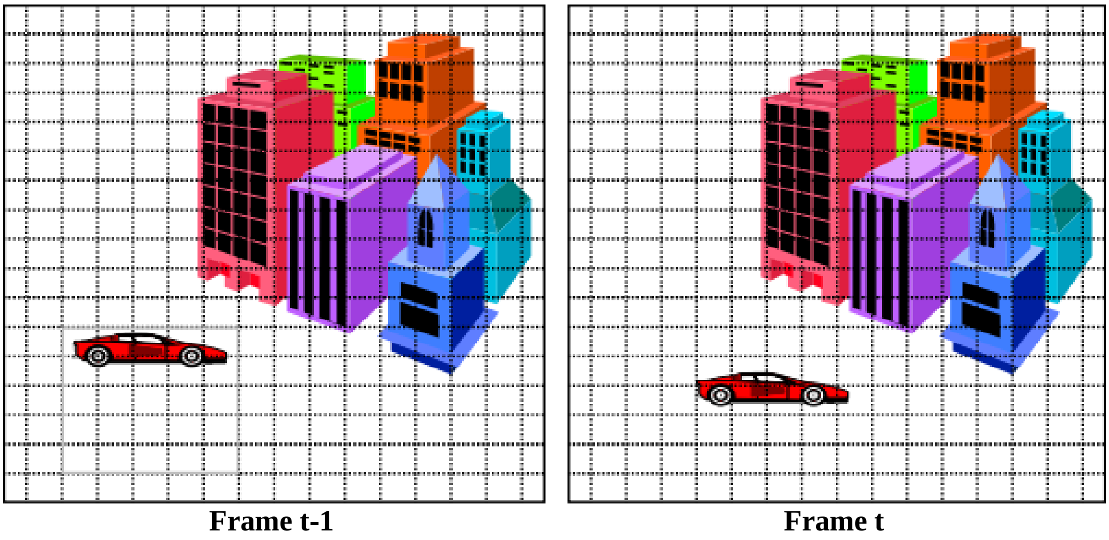


Figure 3.1 Adjacent frames from any video.

A video occur from animation and voices. These animation and voices can change in each time on video. A picture in any time on video is called frame. Video has more than one adjacent frame. In each time, the video frames either can be change or not. The only difference between one frame and another is the result of either the camera moving or an object in the frame moving. In figure 3.1, there is a motion the car is driven, on the other hand the buildings are in the same place in each frame. So, we can split each video into frames according to time.

4. VIDEO CODEC, BITSTREAM, TRANSPORT

4.1. VIDEO CODECS

Codec is compression algorithm software that is used to compress or decompress a digital media file such as video to reduce the size of a stream. The compression is lossy because when the video is coded take some part from the original file and discarding other parts of the video to reduce the size of the video. The codec decides what data take part in compressed version, what are not with translator. Each codec does not use the same way to translate the video file so we must use the same codec either encode or decode. We must choose the best codec for what we are trying to do in order to maintain the best ratio of the file size to quality.

The codec have two main parts encode and decode. To compress the video encoder performs the encoding function to decompress the video decoder performs the decoding function to get video frames. There are different video codecs; some of them include both encode and decode part, some codecs only include whether encode or decode part. For instance, when we open any video file from CD, player use the video codec to decompress the file so the video can be played.

There are different video codecs which are use different file format. We will compare two codecs that are H.264 and VP8.

4.1.1. H.264

H.264 is known as MPEG-4 Part 10 or AVC (Advanced Video Coding) that is lossless video codec. Encoder part of it uses x264 which is only encoder and a GPL licensed implementation of the H.264 video standard, decoder part uses FFmpeg decoder. It is most commonly used and known video compression standard, and also it is used by streaming internet sources such as videos from Vimeo and Youtube.

The main function of this standard is high compression and quality for broadcasting. This standard gives answer to solve technical solution in the following application areas

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

• Interactive or serial storage on optical and magnetic devices, DVD, etc.

• Conversational services over ISDN, Ethernet, LAN, DSL, wireless and mobile networks, modems, etc.

• Video-on-demand or multimedia streaming services over ISDN, cable modem, DSL, LAN, wireless networking, etc.

• Multimedia messaging services (MMS) over ISDN, DSL, Ethernet, LAN, wireless and mobile networks, etc.(Wiegand et al., 2003).

The features of H.264 (Robertson; Richardson)

• It provides good video quality with lower bitrate than previous standards.

• It allows using in wide area applications on variety of networks and systems.

• It is using multi-picture inter-picture prediction that provides improvement in bit rate and quality in scene.

• It supports variable block-size motion compensation (VBSMC) with 16x16 large block size and 4x4 small block size.

• This standard enables quarter-sample-accurate motion compensation.

• It uses multiple reference pictures for motion compensation that means more than one previous picture to predict the values in an incoming picture.

• Encoder weighted the motion compensated prediction signal to improve coding efficiency for scene containing fades.

• Dependency between the ordering of pictures for motion compensation and ordering picture for display is largely removed with this standard.

• It support sample depth precision ranging from 8 to 14 bits per sample.

• It is more convenient for video network delivery and delivery of HD, high definition video.

- Entropy coding part includes context-adaptive binary arithmetic coding (CABAC), Context-adaptive binary-length coding (CAVLC) and variable length coding (VCL).
- It use in loop deblocking filter which support to prevent blocking artifacts.
- It uses discrete cosine transform (DCT) and Hadamard transform.
- It is available for everyone to implement.

4.1.2. VP8

VP8, which is developed by Google, is kind of video compression format. It has software library capable of encoding video streams that is libvpx. Libvpx is not only encoder but also decoder because it is capable of decoding VP8 video streams. There are another decoder is ffvp8 decoder is faster than libvpx decoder (Garrett-Glaser, 23/07/2010).

The features of VP8

- Compiled with VP8 library
- It includes most color space conversions supported by Xvid codec
- Uses several threads on multi-core processors.
- It encode file with libvpx and decode file not only with libvpx but also ffvp8
- Firefox, Chrome, Opera and Adobe support VP8.
- Solves the HTML5 video problem.

Comparison H.264 with VP8 (Ozer, 2010)

Both of them are free. When we compare the feature on implementation H.264 has advantages than VP8. So H.264 not only is more efficient but also has slight quality advantage. H.264 is supported by iDevices (with CPU), Blackberry, Palm, Android (with CPU acceleration). Compression style of H.264 is both lossy and lossless; compression style of VP8 is only lossy. In conclusion, VP8 is a great codec but H.264 has superior integration in streaming and device world.

4.2. BITSTREAM

It is sequence of bits and set of headers allow simple access to binary structures such as MPREG, DVB, IEFT, etc.

4.2.1. MP4

MP4 ("MPEG-4 Part 14," 2004) is known as MPEG-4 which is digital multimedia format to store video and audio. This format allows streaming over the Internet like other format. The file extension is .mp4.

4.2.2. WebM

WebM is an open media file format designed for the web and it file consists VP8 video codec and file compressed Vorbis audio streams ("About WebM,"). Youtube, Wikimedia, Skype uses WebM for HTML5 player. It has simple container format and minimal encode with sub-option and satisfy highest quality real-time video transmit. The file extension is .webm.

4.2.3. ASF

ASF ("Advanced Systems Format,") is part of the Windows Media framework so it is Microsoft’s digital video container format for streaming media. .asf, .wma, .wmv are the file extension of ASF. It determines the structure of the video / audio stream instead of how the video or audio should be encoded.

4.3. TRANSPORT WITH A NETWORK PROTOCOL

The most important part of the video streaming is transport encoded video file to target node or all neighbor of sender node. After encoding the video stream, we need to transport this stream with any network protocol. According to your application, you can choose any protocol such as UDP, RTP, and TCP.

4.3.1. User Datagram Protocol (UDP)

UDP is unreliable and connectionless communication protocol that is used for transport of data across an internet protocol based network. UDP does not need to connection when send data. It uses to transmit real time data transport like voice and video in wide area network (WAN). The time of data transfer is little because it is not deal with connection, retransmit data and flow control. DNS, TFTP and SNMP use UDP that does not need to more bandwidth.

UDP is not reliable protocol because it sent packet in network but it does not control the packet is received or not to target node and it have not approve the packet is received or not. Also it does not check the reliability of the packet and take all packets to its system so it is not convenient to security of the system. If the time is more important for your system you can use UDP because it is not waiting acknowledge from target. The quality of real-time data transfer of UDP is less than TCP because UDP and TCP use the same network path and TCP has more data transfer than UDP.

According your application, you can choose UDP to transmit packet because some properties of the UDP. In table1, there are some features of the UDP and in which situation it used for transmission.

Table 4.1 UDP's attributes and suitable application

Attribute	**Suitable for**	**Such as**
Transaction oriented	Simple query response protocols	Domain Name System or Network Time Protocol
Datagrams	Modeling other protocols	IP tunneling or Remote Procedure Call and Network File System
Simple	Bootstrapping or other purposes without a full protocol stack	DHCP and Trivial File Transfer Protocol
Stateless	Very large numbers of client	Streaming media application for example IPTV
Lack of retransmission delays	Real time application	Voice over IP, online games and Real Time Streaming Protocol
Unidirectional communication	Broadcast information and shared information	Broadcast time or Routing Information Protocol

Packet Structure

In figure 4.1 illustrate packet structure of the UDP. UDP header has four parts; source port, destination port, length, and checksum, size of them are equal 16 bit. Source port is not used for each transfer, when the packet is received to destination port, sometimes it need to reply the packet is received or not according to application. Source port is optional both IPv4 and IPv6, checksum is optional in IPv4 only. Destination port include where data packet will be transmitted. Length part stores the length of the entire datagram which includes header and data. Checksum field is for checking any error of the header and data. If there is no any checksum, the field value fills with zeros. There are difference between IPv4 and IPv6 calculation of the checksum. UDP has checksum for data integrity and port numbers for transmit datagram from source to destination. Data field contains the actual data.

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Length																Checksum															

Figure 4.1 UDP Header

4.3.2. Real-Time Transport Protocol (RTP)

RTP is widely used in communication and entertainment systems and these systems involve streaming media such as television services web-based push-to-talk, telephony and video conference applications. RTP is send real-time audio, video or simulation packets from source to destination over IP networks. It provides end to end delivery services for data with real time properties (H. Schulzrinne, Casner, Frederick, & Jacobson, January 1996).

RTP has connection between source and destination during data transfer. Machine starts to communicate with this connection. For each different media type machine has different connection. RTP sets up for each stream.

RTP when is transferring media stream such as audio and video usually uses with RTP Control Protocol (RTCP) that is used to follow transmission statistic and quality of services (Qos) information. RTP is unidirectional, RTCP is bidirectional. RTP framing if needed identify synchronization source, transfer media data (unreliable), demultiplexing (combined audio and video), synchronization and sequencing support, next layer (e.g. media) identification (H. Schulzrinne, December 1992). RTCP identify participants, describe content, quality of services information, request for retransmission (H. Schulzrinne, December 1992).

Packet Structure

Figure 4.2 shows the RTP packet header information. Version stores 2 bits information about the version of the protocol. P represents padding which determines if there are extra padding bytes at the end of the RTP packet. X is the extension header between standard header and payload data. CC is the short term of the CSRC Count which is contains number of CSRC identifiers. M is marker of the application level. PT is payload type that determines the format of payload and its interpretation. Sequence number store incremental value when the packet is received to target, it is important when the packet is loss and restore packet sequence. SSRC is synchronization source which identify the source of a stream. CSRC is contributing sources which ones generated from multiple sources. Extension header part is optional, first part of it store profile specific identifier, second part is extension header length.

bit offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	Version		P	X	CC				M	PT							Sequence Number															
32	Timestamp																															
64	SSRC identifier																															
96	CSRC identifier																															
96+32xCC	Profile-specific extension header ID																								Extension header length							
128+32xCC	Extension header																															

Figure 4.2 RTP header

4.3.3. Transmission Control Protocol (TCP)

TCP is a reliable, error checked delivery of stream, connection-based, and host-to-host protocol. Most of the data transfer on the internet makes over TCP because it ensures to deliver the data to target node. HTTP, HTTPS, POP3, STM and FTP are the most popular protocols transfer the data via TCP. The most popular protocol uses TCP because of the characteristic features of the TCP;

Stream Data Transfer: TCP gives a sequence number to an unstructured stream of bytes. TCP must be packed the byte streams to packet to satisfy the communication between sender and receiver. Then TCP sends this packet to IP layer to transfer to the destination machine. Application does not have divide data to blocks because TCP decides how to segment the data and forward due to own convenience of the data.

Reliability: TCP gives a sequence number for each byte of data. When the TCP sends any data, it follows the order of the data from its sequence number. After sending data, TCP waits for acknowledge from destination port until timeout. When the destination host sends ACK with sequence number, sender understands this data is received by destination. If acknowledge did not come, sender transmits data again until it takes ACK. According to this ACK, TCP does not send any segment again if it is received. So, TCP is reliable and eliminates duplicate segments.

Flow Control: Sometimes sender sends more fast data to receiver. If the receiver has not enough places to take this data, it stores data to the buffer. After some times, memory of the buffer will be full and the receiver does not take data any more. Some packet cannot reach to receiver so the sender will send these packets again and again. This situation will affect the performance of the communication. To avoid from this situation, TCP uses flow control. If the received computer took more data than its capacity of the data received and it has not enough memory buffers to store this data, it will send “Not ready” alert message to sender with flow control mechanism. Sender stops sending data until received computer sends “Ready” message.

Multiplexing: TCP uses the IP addresses or port number to communicate each other over Internets. Socket consists network and host addresses and uses for each TCP connection. A socket may be simultaneously used in multiple connections (Information Sciences Institute University of Southern California, September 1981).

Connections: Connection contains sockets, sequence numbers and window sizes information. Two hosts must be established connection before starting communication. The connection is terminated or closed after communication is finished.

Precedence and Security: Each machine may specify precedence and security of their communication. According the importance of the stream, TCP sent data streams.

Full duplex: TCP can send and receive data streams simultaneously.

TCP is connection oriented services so to transfer data among network first must be open a connection on host, then host can send any data to another until connection is open. TCP working flow chart illustrated in figure 4.4 in detail. TCP is operated in three main parts; (i) start connection, (ii) data transfer, (iii) finish connection. Before start data transfer, connection must be established in handshaking process. After all data is transferred, the connection is closed and all allocated resources are released.

Table 4.2 Terminology of TCP state diagram in figure 4.4.

STATE	ACTION
CLOSED	Both server and client have no connection.
LISTEN	Server waiting for a connection request
SYN-RCVD	Server waiting for a confirming connection request
SYN-SENT	Client sent connection request to server
ESTABLISHED	The connection is open both server and client to data transmission.
FIN-WAIT-1	Both server and client waiting for connection termination request.
FIN-WAIT-2	Both server and client waiting for connection termination request.
TIME-WAIT	Either server or client waiting for enough time to transfer the packet to destination and acknowledge.
CLOSING	Both server and client waiting for connection termination request acknowledge from the remote TCP.
CLOSE-WAIT	Both server and client waiting for a connection termination request from the local user.
LAST-ACK	Both server and client waiting for an acknowledge of the connection termination request previously sent to the remote TCP.

Before starting the data transfer connection must be between two computers. Assume that computer A wants to send data to computer B, first computer A sends TCP-SYN message to computer B. When the message received to computer B, computer B sends acknowledge message TCP SYN+ACK to computer A. Then computer A sends acknowledge (ACK) message to computer B. Finally, computer B takes acknowledge (ACK) message which is means TCP connection is established. The TCP connection is setting up between to computers and it is called three-way handshake (Figure 4.3).

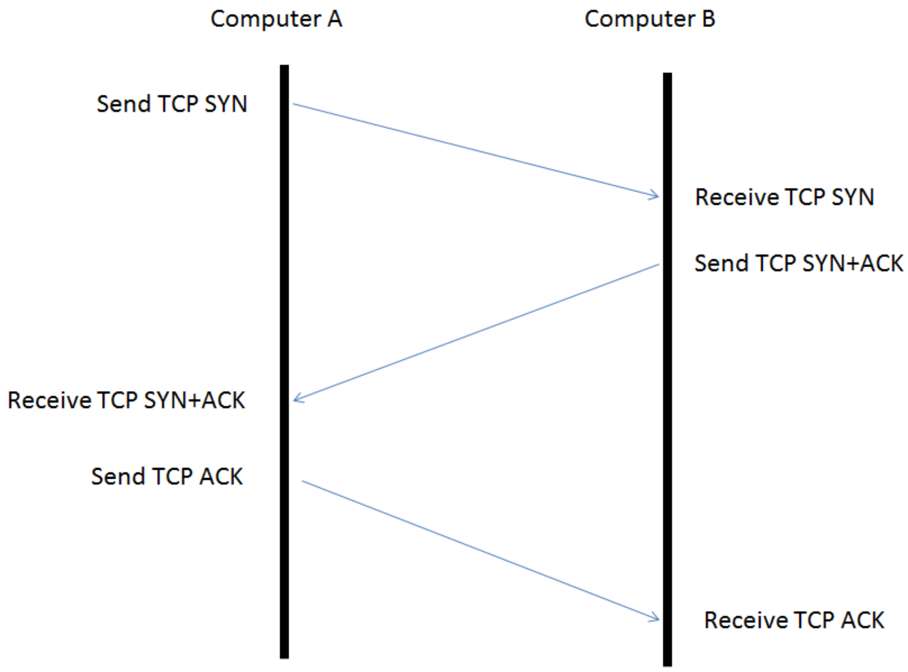


Figure 4.3 Three-way handshake

Now computer B received packets from computer A after connection is started. Computer B replies to computer A for each received packet. According to this information, computer A makes decision which packet it must be send to computer B. If any packet is loss, it sends to computer B again until all packets are received by computer A correctly. TCP uses sliding window concept when send data to destination port. Sliding window contains specific size of data start to send, after each acknowledges windows choose another data which has not sent yet from the data sequence.

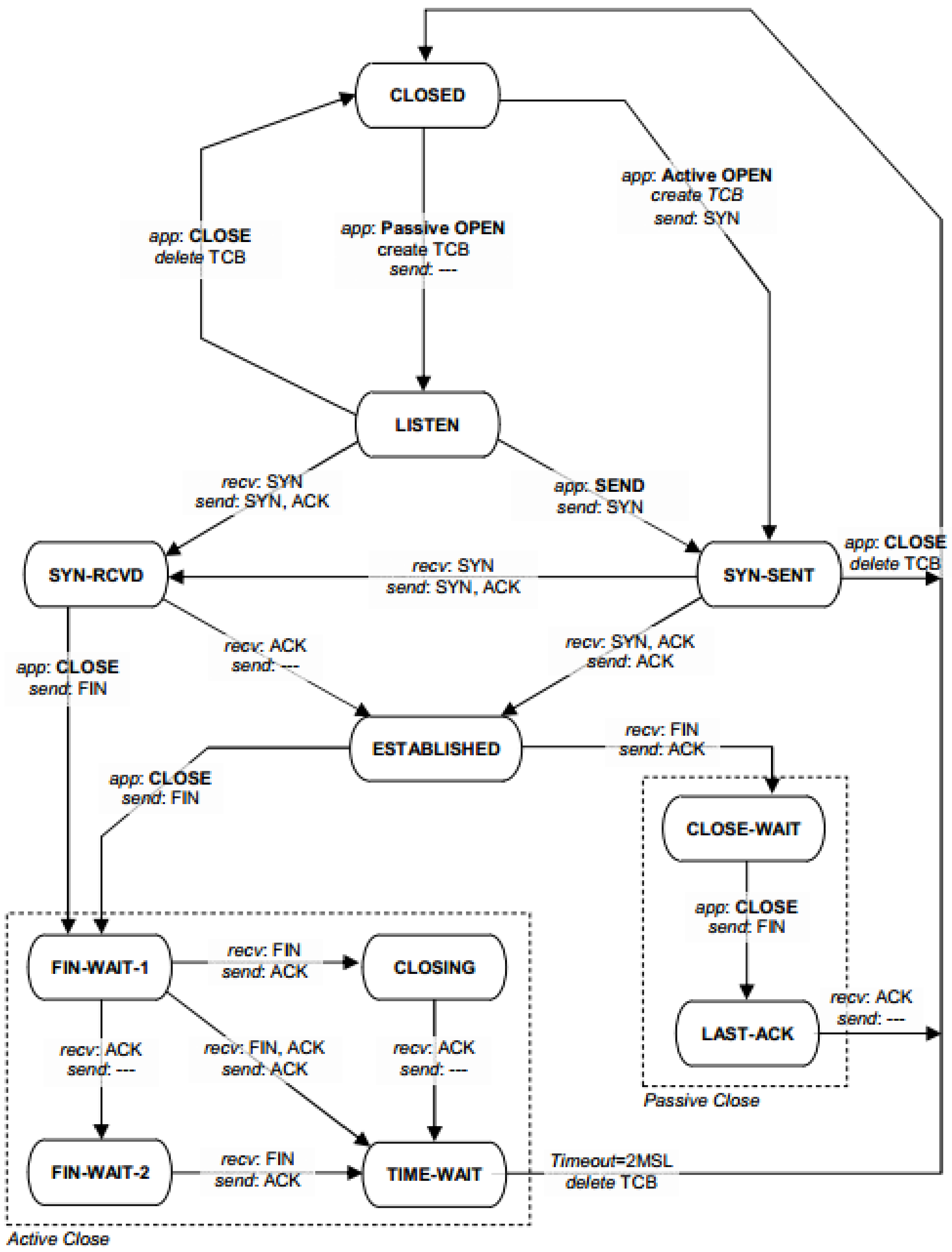


Figure 4.4 TCP state diagram (Zaghal & Khan, 2005)

After finish data transfer, one of the computers which have TCP connection sends finish message to other. For example, computer A sends TCP FIN message to computer B to finish the connection. Computer B replies with TCP ACK message and send TCP FIN to computer A. Computer A replies TCP ACK to computer B. thus, the connection is closed between computer A and B, it is called four-way handshake (Figure 4.5).

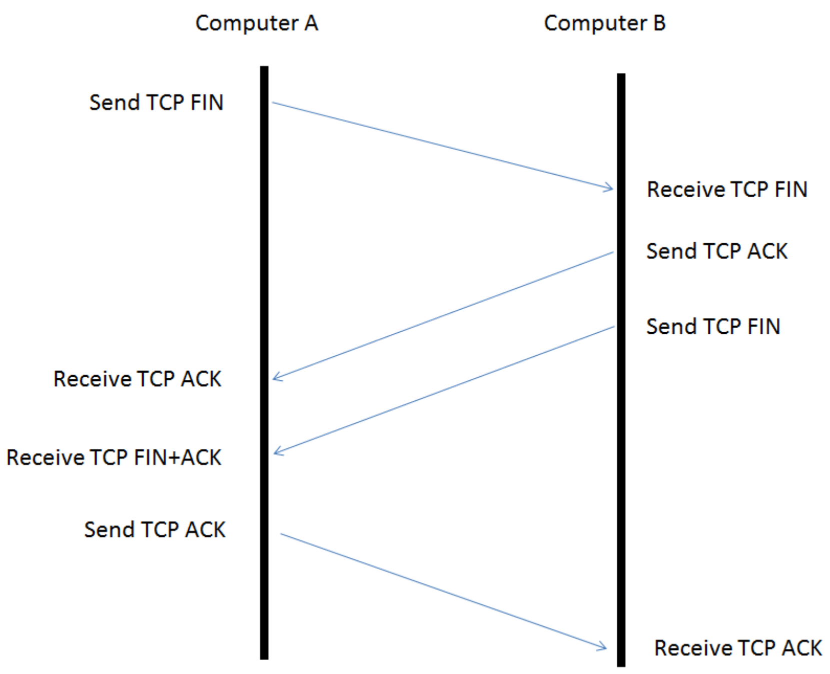


Figure 4.5 Four-way handshake

Packet Structure

Two computers must be TCP connection to communicate each other. TCP separates the data in data stream and adds header to create TCP segment. Each TCP segment is encapsulated into an internet protocol (IP) datagram to transmit data over network. Each TCP segment includes segment header and data section. Header part, which contains ten mandatory fields, illustrates in figure 4.6 and data section comes after header. Source port stores the TCP port information of sender computer. Destination port stores the TCP port information of the target computer. TCP gives number for each segment of the data and it stores this information in sequence number. When the target node take the packet, the acknowledge message is sending with acknowledge number. If the ACK bit is set this field contains the value of the next sequence number the sender of the segment is expecting to receive (Information Sciences Institute University of Southern California, September 1981). Data offset determine the size of the TCP header. Reserved is keeping to use in the future. After reserved part, there are code bits or flags part that is about control information of the segment. Each of them has 1 bit size in TCP header. Control bits from left to right:

- **Nonce Sum (NS):** This field is optional and added to ECN to protect against concealment of marked packets from the TCP sender.

• **Congestion Window Reduced (CWR)** is set when the TCP segment is received from sender host ECE flag set and had responded in congestion control mechanism.

• **ECE:** Echo indicates

• **Urgent Pointer (URG)** shows that urgent pointer field is important.

• **Acknowledgement (ACK)** shows that acknowledgement field is important.

• **Push Function (PSH)** asks to push the buffered data to the receiving application.

• **Reset the Connection (RST):** When the sender is abort the connection this part signals the receiver to reset the connection.

• **Synchronize (SYN):** The sender embarks to synchronize sequence numbers when the field is set.

• **No more data from sender (FIN):** When the receiver and sender finish the byte stream for current TCP connection, this field is set.

Window size store the size of the segment that is sending is current willing to receive. Checksum part checks the segment is received correctly or not. TCP can runs over IPv4 and IPv6, the checksum computation will be different according to IP address' type. Urgent pointer is used when the data must be sent urgently. Options part generally store the maximum size of the TCP segment.

<table>
<tr><th>Offsets</th><th>Octet</th><th colspan="8">0</th><th colspan="8">1</th><th colspan="8">2</th><th colspan="8">3</th></tr>
<tr><th>Octet</th><th>Bit</th><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td><td>31</td></tr>
<tr><th>0</th><td>0</td><td colspan="16">Source port</td><td colspan="16">Destination port</td></tr>
<tr><th>4</th><td>32</td><td colspan="32">Sequence number</td></tr>
<tr><th>8</th><td>64</td><td colspan="32">Acknownledge number (if ACK set)</td></tr>
<tr><th>12</th><td>96</td><td colspan="4">Data offset</td><td colspan="3">Reserved</td><td>NS</td><td>CWR</td><td>ECE</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td colspan="16">Window Size</td></tr>
<tr><th>16</th><td>128</td><td colspan="16">Checksum</td><td colspan="16">Urgent pointer (if URG set)</td></tr>
<tr><th>20
...</th><td>160
...</td><td colspan="32">Options (if data offset > 5. Padded at the end with "0" bytes if necessary.)</td></tr>
</table>

Figure 4.6 TCP header

TCP/IP

TCP provides reliable and in order transfer on IP which is unreliable services. TCP includes ACK and timer to satisfy the reliability. It approves when the packet is received, but it sends packet again when the packet is loss (Kurose & Ross, 2003). TCP can use Fast Retransmit algorithm to send missing packet unless timeout expires. If the same ACK information is sent in three times that packet sends again in this algorithm (W. Stevens, January 1997). TCP uses sequence number to determine loss packet and same packet which are transmitted. TCP can send more than one packet at the same time with pipeline method. To determine the packet number this will send as using flow control and congestion control. Flow control specifies how many packets the receiver will be accept, congestion control specifies how many packets the sender delivers according to bandwidth. Then choose the minimum packet number to transmit (W. R. Stevens, 1994). This packet number represents as window size.

Comparison UDP with TCP

There are a lot of difference between UDP and TCP which is given in table 4.3 so properties of TCP are much better than UDP to transfer video data. The most common similarity is both of them have source port, destination port and checksum fields in header structures.

Table 4.3 Differences TCP and UDP

TCP	**UDP**
Stream-oriented	Datagram-oriented
Connection-oriented	Connectionless
Reliable delivery	Unreliable delivery
Provide flow control	Does not provide flow control
Has order for data packets (Sequencing of data)	Has no inherent order (No sequencing of data)
Absolute guarantee that data will send correctly without lossy in the same order.	There is no guarantee for data transfer.
Header size 20 bytes	Header size 8 bytes
Sends acknowledge	No acknowledgement
Heavyweight	Lightweight
Retransmission of lost packets.	No retransmission of lost packets
Slow because of error checking mechanism	Faster, simpler and more efficient but less robust

5. VIDEO STREAMING

Video streaming is multimedia streaming media that sends the content in compressed form over the internet to the end user and the content are displayed by the viewer in real time. The end users do not have to wait to download the entire video file to play it. The media sends the content in a continuous stream and it can be played when it arrives. The compression of the media can be with any video codec. The video stream is compressed using a video codes such as H.264 or VP8. Encoded video streams are assembled in a container bitstream such as MP4, FLV, WebM, ASF or ISMA. The video stream can be transport with a network protocol such as UDP, RTP or TCP.

In figure 5.1 shows the work principle of the streaming on video. Streaming technique used for transferring large multimedia files quickly without downloading entire file. With video streaming the client browser can start downloading the data before the entire file has been transmitted.

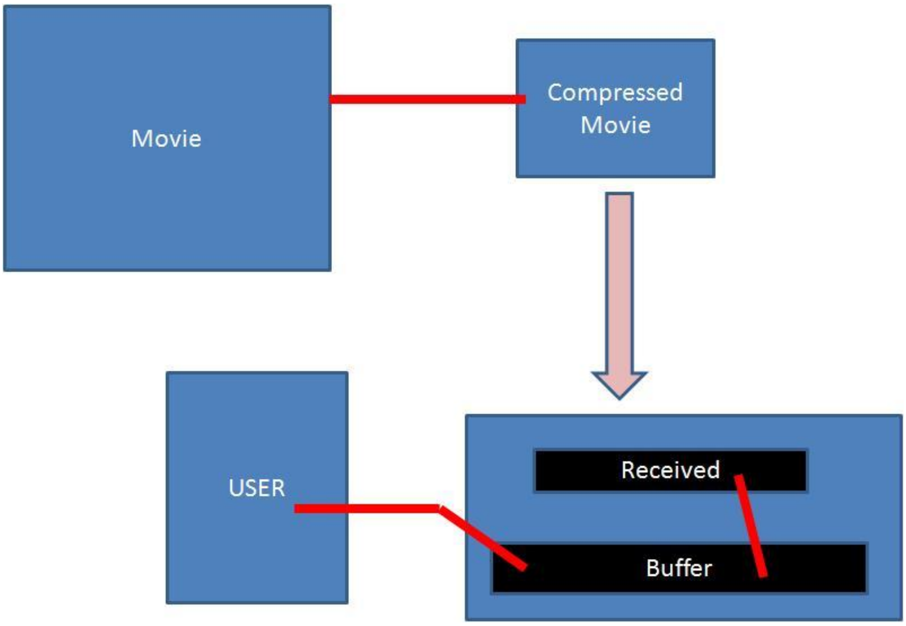


Figure 5.1 Video Streaming

Example 1: Assume that in figure 5.1 the big blue square is movie which the user wants to watch on internet. The size of the movie is very big to download entire without any interruption. We need to use codec to make movie smaller. So, the codec compresses the movie and this compressed movie transmitted to target. When the data received, the application is processing data and converting it to sound and picture to get video and this data stored in buffer. If streaming client receives the data more quickly than required, it needs to save the excess data in a buffer. If the data does not come quickly enough, the presentation of the data will not be smooth. While user watching the movie, buffer continuously sends these data to user. If everything works correctly, which is means that there is always extra data in the buffer, it was signing is everything live from the user perspective.

There are a lot of online web streaming activities. These web applications provide the user whether they want to watch video, news recaps, TV shows, tutorials or funny videos and user can watch video in any time without downloading all video file. The most popular websites are youtube.com, vimeo.com, metacafe.com, hulu.com, veoh.com which provide videos to their user with streaming. When we look this web page, there are some advantages and disadvantages part of them.

1. **Youtube.com**

   It is the most popular video streaming website which offers the all user to submit and watch thousands of video in different topic freely. The advantages of Youtube are high quality video playback; support almost all video formats, offer 16:9 aspect ratio. The disadvantages of it are to limit video access and the video limit of 100MB.

2. **Vimeo.com**

   This website allows the user to upload up to 500MB of video content per week. It supports full HD streaming and widescreen format and also it is suitable to watch and share high-quality and HD personal video because of a wide array of video codec support. On the other hand, convert a single video takes over an hour because it is used to upload large movie files.

3. **Metacafe.com**

   This website duplicate videos from Youtube, but there is a wide array of video tutorials on many subjects such as magic and science experiments. This tutorials is high-quality but the video resolution is high and videos are pixelated due to up-sampling

4. Hulu.com

Hulu which offers a wide array of free TV shows legally, videos can be viewed in 360p for slower Internet connections. But high definition video streaming is not available for most videos.

5. Veoh.com

Veoh offers TV shows for free and the video stream in higher quality resolution.

In generally, this video transmission process presents some advantages and disadvantages.

Advantages of the video streaming:

- Minimal Wait
- Instant viewing
- No long download times
- Do not have to waste memory space on hard drive
- Use specific bandwidths
- More security of content publishing
- Professional Training
- Education

Disadvantages of video streaming:

- Slow playback
- Possible start/stop of video if connection is not good
- Copyright issues
- Poor quality of video

• Quality depends on number of people also using the video

• Bandwidth

• Cost

5.1. Video streaming technologies

There are different technologies for streaming video over the internet. Some of them are data- independent, the others are data-independent.

5.1.1. Data-independent delivery technologies

5.1.1.1. Automatic Repeat request (ARQ)

ARQ is suitable only for point-to-point needs feedback, and added delay arbitrarily large (Vitali, October 2007). If the data losses appear time to time, this technology is convenient to send packet successfully only once. If the data loses frequently, data retransmits. The receiver always sends ACK to sender. If the ACK is not received, sender wait limited time then sends data again. When the receiver replies ACK as "the data is received wrongly", the wrong part is sent again.

5.1.1.2. Forward Error Correction (FEC) / Erasure Recovery

FEC or channel coding is a digital signal processing technique used for errors in data transmission over unreliable or noisy communication channels. FEC is no feedback required, all-or-nothing performance, waste of capacity when tuned for worst case, complexity, significant added delay (Vitali, October 2007). Sender uses error-correction code (ECC) and encodes message in a redundant way. Redundancy satisfies the receiver to detect and correct the errors without retransmission. FEC uses redundant bit which is added parts of the data again when transmits message to receiver. This information may be or not seen in encoded output. When the losses are too much, added redundancy is not enough and losses are not recovery. In this situation, the final video decoded quality will be very bad. The losses are very important to recover the video packets with redundancy. Encoding and decoding of redundant packets need memory and computational power so the complexity can be very high (Vitali, October 2007).

5.1.2. Data-dependent delivery technologies

5.1.2.1. Robust source coding

There are different techniques to aim robust source coding. When we encode the video packet, the importance of the packet is related with more efficient video encoder. Also the loss of packet is related with compression efficiency. If the compression efficiency is high, the loss of packet has destructive effect. Vice versa the loss of packet has slight effect (Vitali, October 2007). Prediction, transform, quantization, and entropy coding techniques are used for more efficient encoding. After efficient squeeze the video, video must be sent it to neighbors. Transmission can affect robustness because of transmission delay.

5.1.2.2. Layered Coding (LC)

It is similar to Multiple Description Coding. But there is difference between two coding such as dependency. LC entails prioritization and recovery mechanism, allows efficient scalability (Vitali, October 2007). LC has two layers; one base layer and several enhancement layers. Enhancement layers can use after one another to improve the decoded quality of the base layer. If it is necessary, layers can be dropped but not base layer. Enhancement layer can be drop, last enhancement is dropped first. Each layer has different importance on decoding; the most important one is based layer. So the recovery mechanism needs to guarantee to transmit at least base layer.

5.1.2.3. Multiple Description Coding (MDC)

Multiple description coding (Goyal, 2001) is a kind of data transmission coding method which is used to enhance the error resilience. It is very robust and the quality very good even at high loss rates.

Assume that a packet must be send to all nodes (A, B, C) on network is shown in figure 5.2. First the source is encoded by any encoder which is lossy. Packet is created to send all receivers. Then packet is sent and to guarantee the packet received the all node send more than one the packet. When the packet reaches any receiver more than one time, it is not advantages. When the packet is not received the packet is lost and this node never will be get. This is simple scenario to send any packet. On the other hand, the source can divide and all packets are not same. In this scenario more received packet mean more quality of the source. The second scenario satisfies with MD coding.

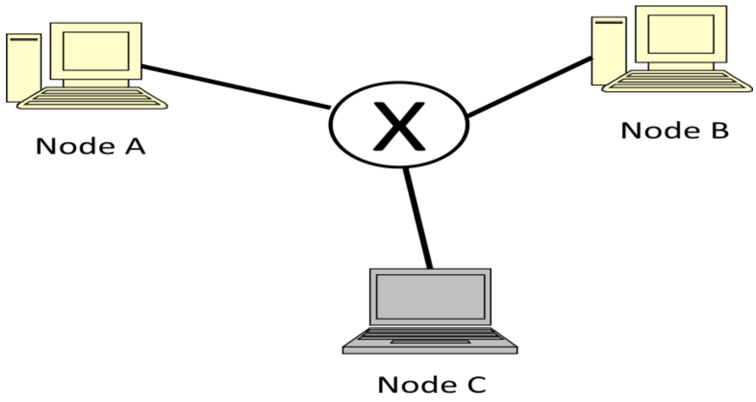


Figure 5.2 An example of network

MD separates a single media stream two or more independent sub parts, each of them are called description. All description is individually packetized and sends either same or separate channels. Description has more contributions such as spatial or temporal resolution, signal to noise ratio, and frequency content when packet transmits. Description can be either same or different priority.

MD is an important coding because it provides high video quality even the any of the packet lost when transmission and the packet is not retransmitted. MD generally is used in real time interactive application such as video conferencing which is not possible to retransmit packets. It can be preferred in simple network which has no feedback and is need retransmission.

When we send a picture, first encodes picture with multiple description coding. For example, to create description we divide picture both horizontally and vertically to get four descriptions (Figure 5.3). These descriptions can be sent in either same or different channels. If there is no any loss, the picture will be decoded correctly. If there are any lossy packets, MD coding can be fixed the picture with motion compensation. The missing part can be predicted and fixed according to neighbor pixels. Motion compensation also satisfies integrity between each frames of the video.

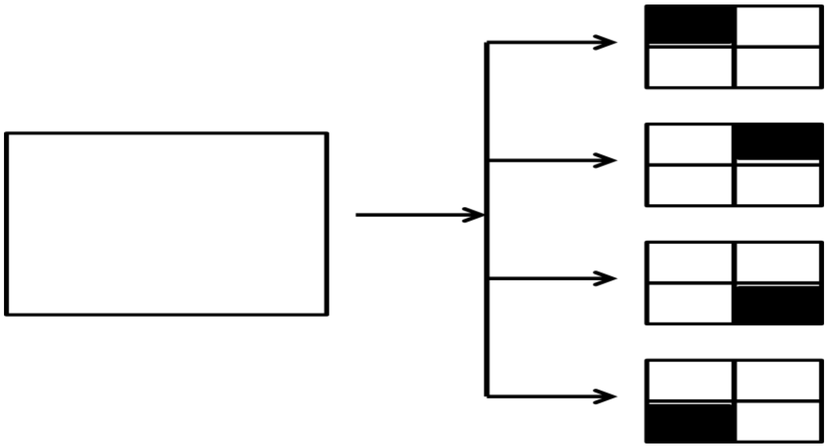


Figure 5.3 Preprocessing stage of MDC in picture.

MDC is more robustness and scability. If there is any corruption the sequence of the descriptions, it can be reconstructed with other received packets. Thus, the user does not notice the corruption in picture or video. MDC overwhelms limited bandwidth and packet loss.

MDC is used most commonly in packet networks, distributed storage, frequency hopping wireless systems, hybrid digital broadcast.

In (Vitali, October 2007) mentions about that there are three different compression algorithms to transmit the same picture. These three source codes respectively non-progressive, progressive and multiple description. First two algorithms use retransmission protocol, MD is not. Assume that there is software which send packet in specific time slot. This software starts to send packet to receivers. One of the packets is lost for example the third one. In this situation first two algorithms will be retransmit packet until it is received. On the other hand, MD will be lost the packet because it is not use retransmission protocol. Packet received time will be increase because of retransmission for the first two algorithm. In first two algorithms, the packet will be received to target node despite long time. On the contrary, MD coding recovers the lost part unless the packet is never received and the picture will have minimum aircraft. As a result, if there is no lossy packet, progressive code gives the best picture quality. But, MD has the best quality and is very fast when the packet is lost.

6. MULTI-STATE VIDEO CODING

There are different video coding techniques and standards to improve video quality. One of them is Multi-State Video Coding. Multi-State Video Coding is a multiple description scheme where the video is split into two or more subsequences.

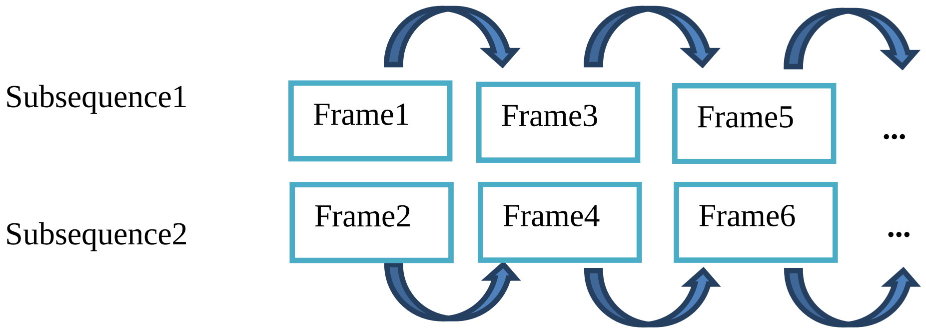


Figure 6.1 Subsequences of video in MSVC.

Why is it important?

Assume that, we have K packets to send the destination address. We give a number for each packets start from 1 to K. In a progressive transmission, the quality depends the number of successive packets received, starting from first packet, increases. If the packets are sent and received without loss, the progressive transmission works well. On the other hand, the sequences of the packet transmission will be changed if a packet is loss. The sequences of the packet are important when we sent a video to destination address. If there is any loss packet, not only the quality of the video will be decrease but also the transmission will be delayed.

With MSVC, we can separate video to more than one subsequences; the first subsequence is include odd numbered frames, the second subsequence is include even numbered frames like Figure 6.1. Each subsequence is encoded and transmitted separately and can be decoded independently. Thus, error resilience of the system increases because reconstruction of the video is well even there are some loss frames. The lost frames in one subsequence are reconstructed by using state recovery such as interpolation.

Interpolation is a numerical analysis method of construct a new data point within the range of a discrete set of known data points. In this scenario, each pixel value in each video frames is discrete set of data points. These points are known, the loss frames pixel values is unknown data point. This method used the previous and the last frames from other subsequence to generate loss frame with using (6.1).

$$y = y_a + (y_b - y_a)\frac{(x - x_a)}{(x_b - x_a)} \qquad \text{at point } (x, y) \qquad (6.1)$$

In formula (6.1), if the x_a and x_b is equal, the formula will be

$$y = \frac{(y_a + y_b)}{2} \qquad (6.2)$$

We will use (6.2) formula, because we try to find unknown value same location with previous and next frame. To find the first point from unknown frame, we apply (6.2) and find result 0 (zero) as shown in Figure 6.2.

X→

0	1	1	1	1
1	0	0	0	0
0	1	1	1	1
0	0	0	0	1
1	1	1	1	0

Previous Frame

X→

0	0,5	0,5	1	1
0,5	0	0,5	0	0
0	0,5	0,5	1	1
0	0	0	0	0,5
0,5	0,5	1	1	0,5

Unknown Frame

X→

0	0	0	1	1
0	0	1	0	0
0	0	0	1	1
0	0	0	0	0
0	0	1	1	1

Next Frame

Figure 6.2 Using Interpolation in video frame to find values of the unknown frame.

A MSVC system (Figure 6.3) has two main components: multiple state encoding & decoding and path diversity transmission (Flierl & Sikora, 2005). Encoded part generates transmission packets for each subsequences, decoded part generates frames from receiver packets. Path diversity transmission part is related with transmission of each encoded packet to the destination address separately. After system generates the unknown frames of the video, system merges frames to create reconstructed video.

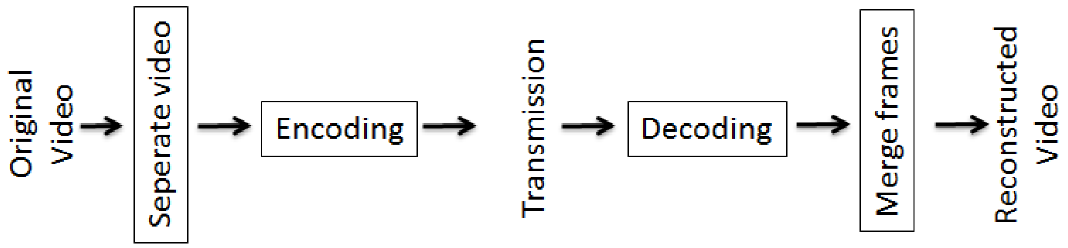


Figure 6.3 MSVC System

Encoder part of MSVC

A video input is separated in two subsequences (even and odd frame). These subsequences are coded in bitstreams separately. Each of them has different prediction loop and state and independently decodable (Apostolopoulos & Wee, 2001). After encoder part, each stream is separately transmitted to destination address.

The aim of the MSVC to split video into description is using state recovery properties (Ekmekci & Sikora, 2003). If any frames or description is lost this lost frame may reconstruct with state recovery which increases error resilience. On the other hand, if the video delivered within single description, the loss packet cannot construct so the video quality decreases (Ekmekci & Sikora, 2003).

Decoder part of MSVC

A video streams receive to destination then decoder part of MCVS reconstructs video. MSVC reconstructs video and displays it with high video quality when all subsequences are received to target properly. If an error is occurred when transfer the stream to target due to any kind of reasons such as time out, corrupted connection, etc., MSVC reaches to previous and next frames to reconstruct the video.

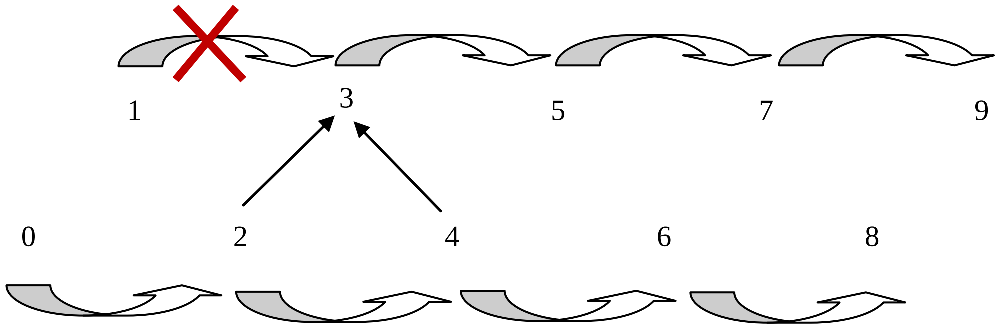


Figure 6.4 An example of state recovery in balanced description

In figure 6.4, we see an example of state recover in balanced MD coding that has approximately same number of frame in each subsequence. Assume that the third frame did not reach the target because there is a corrupted connection. MSVC applies frame 2 and frame 4 from the other subsequence. It recovers frame 3 with using state recovery property. In unbalanced MD coding schema state recovery works same as balanced. Figure 6.7 shows unbalanced two descriptions, the working principle is the same to recover frame 3 use frame 0 and frame 4 from the other description.

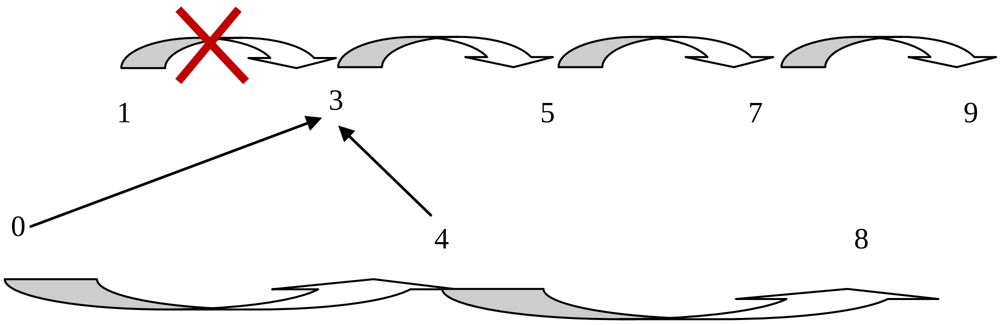


Figure 6.5 An example of state recovery in unbalanced description

State recovery is similar to Motion Compensation interpolation that estimates the loss frames using both previous and next frames.

Path diversity transmission of MSVC

MSVC delivers the packets either same or different path. Each packet stream received to target in any path over the network.

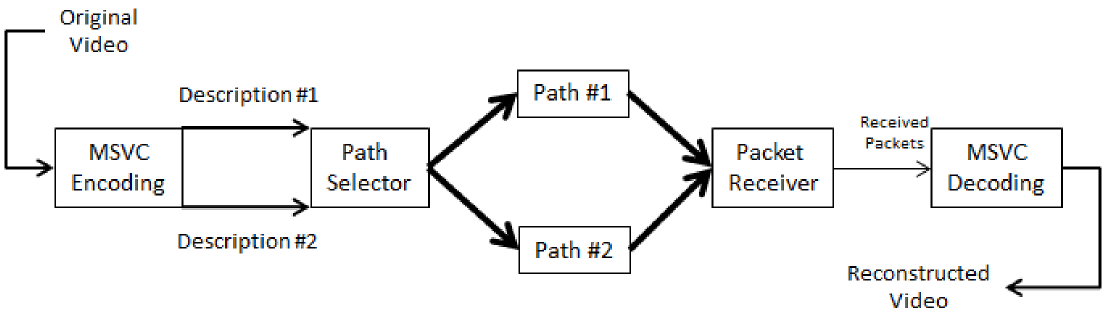


Figure 6.6 Multi State Video Encoding/Decoding and Path Diversity

Path selector separates the encoded streams to path and this packet is delivered over this path to receiver. In figure 6.6, MSVC splits video two streams and the path diversity system has two paths to send packets. Each packet reaches to receiver in either Path#1 or Path#2. Then packet receiver sends received packets to MSVC decoder.

7. NETWORK CODING ALGORITHMS

Another important part of the system is how we can generate the network code to transmit packets to target address. We need to construct and select the network code. When we generate this code, we must consider this codes guarantee decodable by target node. Then we must select the best code which optimizes video quality. We have two algorithms to generate and improve video quality. The first algorithm is NCV; create a network candidate code to improve throughput and video quality. The second algorithm, NCVD, based on NCV to generate candidate code but it considers more coding option.

We deal two different coding algorithms to broadcast packets to the entire neighborhood. In our network, there is an intermediate node which is able to transmit packets to other nodes and/or clients. When sending the packet, we use different algorithms to improve video quality and throughput. First we will give some description which is related coding algorithms and Table 1 gives some terms of the algorithms. After that we will mention about these algorithms in 7.1 and 7.2 sections.

Code Selection at an Intermediate Node: Intermediate node is a part of any network and it receives packets and sends these packets to neighbors. Assume that, it receives N packets from different video streams and forwards them to N nodes in its neighborhood (Seferoglu & Markopoulou, 2009). The intermediate node has a transmission (T_x) queue that store incoming video packets. One packet is selected from queue and it is called primary packet the destination node of this packet is called target node. Primary packet is main packet so intermediate node try to transmit this packet to target node. Primary packet must be active packet and can be the head of the queue according to algorithm it can be change. All packets are candidate side packet other than primary packet. After choosing primary packet and side packets, intermediate node are XOR-ed together into a single packet and this packet is called the network code. Then according to network coding algorithms, which are describe in the 7.1 and 7.2 parts, intermediate node selects and transmits network code to maximize the total video quality and throughput.

Table 7.1 Terminology of the coding algorithm.

Term	Definition
Primary Packet	This packet is selected from T_x queue before generate network code. This packet must be active and it is main packet so we have to transmit this packet. When we generate network code this packet must be in all network packets.
Side Packet	This packet is selected from T_x queue other than primary packet, this packet included in the network code. This packet can be either active or inactive.
Network Code	Single packet which is generated from one primary packet and all side packets after XOR-ed all packets.
Active Packet	Packet, which can be primary packets, from T_x queue.
Inactive Packet	Packet, which cannot be primary packets, in T_x queue. This packet has already been transmitted within last transmission.
Target Node	The recipient of the primary packet.
T_x Queue	This is output queue of the transmitting node (intermediate node) which store packets from different video stream.
R_x Buffer	The receiving queue of the receiving node. It store received packets which are belong to this node.
Virtual Buffer	It store overhead packets which are belong to another node.

Receiving, Overhearing and ACKing a Packet (at Receiving Nodes): Intermediate node choses network code and transmits to all nodes in the neighborhood. Some nodes successfully receive network code but some nodes do not depending on the channel conditions. The most important part is sending code to target node. If target node receives it, target node decodes it. After decoding code, target node stores the primary packet in its receive buffer R_x and send an acknowledgement (ACK) back to the intermediate node. Intermediate node makes received packet as inactive. The other nodes overhead the transmitted packet, they also try to decode network code. If they get a new packet which is belong to itself, they store this packet in their receive buffer and send ACK to intermediate node. If they get a packet which is belonging to another node, they store this packet in virtual buffer and this packet is called overhead packet.

Active / Inactive Packets (at an Intermediate Node): Intermediate node stores all packets in T_x buffer. When intermediate node starts to code packet, chooses packet from T_x buffer either active or inactive. Before starting transmits any packets, all of them in T_x queue is marked as active. The first packet from the head of queue chooses as primary packet and intermediate node try to receive this packet to target node and wait an ACK from target node. Until ACK received to intermediate node all of the packets (primary packet and side packets) which is used in network code, marked as inactive and stay in T_x buffer. Intermediate node removes primary packet from the T_x buffer if the target node sends an ACK. If the packet was not received by target node according to any channel condition, the packet is marked active and the process is repeated. If the process is failed because of deadline expires, the packed removed from the T_x buffer. So, the packet is loosed.

Importance of Packets: In transmitted queue, all packets have priority. The priority of the packet is marked when the packet comes in T_x buffer. Some packet may have the precedence than other packet. So, we give maximum priority value to this packet than others. When the intermediate node chooses the packet from T_x queue to transmit the nodes, it is looks priority value of the packets. If the packet has high priority, intermediate node tries to send it to target node. If the priority of the all packets in T_x is some just find the first active packet to transmit its target node.

Handling Video vs. Data Packets: The intermediate node just stores all data packet, data includes only video packets. Our coding algorithm using when our T_x buffer includes video packet from different video stream.

7.1. NCV Algorithm: Network Coding for Video

Network Coding for Video (NCV) choses the best network code from generated candidate code to improve video quality. Intermediate node includes several video streams. When intermediate node sends this video stream to target node, it generates several combination which depends on the content of the virtual buffers at the clients. NCV deals with selection of this candidate code to send target node.

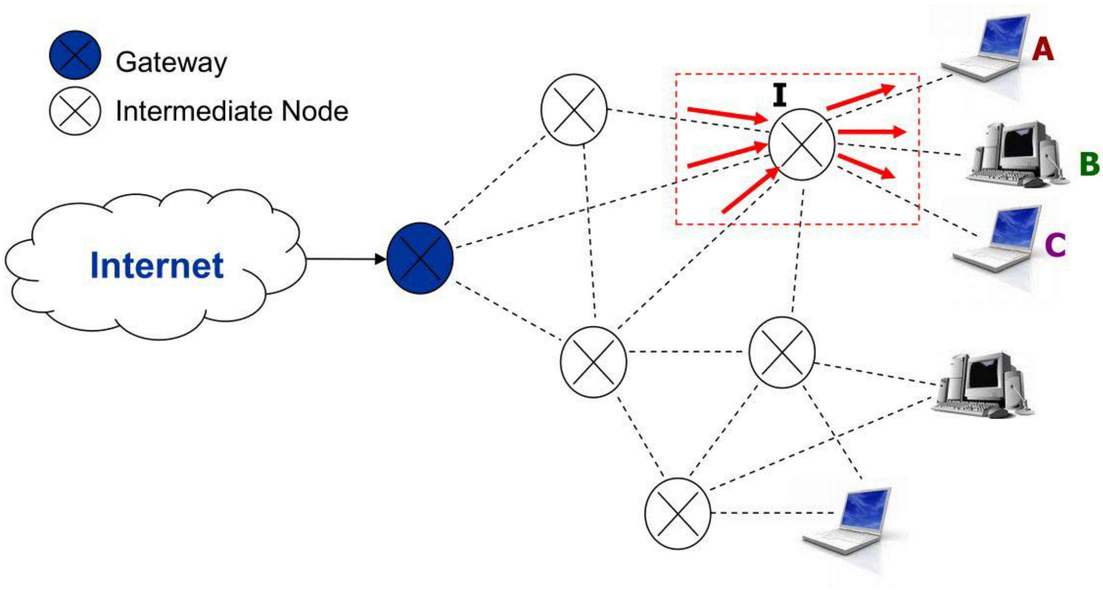


Figure 7.1 A network example. I is an intermediate node, A, B, C are receiving nodes.

There are two main problems; code construction and code selection. Code construction deals with generating candidate codes. Code selection deals with which candidate code is the best to improve video quality.

Code Construction in NCV: There is more than one node in any network. Assume that, N is a number of nodes in one network. This network node is n_1, n_2, … n_N. One of them is intermediate node which is send packet to neighbor nodes. Number of packets in the T_x queue of the intermediate node n is ϕ_n and the packets like that ϕ_n = {p_1, p_2, …, p_{ϕ_n} }. Choose primary packet which is first active packet p_i from the head of T_x queue and this primary packet has target node $t(p_i) \in N$. Then it applies XOR operation, input of the operation are primary packet and side packets. Intermediate node n will transmit this XOR-ed packet to not only target node but also all neighbors. Target node $t(p_i)$ will decode and get p_i on the other hand other packets are overhead. The overhead packets $\Psi_{t(p_i)}$ denoted by $\Psi_t(p_i) = \{v_1, \dots, v_{\Psi_t(p_i)}\}$. The candidate codes at node n are:

$$c_k^i = \{p_i\} \cup S_k^{t(p_i)}, k = 1,2, \dots, 2^{\Psi_{t(p_i)}} \tag{7.1}$$

where $S_k^{t(p_i)}$ is the k^{th} subset of $\Psi_{t(p_i)}$(Seferoglu & Markopoulou, 2009). To generate network code, do bit-wise XOR after choosing the best candidate code.

Code Selection in NCV: After obtaining all candidate code, we must choose the best network code according to some parameters which defined in (Seferoglu & Markopoulou, 2009).

p_i : primary packet

$t(p_i)$: target node

$\{c_k^i\}_{k=1}^{k=2^{\Psi_{t(p_i)}}}$: candidate codes equation (7.1)

Let $I_k^i(n_\eta)$ be the improvement video quality at node n_η for η = 1, 2, …, N, when c_k^i is received and decoded:

$$I_k^i(n_\eta) = \sum_{l=1}^{L_k}(1 - P(l))\Delta(l)\,\gamma(l) g_l^k(n_\eta) d_l^k(n_\eta) \tag{7.2}$$

where each factor in this formula is defined as follows:

- The number of original packets included in network code c_k^i is L_k. L_k packets can be useful to a particular node n_η, but different packets are useful to different nodes.
- d_l^k and g_l^k are indicator functions that express whether code k is useful for node n_η . We define $d_l^k(n_\eta)$ = 1 if c_k^i is decodable at node n_η , otherwise is 0. We define g_l^k (n_η) = 1 if packet l is targeted to node n_η , otherwise is 0.
- $\Delta(l)$ is the improvement in video quality (SNR) if packet l is received correctly and on time at node n_η .
- $\gamma(l)$ is priority of the flow packet l belongs to. Priority depends on the importance of the packets in flow. So, some packets have more importance than other, but all packets in the same flow have the same importance. The default importance is 1, but some of them can have more importance then higher importance should be assigned to them.

- $P(l)$ is the probability of the loss packet l according to any situation such as late arrival or channel errors.

The sum of the video quality improvement at all clients due to code c_k^i :

$$I_k^i = \sum_{\eta=1}^{N} I_k^i(n_\eta) \tag{7.3}$$

NCV algorithm is given below in Algorithm 1. For each transmission algorithm chooses the primary packet and side packet to get all candidate network codes. NCV chooses the best network code, which decrease the total video quality, among all candidate codes. The best network code transmission to target node. This network code sometimes has no side packet according to the content of the virtual buffers. In such case, network code will be simple $\{p_i\} \cup \emptyset = \{p_i\}$, just transmit primary packet.

Algorithm 1 The NCV Algorithm

Step 1: Initialization: $I_{max}^i = 0$, $c_{max}^i = \emptyset$

Step 2: Choose the first head of queue active packet as primary p_i.

Step 3: Let $t(p_i)$ be target node of packet p_i. Let $\{v_1, \ldots, v_{\Psi_{t(p_i)}}\}$ be the overhead packets at $t(p_i)$.

Step 4: **for** k = 1 ... $2^{\Psi_{t(p_i)}}$ **do**

Step 5: $c_k^i = \{p_i\} \cup S_k^{t(p_i)}$

Step 6: calculate I_k^i with Eq. (7.3)

Step 7: **if** $I_k^i > I_{max}^i$ **then**

Step 8: $I_{max}^i = I_k^i$, $c_{max}^i = c_k^i$

Step 9: **end if**

Step 10: **end for**

Step 11: Choose c_{max}^i as the network code. XOR all packets and transmit.

When we look Figure 7.1 node I is intermediate node which is takes video streams, A, B and C nodes are neighbors of the intermediate node. Assume that, intermediate node receives three independent video streams to send its neighbors. Intermediate node takes all video streams from Internet through the gateway and it stores packets in a FIFO T_x queue. T_x queue includes $\{A_1, A_2, \ldots\}$ packets destined to node A, $\{B_1, B_2, \ldots\}$ packets destined to node B, $\{C_1, C_2, \ldots\}$ packets destined to node C. Each client A, B, and C have its virtual buffer which includes overhead packets for each client. In figure 7.2, there is detail information about virtual buffer of the each client. $\{B_1, C_1\}$ is overhead packets of Node A, $\{A_1\}$ is overhead packets of both Node B and Node C from previous transmission. Intermediate node has T_x queue which includes $\{A_1, B_1, C_1, A_2\}$ packets to send to target. Some packets is marked as active, some of them is inactive in the output queue. To find primary packet, we choose first active packet from head of queue. Consider A_1 is the first active packet from the head of queue in Figure 7.2, we select it as primary packet. Now, A_1 is selected as primary packet, we must select side packets, which are either active or inactive, from output queue. After choosing side packets, it constructed different network codes which should be decoded by node A to obtain A_1 packet. Network code will be generated from $\{A_1, B_1, C_1\}$ packets. Candidate network code generated like that $c_1 = A_1$, $c_2 = A_1 \oplus B_1$, $c_3 = A_1 \oplus C_1$, $c_4 = A_1 \oplus B_1 \oplus C_1$ can all be decoded by node A.

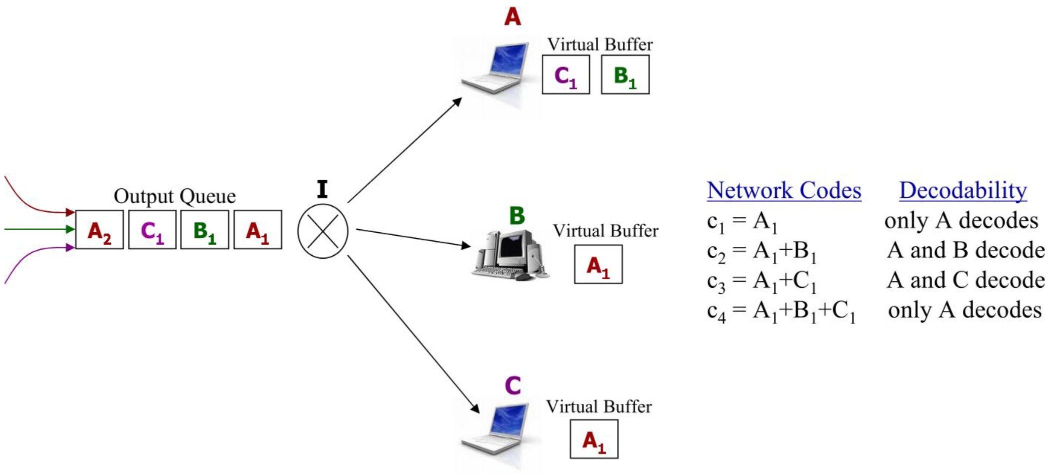


Figure 7.2 Example of Network Coding for Video (NCV).

All of the candidate network code can be decoded by node A to get A_1 packet. When we compare this four candidate code c_1 and c_4 is decoded by only node A. Besides this, c_2 and c_3 are decoded by more than one node; c_2 is decoded not only by node A to get A_1 packet but also by node B to get B_1 packet, c_3 is decoded by not only by node A to get A_1 packet but also by node C to get C1 packet. Therefore, c_2 and c_3 are better codes than c_1 and c_4. Now the important part is which candidate code c_2 or c_3 is better to transmit over the network. NCV is decided to find which one gets the best improvement with code selection part.

7.2. NCVD Algorithm: looking into the queue in Depth

NCVD algorithm uses NCV algorithm to generate candidate network code but there are some differences. In NCV, the primary packet is chosen from the head of the T_x queue, just look the active packet. If the packet is inactive, it ignores this packet. Then NCV choose some side packets to generate network codes from the queue. In this situation, NCV is not notice that there can be another primary packet which maximizes the video quality more than previous primary packet.

NCV does not optimize the primary packet have two implications:

(i) The primary packet is important for video quality and

(ii) The candidate side codes are limited to those that are decodable for this single primary packet (Seferoglu & Markopoulou, 2009).

NCVD algorithm, which is given below, improves selection of the primary packets of the NCV algorithm. NCVD looks into all T_x queue "in depth", and deals with all packets can be as primary packets instead of looking just head of the output queue. According to these primary packets, NCVD generates more candidate network code than NCV. Choose the best network code from this candidate network codes to maximize the total improvement in video quality for all nodes.

Algorithm 2 The NCVD Algorithm

Step 1: Initialization: $c_{max} = \emptyset , I_{max} = 0$

Step 2: **for** every packet $i = 1, \dots , \varphi_n$ from the head of T_x queue **do**

Step 3: Consider this packet p_i, as candidate for primary

Step 4: Construct all possible codes c_k^i for p_i

Step 5: Determine the max improvement $I_{max}^i = max_k \, I_k^i$

Step 6: and the corresponding code $c_k^i : k = argmax \, I_k^i$ as in NCV

Step 7: **if** $I_{max}^i > I_{max}$ **then**

Step 8: $I_{max} = I_{max}^i, c_{max} = c_k^i$

Step 9: **end if**

Step 10: **end for**

Step 11: Choose c_{max} as the network code. XOR all packets and transmit.

NCVD has a new parameter depth d of the T_x queue which is used to choose primary packet before generating candidate network packet. When we consider d is equal to 1, this step same with NCV. On the other hand, when we consider d is infinite, this means NCVD looks all packets can be as primary packet. Having the more primary packet means that the more coding option so algorithm generates more candidate network packet than NCV. Therefore, NCVD improves better performance on video quality.

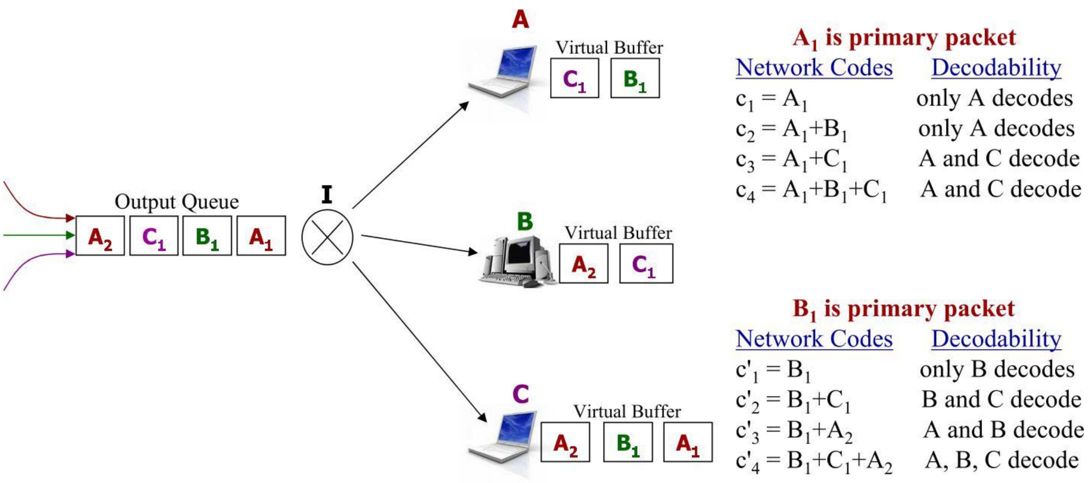


Figure 7.3 Example of NCVD.

In figure 7.3, we have same topology, I is intermediate node and A, B, and C nodes are clients of the network. T_x queue is the same but the virtual buffer of the each node is different. $\{B_1, C_1\}$ is overhead packets of Node A, $\{C_1, A_2\}$ is overhead packets of Node B and $\{A_1, B_1, A_2\}$ is overhead packets of Node C from previous transmission. Assume that, all packets are active in this example, so we can choose all of the packets as a primary packet. When we chose the first active packet A_1 as primary packet and B_1, C_1 as a side packets, intermediate node generate $c_1 = A_1$, $c_2 = A_1 \oplus B_1$, $c_3 = A_1 \oplus C_1$, $c_4 = A_1 \oplus B_1 \oplus C_1$ candidate network packets. The best network codes are c_3 and c_4. We can generate this code with NCV. When we choose the depth of the T_x queue as two. NCVD will generate more candidate code than NCV. The first option is when A_1 is primary packet; the second is when B_1 is primary packet. When we chose the second active packet B_1 as primary packet and C_1, A_2 as a side packets, intermediate node generate $c_1 = B_1$, $c_2 = B_1 \oplus C_1$, $c_3 = B_1 \oplus A_2$, $c_4 = B_1 \oplus C_1 \oplus A_2$ candidate network packets. The best network codes c_4. Now, we have three different candidate code c_3 and c_4 from A_1 as primary key, c_4 from B_1 as primary key. NCVD chooses the best candidate network packet according to achieve the maximum throughput and video quality. In this example, $c_4 = B_1 \oplus C_1 \oplus A_2$ candidate network packet is the best code. This example shows that the choosing correct primary packet is important when we generate candidate code which improves both throughput and video quality.

8. SIMULATION

Basically, our work contains not only video streaming but also network coding ideas. First we prepare the video with using video streaming technology. After applying Multi-State Video Coding, we get two subsequences. Then we apply network coding to generate single video packet from video frames in each subsequence. Then the packets are transmitted to not only target node but also all nodes in network. Each node takes packet and decodes to get the primary packet and side packets. If it is not received to target node, system generates frame from previous and next frames of the lost frame.

8.1. PRE-PROCESSING

In our work, we just deal with sending only video packets. First of all, we need to prepare videos which are stored in intermediate node. We specify which video we want to send to all networks, or which video wanted from any receiver on network. We need to seperate video to frames. Frames is a kind of picture takes from video sequence in specific time.

We use H.264 codec to determine the videos frames. We get all frames of the video. In our simulation, we use football_single video which is included american football game video. After applying codec, we get 100 frames from football_single. Each frame has unique named (like videoName.26lframeNo) with respectivelly football_single.26l1, football_single.26l2, football_single.26l3, …, football_single.26l99, football_single.26l100. We store these frames in local disc of the intermediate node.

8.2. SYSTEM DESIGN

In our system, we combine two approach MSVC and network coding. Also we have the transmission part after creating packets. First we need to apply MSVC, then encoded packets to generate single packet with network coding algorithm. Finally, we transmit these packets to all nodes.

First of all, our system needs to separate odd even frames from all frames. Intermediate node needs to two subsequence after applying MSVC. System generates even numbered frame subsequence and odd numbered frame subsequence. Intermediate node stored these two subsequences separately. Before applying network coding, system must generate these subsequences. Then system creates candidate network code from each subsequence.

Our system has two main component sender and receiver for transmission of the packets. You can think that sender part is designed to send encoded packet to all nodes. Receiver part takes encoded packets and decodes it.

Sender part has two different duties. The first one is sending encoded packet to all nodes in network, second is sending virtual buffer of each nodes to intermediate node. We separate these two duties with host number of each node. We gave the special host number to intermediate node such as 1010. If the host number of the node is 1010, the sender works to send encoded packet. Otherwise, it sends virtual buffer of own.

Also, receiver part has two different duties. It is same as sender part to separate these two duties. System checks the host number. If the host number is 1010, it starts to wait to receive the buffer information of each node. Otherwise, it starts to wait encoded packet and then decodes packet.

The listening and sending operation goes on transmission control protocol which is retransmission and reliable protocol. TCP is used most commonly to transfer video because of less lossy packet. We prefer to use TCP in our simulation to transfer video frames because we have almost 100 frames to send the destination. Think each frame as a picture. More than one picture comes together according to algorithm and XOR-ed to create single packet. Then each packet sends to nodes. And these nodes use these frames to create video. So, the sequence of the frame is important when they receive to target. Each correctly received support to recover the lost packet. TCP ensures that the packet will be send more reliable than other protocols.

Receiver creates TCP listener to take packets, sender creates TCP client to send packets. When we use the TCP, we need IP address and the port number of the each node. According to this address and port number nodes communicate each other. At the sender part, we also take the own port number of the node. On the other hand, we just need port number of the node at the receiver part.

When we start the system, sender needs to all buffer information each node. So, each node sends its own overhead buffer information to intermediate node. Intermediate node stores this information for calculation of the best code.

Intermediate node needs to create the best candidate code to send. First of all, some information about each frame must be generated from frame files such as node name, filename, binary and byte form of the frames, loss probability, snr, priority, active flag, and counter. System reaches the local disk to create this information of each frame. We generate T_x queue which include all video frames for either same or different nodes.

After creating T_x queue we apply to generate candidate network code and encode this code to deliver. Now, we apply network coding algorithm and each of them has different features when create XOR-ed packet.

Applying Network Coding for Video (NCV)

If we apply the first network coding algorithm which is Network Coding for Video (NCV), first we determine the first active frame from the T_x queue. This first active frame stores as primary packet and the aim of this algorithm to send this primary packet to its destination. Also we choose either active or inactive frames from the queue as side packets. To create XOR-ed packet we need a primary packet and more than one side packets. The number of the side packet is optional in our system.

The next step in algorithm is to generate the sub packet. We find all sub packets from the combination of primary packet and side packets. These sub packets is called candidate code and it must include primary packet. So, we check which sub packet include primary packet. For example, our primary packet is football_single.26l2 and the side packets are football_single.26l3 and football_single.26l4. We generate a packet includes three packets. System creates sub packets of this packet:

{football_single.26l2, football_single.26l3, football_single.26l4}
{football_single.26l2, football_single.26l3}
{football_single.26l2, football_single.26l4}
{football_single.26l2}
{football_single.26l3, football_single.26l4}
{football_single.26l3}
{football_single.26l4}

System chooses just some of them, which must include primary packet, as candidate network code. The candidate network codes are the first four sub packet;

Candidate code 1: {football_single.26l2, football_single.26l3, football_single.26l4}
Candidate code 2: {football_single.26l2, football_single.26l3}
Candidate code 3: {football_single.26l2, football_single.26l4}
Candidate code 4: {football_single.26l2}

Applying Network Coding for Video looking queue in depth (NCVD)

If we apply the second network coding algorithm which is Network Coding for Video looking queue in depth (NCVD), first we determine the first active frame as primary packet and side packets from the T_x queue. We generate candidate codes like NCV. In addition to NCV, NCVD looks the next active frame as primary packet and also the side packets of its. NCVD generates the next candidate codes. For example, we have four candidate code which are created with NCV, now our primary packet is football_single.26l3, side packets are football_single.26l4 and football_single.26l5. The new sub packets for this;

{football_single.26l3, football_single.26l4, football_single.26l5}
{football_single.26l3, football_single.26l4}
{football_single.26l3, football_single.26l5}
{football_single.26l3}
{football_single.26l4, football_single.26l5}
{football_single.26l4}

{football_single.26l5}

We have more candidate code than the NCV algorithm. Candidate network codes are:

Primary packet: football_single.26l2

Side packets: football_single.26l3, football_single.26l4

Candidate code 1: {football_single.26l2, football_single.26l3, football_single.26l4}

Candidate code 2: {{football_single.26l2, football_single.26l3}

Candidate code 3: {football_single.26l2, football_single.26l4}

Candidate code 4: {football_single.26l2}

Primary packet: football_single.26l3

Side packets: football_single.26l4, football_single.26l5

Candidate code 5: {football_single.26l3, football_single.26l4, football_single.26l5}

Candidate code 6: {football_single.26l3, football_single.26l4}

Candidate code 7: {football_single.26l3, football_single.26l5}

Candidate code 8: {football_single.26l3}

NCVD generated eight candidate codes and to create the more code we can look more depth for primary packet. In our system depth value is dynamic; we can change if we want. The default depth value is 2. System finds the maximum improvement from these eight candidate code according to its own primary packet.

Now, we have candidate network packet and we must select the best improvement packet in all candidate code. To choose the best one we use the equation 7.3 which includes two important functions. The first one is gIndicator function that determines this frame is send to target. The second one is dIndicator function that determines this frame is decodable at receiver node. These two functions return 1 value if it satisfies condition. Otherwise, it returns 0. After getting the maximum improvement candidate code, for example candidate code 5 is the best in NCVD algorithm.

The system will create encoded packet from the maximum improvement code. We use XOR operator to encode the packet. XOR generate 1 when the inputs are different, otherwise generate 0 (Table 8.1).

Table 8.1 XOR gate

Input A	Input B	Output A⊕B
0	0	0
0	1	1
1	0	1
1	1	0

Now, we have XOR-ed packet that is generated from more than one packet. Intermediate node sends this packet with TCP so it needs IP addresses and port number of the all nodes on network. Intermediate node finishes the work after sending.

Each node tries to decode the received packet. If they can decode the packet, they look the all frames in packet. If the node is owner of the frame, it is stored in its local disc. Other packets that are generated from received packet, store in overhead buffer. The most important part is the target node is received or not. If it is received, we achieved the goal successfully. On the other hand, the node generates missing frame(s) from previous and next frames.

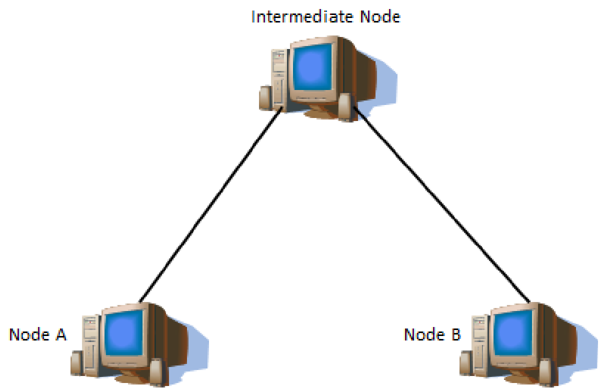
Finally, each node checks their received frames are missing or not. If there are missing frame or frames, it use state recovery of MSVC to generate missing frame(s).

8.3. SIMULATION RESULTS

First of all, we prepare our simulation environment to test our algorithms. We tried to test algorithms in different situation. We create our own simulation in different scenario. We have one intermediate node and more than one node that are connected with intermediate node. After creating intermediate node, we create other nodes and connect all of them with intermediate node.

We test our algorithms both on one computer (local host) and on more than one computer. In local host, one node is intermediate node that has 1010 host number; this host number represents the intermediate node so we do not give this host number to other nodes. Intermediate node has special duties in network. Intermediate node is the sender of our system and also it can communicate with all nodes in network. On the other hand, the other nodes have different host number to communicate with intermediate node. These nodes can send their virtual buffer information to intermediate node and take their video frames from intermediate node.

Besides working in local host, we created our own network with connect more than one computer with each other. We used subnet technique to connect computer with each other. For this scenario that is shown in Figure 8.1, we need one computer which has two Ethernet entrance, this computer is called an intermediate node and two computers for other node. We connect these two nodes with intermediate node separately. These two computers (Node A and Node B) just communicate with intermediate node not with each other.



IPAddress:	198.168.10.18
Subnet Mask:	255.255.255.240
Network ID:	198.168.10.16

IPAddress:	192.168.10.33
Subnet Mask:	255.255.255.240
Network ID:	192.168.10.32

Figure 8.1 Testing scenerio with subnetting

After creating simulation environment, intermediate node collects all video frames to send other nodes. Intermediate node starts to listen other nodes for request video frames. When the node wants its video frames first it sends its virtual buffer information that is overhead packets. According to all information the communication starts between the intermediate node and other nodes.

In each test our simulation has different scenario. We change virtual buffer of the each node at the beginning of the test. We change the algorithm important parameters such as depth value of NCVD algorithm.

We compare our test according to throughput, node selection, primary packet selection, complexity of algorithm.

Throughput: Both NCV and NCVD achieve higher application throughput than without network coding scheme. These two algorithms improve the throughput with a network coding part. To select the best candidate network code and send the best candidate packet to all nodes also improve the throughput. Our algorithms are fairly better than non-network coding and slightly better each other.

Node selection: NCVD performs better performance than NCV because of NCVD runs in all T_x queue. Selection of the primary packet specified the target node.

Primary packet selection: The most important part of our algorithm is to improve throughput with correct selection of the primary packet. NCVD is better than NCV with primary packet selection. NCVD achieves this with node selection and primary packet optimization. While NCV selects the first active packet as primary packet, NCVD looks in depth in the queue to select the best improvement packet as primary packet.

Complexity of algorithm: Our algorithms consider all possible candidate network codes so they are more complex than send all packets without any network coding. These two algorithms have different metrics to specify the best network code. So, the complexity of these algorithms is to consider all possible candidate codes. Besides this, NCVD creates more candidate codes than NCV because it considers entire T_x queue and selects the best code.

In conclusion, our system works efficiently for each different simulation scenario. The most important for each transmission is receiving the primary packet to target node. Although in bad scenario, our system worked successfully to receive packet.

9. CONCLUSION

In this work, we used two different network coding algorithm for video streaming. Before using network coding, we applied Multi-State Video Coding method. Our approach combines these two methods and this idea helps to increase throughput and video quality with priority of transmission. The simulation results show that using MSVC with network coding algorithm improve the video quality. We understood that, using MSVC with network coding is more effective on video quality using only network coding algorithm.

REFERENCES

About WebM. Retrieved 06/05/2013, 2013, from http://www.webmproject.org/about/

Advanced Systems Format. (15 March 2013). 2013, from http://en.wikipedia.org/wiki/Advanced_Systems_Format

Apostolopoulos, J. G. (2001). Reliable video communication over lossy packet networks using multiple state encoding and path diversity. *Visual Communications and Image Processing 2001, 4310*, 392-409.

Apostolopoulos, J. G., & Wee, S. J. (2001). Unbalanced multiple description video communication using path diversity. *2001 International Conference on Image Processing, Vol I, Proceedings*, 966-969.

Ekmekci, S., & Sikora, T. (2003). Multi-state vs. single-state video coding over error-prone channels. *Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, Vols 1 and 2*, 1544-1547.

Flierl, S. E., & Sikora, T. (2005). Multi-State Video Coding with side information. *2005 39th Asilomar Conference on Signals, Systems and Computers, Vols 1 and 2*, 874-878.

Flierl, S. E., Sikora, T., & Frossard, P. (2006). Unbalanced quantized multiple state video coding. *Eurasip Journal on Applied Signal Processing*. doi: Artn 14694 Doi 10.1155/Asp/2006/14694

Fragouli, C., Katabi, D., Markopoulou, A., Medard, M., & Rahul, H. (2007). Wireless network coding: Opportunities & challenges. *2007 Ieee Military Communications Conference, Vols 1-8*, 3363-3370.

Garrett-Glaser, J. (23/07/2010). Diary Of An x264 Developer: Announcing the world's fastest VP8 decoder.

Goyal, V. K. (2001). Multiple description coding: Compression meets the network. *Ieee Signal Processing Magazine, 18*(5), 74-93. doi: Doi 10.1109/79.952806

Information Sciences Institute University of Southern California. (September 1981). RFC:793 TRANSMISSION CONTROL PROTOCOL. from http://tools.ietf.org/html/rfc793

Katabi, D., Katti, S., Hu, W. J., Rahul, H., & Medard, M. (2006). On practical network coding for wireless environments. *2006 International Zurich*

Seminar on Communications: Access - Transmission - Networking, Proceedings, 84-85. doi: Doi 10.1109/Izs.2006.1649085

Katti, S., Katabi, D., Hu, W., Rahul, H., & Medard, M. (2005). The Importance of Being Opportunistic: Practical Network Coding for Wireless Environments.

Katti, S., Rahul, H., Hu, W., Katabi, D., Medard, M., & Crowcroft, J. (2006). XORs in the air: Practical wireless network coding. *Computer Communication Review, 36*(4), 243-254.

Katti, S., Rahul, H., Hu, W. J., Katabi, D., Medard, M., & Crowcroft, J. (2008). XORs in the air: Practical wireless network coding. *Ieee-Acm Transactions on Networking, 16*(3), 497-510. doi: Doi 10.1109/Tnet.2008.923722

Kurose, J., & Ross, K. (2003). *Computer Networking: A top-Down Approach Featuring the Internet* (Second ed.): Copyright © 2003 by Pearson Education, Inc.

Liao, Y. T., & Gibson, J. D. (2008). Refined Error Concealment for Multiple State Video Coding over Ad Hoc Networks. *2008 42nd Asilomar Conference on Signals, Systems and Computers, Vols 1-4*, 2243-2247.

Liao, Y. T., & Gibson, J. D. (2009). Rate-Distortion Based Mode Selection for Video Coding over Wireless Networks with Burst Losses. *Pv: 2009 17th International Packet Video Workshop*, 220-229.

Montpetit, M. J., & Medard, M. (2010). Video-centric Network Coding Strategies for 4G Wireless Networks: An Overview. *2010 7th Ieee Consumer Communications and Networking Conference-Ccnc 2010*, 934-938.

MPEG-4 Part 14. (2004, 17 May 2013). 2013, from http://en.wikipedia.org/wiki/MPEG-4_Part_14

Ozer, J. (2010). VP8 vs. H.264. Retrieved 07/05/2013, 2013, from www.streaminglearningcenter.com

Ramasubramonian, A. K., & Woods, J. W. (2010). Multiple Description Coding and Practical Network Coding for Video Multicast. *Ieee Signal Processing Letters, 17*(3), 265-268. doi: Doi 10.1109/Lsp.2009.2038110

Richardson, I. H.264 Advanced Video Coding. Retrieved 03/05/2013, 2013, from http://www.vcodex.com/h264.html

Robertson, M. R. H.264 Versus MPEG-4 - Video Encoding Formats Compared. Retrieved 03/05/2013, 2013, from http://www.reelseo.com/encoding-formats-mpeg4-vs-h264/#ixzz2VKNWUZQI

Schulzrinne, H. (December 1992). The Real-Time Transport Protocol.

Schulzrinne, H., Casner, S., Frederick, R., & Jacobson, V. (January 1996). RTP: A Transport Protocol for Real-Time Applications. from http://www.ietf.org/rfc/rfc1889.txt

Seferoglu, H., & Markopoulou, A. (2009). Video-Aware Opportunistic Network Coding over Wireless Networks. *Ieee Journal on Selected Areas in Communications, 27*(5), 713-728. doi: Doi 10.1109/Jsac.2009.090612

Stevens, W. (January 1997). TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms. from http://www.ietf.org/rfc/rfc2001.txt

Stevens, W. R. (1994). *TCP/IP Illustrated, Volume 1: The Protocols*: Massachusetts: Addison-Wesley.

Thomos, N., & Frossard, P. (2009). Network coding and media streaming. *Journal of Communications, 4*(9), 628-639.

Vitali, A. (October 2007). Multiple Description Coding- a new technology for video streaming over the Internet.

Wang, Y., Reibman, A. R., & Lin, S. N. (2005). Multiple description coding for video delivery. *Proceedings of the Ieee, 93*(1), 57-70. doi: Doi 10.1109/Jproc.2004.839618

Wiegand, T., Sullivan, G. J., Bjontegaard, G., & Luthra, A. (2003). Overview of the H.264/AVC video coding standard. *Ieee Transactions on Circuits and Systems for Video Technology, 13*(7), 560-576. doi: Doi 10.1109/Tcsvt.2003.815165

Zaghal, R. Y., & Khan, J. I. (2005). EFSM/SDL modeling of the original TCP standard (RFC793) and the Congestion Control Mechanism of TCP Reno.