



---

# Effect of packet loss on VP8 encoded videos using HTML5 based streaming

Seyedeh Marjaneh Ahmadi Mehri

This thesis is presented as part of Degree of  
Bachelor of Science in Electrical Engineering

Blekinge Institute of Technology

June 2014

---

**Blekinge Institute of Technology**

**School of Engineering**

**Department of Electrical Engineering**

**Supervisor: Muhammad Shahid**

**Examiner: Dr. Sven Johansson**



## Abstract

HTML5 based streaming gives opportunity to users for watching videos without using any browser plugins. HTML5 supports three different formats of video. One of them is VP8, which is owned by Google and is used in this thesis. Quality of experience of videos transmitted through the network can be affected by different network distortions. Packet loss is one of the main distortions in communication networks. In this project, the effect of packet loss on video, encoded by VP8, with different resolutions and bitrates, has been investigated using subjective tests.



### **Acknowledgments**

I would like to express my special thanks to Muhammad Shahid, my supervisor in Blekinge Institute of Technology who gave me the opportunity to make this wonderful project and supported me in my thesis with great suggestion.

I would like to thank my family and my friends, for their great help and support through all the time.



## Content

1. Introduction .....	9
1.1 Motivation .....	9
1.2 Outline of report .....	10
2. Background and related work .....	11
2.1 Related work .....	11
3. Foundation / Design and implementation .....	12
3.1 Requirements.....	12
3.2 VP8 video codec .....	14
3.3 Video format .....	15
3.4 Work design .....	19
3.4.1 Network emulator .....	19
3.4.2 Network emulator setup .....	20
3.4.3 Web design .....	22
3.5 Implementation .....	23
4. Data analysis .....	24
4.1 Calculation .....	24
4.1.1 Mean opinion score.....	24
4.1.2 Confidence interval .....	24
4.2 Screening of the subjects .....	25
5. Result and conclusion .....	26
Appendix A .....	29
Appendix B .....	33
Appendix C .....	36
References .....	37

Figure 3.1 .....	12
Figure 3.2 .....	13
Figure 3.3 .....	14
Figure 3.4 .....	14
Figure 3.5 .....	15
Figure 3.6 .....	17
Figure 3.7 .....	20
Figure 3.8 .....	21
Figure 5.1 .....	26
Figure 5.2 .....	27
Figure 5.3 .....	27
Figure 5.4 .....	28
Figure 5.5 .....	28

Table 3.1 .....	17
Table 3.2 .....	18
Table 3.3 .....	21

## 1. Introduction:

Videos are one of the most popular media, which are transmitted through the network. The usage of mobile video will increase to 69 percent between 2013 and 2018 [1]. The usage of videos is rapidly growing on the web. Watching videos on the Internet is preferred for both entertainment and education purpose. It is useful as it saves time and videos of the past can also be viewed with ease on the Internet.

Videos on HTML5 based streaming can be shown without plug-in through the browsers and other video streams require extra software like Adobe Flash Player on the system to show video. It is not only time consuming but also need to upgrade the system, which is used for watching video. HTML5 is the latest standard for HTML, which is markup language and is used for structuring and presenting content for the World Wide Web and core technology of the Internet. HTML5 arrived to reduce the necessity for external plug-in and improve the content of World Wide Web to use multimedia and graphical concept easy and convenient. HTML5 is a device-independent, which is compatible with different type of devices like computer, tablet, smart phone, and smart TV [2].

Until now HTML5 is supporting three different types of video as follows:

- i) WebM which is encoded by VP8 codec
- ii) Mp4 that is encoded by H.264 video compression
- iii) Ogg

VP8 is a one of the types of video format of video compression; it belongs to Google and is created by On2 Technologies. VP8 uses WebM as a container that is built for the web and supported by some browsers like Google, Mozilla, Opera, and etc. VP8 is based on HTML5 stream, which is not using Adobe Flash Player, and from May 2010, most of YouTube videos play in browsers that is supported HTML5 [3]. Videos that are encoded by VP8 are used for experiments in this thesis.

### 1.1 Motivation

Nowadays, many people watch TV programs, games or videos over Internet and they need good quality of experience. According to International Telecommunication Union, definition of

quality of experience is “contained degree of satisfaction by a user of service”. Therefore, all users have their own experience in different situations of services. For instance, for the one who is watching news, the voice quality is more important than the video but on the other hand, video is much crucial for the one who is watching a movie. Therefore, network performance is a big challenge today and it can be measured by bandwidth, throughput, latency, jitter and error rate. In the case of video transmission jitter, delay, and loss are parameters that should be considered. Many researches have studied in these areas. However, this experiment is also about the study of WebM format videos in different resolutions and bit rates for encoding by VP8 and is being run on web page by HTML5 on Google chrome browser. KauNet network emulator is used to apply different rates of packet loss on video stream. At the end of the experiment, user feedback, is applied by use of mean opinion score (MOS).

## **1.2 Outline of the report**

Background and related work gives history about HTML5 based streaming in chapter 2. In chapter 3, basic information about video codecs, video frames and network emulator is presented. Web design for testing HTML5 based streaming is needed that is mentioned under this chapter. Chapter 4 presents details on the analysis of the obtained data and in chapter 5 results are shown.

## 2. Background and related work

Videos are widely used for entertainment, advertisements and education purposes. People make videos and upload them on the Internet, as they need users' feedback for either improvement or to make more videos with respect to users' need in some aspects. Hence, increasing the performance in order to deliver good quality makes a challenge for network administrator. By using HTML5, web designer can put videos with different formats in video tag. Therefore only videos of WebM format, which is able to open the video format for the web, can run on the browser. Based on user's system application, different types of video could play as an example mp4 video can be run on windows media player 12 or later version. The videos, which transmitted on the web, are different from traditional broadcast and offline mediums. WebM is concentrated on the requirements of serving video through the web like low computational footprint for enabling playback on any device, simple container format, highest quality real-time video transmission, minimize codec profile, and etc [4].

Packet loss is one of the parameter that has more effect on quality of experience. Based on the subjective test, packet loss will decrease subject satisfaction of performance. The percentage of the packet loss on the same video content with different bitrates and resolutions can have varied result according to subject's opinion. During experiment the effect of different rate of packet loss on videos with different resolutions and bitrates are tested by subject's vote.

### 2.1 Related work

HTML5 and Adobe Flash use different technology to play audio and video on webpage. HTML5 requires JavaScript and cascading style sheets 3(CSS3) to be able to show animation. Since 2004, the Web Hypertext Application Technology Working Group (WHATWG) has started to work on this new standard HTML5. The first Public Working Draft of specification was produced by WHATWG on January 2008. In 2009, WHATWG has joined the World Wide Web Consortium (W3C) for developing HTML5. In 2012, they made a decision to divide the standard into two parts: specification and living standard. Specification includes syntax and formats of audio, video, image, and character encoding. The living standard is about updating and improving it forever, which means that it is not completing. W3C works on specification part and WHATWG on the living standard. The former was completed in December 2012 as HTML5.1, while the latter's release is expected by the end of 2015 [2].

Developers test new features of HTML5 and currently research is carried out on various topics applicable for this standard. There is no similar study for effect of packet loss on different resolution of WebM video in HTML5. This topic is unique and the result might help other researchers who want to investigate HTML5 based streaming.

### 3. Foundations/ Design and implementation

In this chapter, requirement and design for subjective test will be explained. Requirement part mentioned video compression aspects. It follows by VP8 video codec specification and video formats, which is used in this project. Network emulator, web design, and implementation are also presented.

#### 3.1 Requirements

Video is a sequence of individual images or frames. Raw video needs a process to reduce the number of bits, which is suitable for transmission or storage. Video compression or video coding is the process that is used for this purpose and contains two steps as it shows in figure 3.1, encoder and decoder. Encoder transforms the source video into the compressed one to decrease number of bits and decoder return compressed form into a representation of original video. These two pair encoder/decoder are mostly called CODEC [5].

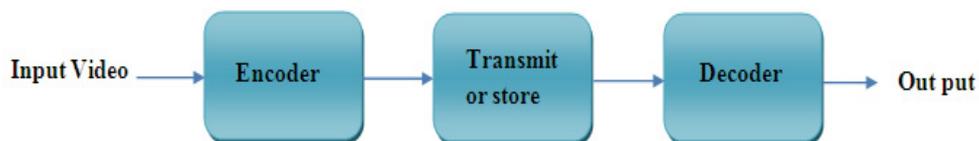


Figure 3.1 Encoder/Decoder [5]

By removing the redundancies, the data compression is achieved. Lossless compression of image and video does not give entire amount of compression. Data consisting of statistical redundancy can be subjected to lossless compression so that the decoded output has reconstructed data that is replica of original data. Higher compression ratios are possible with lossy compression compromising the visual quality. The principle of this system is to remove the subjective redundancy concerning the subject's perspective of visual quality. In this system the decoded data is not identical to the original one [5].

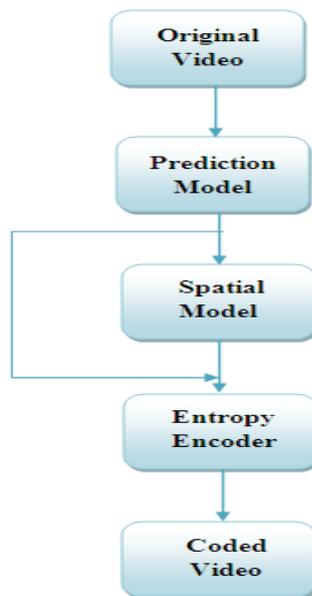


Figure 3.2 Encoder block diagram [5]

Video encoder block diagram in the figure 3.2 shows the main section of encoder that contains prediction model, spatial model, and entropy encoder and their order. Original video breaks down into small block or frame, which is going to prediction model. Prediction model decreases redundancy with using similarity between neighboring video frame or block. These are two ways for applying prediction model, intra prediction that predicts from neighboring image samples, and inter which compensating for differences between the frames. Residual frame, which is the output of prediction model, is obtained by subtraction the prediction from the actual frame. The out put of prediction model, the residual frame, goes to spatial model that take help of similarities between local samples in the residual frame to decrease redundancy of spatial. Entropy encoder makes compressed file or bit stream that can transmit or store.

Compressed file or bit stream passes through the decoder in order to reconstruct the video. Decoder block diagram is shown in a figure 3.3, which is inversed of encoder process that is explained in previous section. Decoder algorithm that may be lossy to perfectly reconstruct the original from transform coefficient.

Video compression algorithms perform vast number of transforms and inverse transforms per second according to video's resolution and frame rate per second. In old video codecs inverse computation takes as much as 30% of the process cycle while the new ones such as VP8 and H.264 take only a few percentages of the decoder cycles [5].

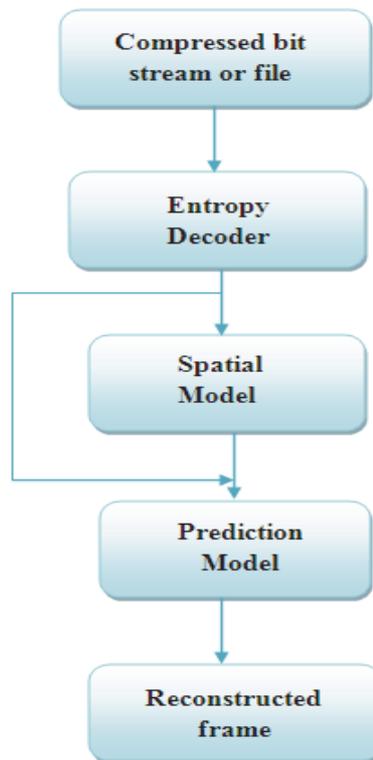


Figure 3.3 Decoder block diagram [5]

### 3.2 VP8 video codec

VP8 is video compression format that is studied in this thesis. It uses WebM as an audio and video container format that is designed to provide open video compression for HTML5 based video streaming.

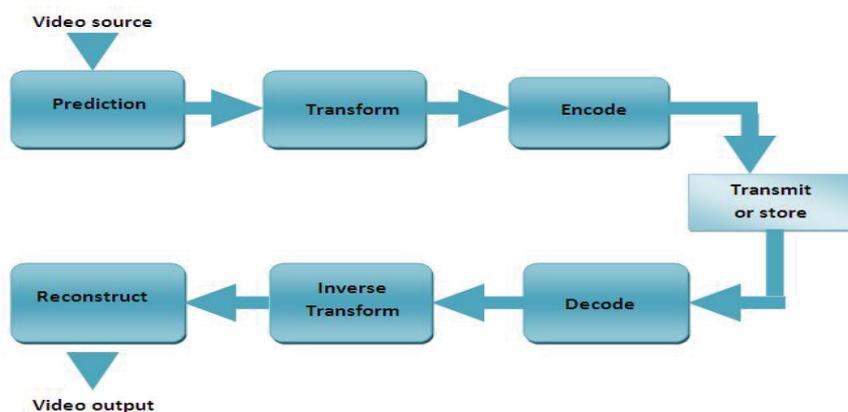


Figure 3.4 VP8 block diagram

VP8 is a block-based codec that divides the frame to smaller segments called macroblock. For each macroblock the encoder does the prediction of redundant motion and color information based on blocks previously processed. For more efficient compression, the redundant data can be subtracted from the block [6]. The algorithm of VP8 is shown in the figure 3.4, which is obtained from encoder process in first part, and it follows by decoder process. VP8 has two class of prediction: Intra prediction that uses data from each video frame and Inter prediction that takes data from previous encoded frame. Each prediction class has its own mode to be used in macroblock. The next step as shown in figure 3.4 is transformation that has own way for encoding. After encoding video, it will be stored in the buffer of system or transmitted for next step. (Deep explanation of VP8 algorithm is out of scope in this thesis report).

VP8 is supported by almost all browsers such as Google-Chrome, Opera, Chromium, Android browser, Firefox, Konqueror and new version of Internet Explorer (version 9 or later). Google chrome is chosen for this research, as it is one of the most popular browsers all over the world.

### 3.3 Video format

Video coding is a technique of compressing and decompressing a digital video signal. Natural or real world video scenes of several items are collected each with their unique characteristic of shape, depth, texture and illumination. The characteristics of natural video are related to video processing and condensing involved spatial characteristics like variations of texture in the scene, the shape, color and number of objects and other temporal characteristics like motion of object, illumination changes and the viewpoint of the camera. The sample of a real scene is rectangular grid in video image plane spatially and as a series of still frames i.e., components of frames sampled at regular intervals of time temporally (Figure 3.5) [5]. Each sample illustrates one or more numbers, which express the brightness or luminance and the color of sample.

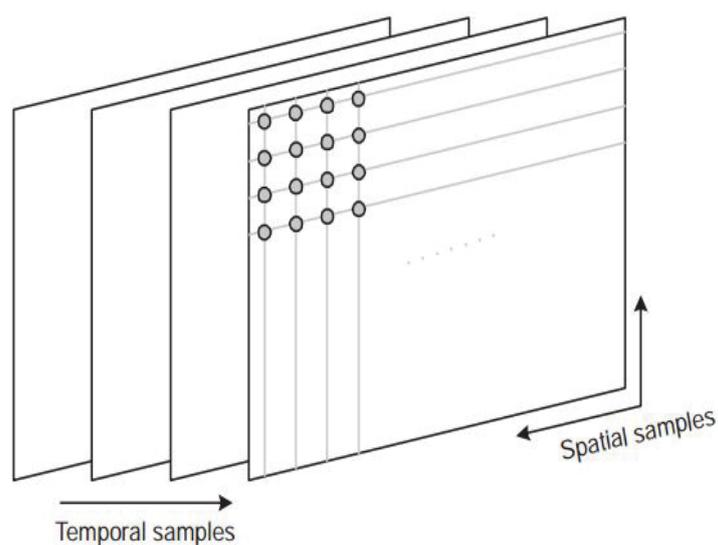


Figure 3.5 Spatial and temporal sampling of a video sequence [5]

Digital video applications depend on the exhibition of color video and therefore it requires the process to capture color information. A monochrome image needs only one number to specify the brightness or luminance of each spatial sample. Color image, entails at least three numbers per pixel position to constitute the precise color. The mechanism that is used for representing brightness, luminance or luma and color is called color space. In the RGB color space, a sample has three numbers that show the corresponding colors Red, Green and Blue. Other colors can be derived by the combination of these three colors. RGB image is captured with each color in separate array. In other words, it displays images with individual illuminations of red, green and blue constituents per pixel with regard to the strength of each item.

Luminance has more influence on perception of Human Visual System (HVS) rather than that of color. In RGB color space, these three colors will be saved in same resolution. However, for increasing efficiency of color image, separate luminance from the color data and the luma is saved in higher resolution than the resolution of color.

The Y:Cr:Cb color space is the most popular way for color image. Y is the luma factor, which is obtained by weighted average of R, G, and B.

$Y = k_r R + k_g G + k_b B$ , where k is weighted factor

Also color information can be shown by color differences or chrominance elements, which are variations of R, G, B and luminance Y.

$$Cr = R - Y$$

$$Cb = B - Y$$

$$Cg = G - Y$$

As it shows, Y:Cr:Cb color space has four items unlike RGB. But chrominance components are constant and if two of them are known, the third one can be calculated from others. In Y:Cr:Cb color space, just luma(Y), red and blue chroma (Cr,Cb) are carried out. This method decreases the amount of data required to constitute chroma component without clear outcome on visual quality. This is the advantage of Y:Cr:Cb over RGB [5].

### **Standard Definition (SD)**

In standard definition, Luminance and chrominance are sampled at 13.3 and 6.75 MHz respectively to build a “4:4:4”Y:Cr:Cb component signal. The specified parameters of the sampled digital signal rely on the frequency of video frame- 30Hz for an NTSC signal and 25Hz for a PAL/SECAM signal as shown in the table 3.1. The higher frame rate of NTSC (30Hz) is recompensed by a lower spatial resolution so that the total bit rate is alike in each case- 216Mbps. The active area, which is shown on the display, is smaller than the total area because some horizontal and vertical intervals are outside of the edge of the frame [5].

Format	Luminance resolution (horiz.×ver.)	Bits per frame (4:2:0,8 bits per sample)
Sub-QCIF	128×96	147456
Quarter CIF (QCIF)	176 ×144	304128
CIF	352× 288	1216512
4CIF	704×576	4866048

Table 3.1 “Video frame formats”[5]

The possible range of each sample is 0 to 255. 0 and 255 are reserved for synchronization and the active luminance signal is restricted to a range of 16 (black) to 235(white).[5]



Figure 3.6 “Video frame sampled at range of resolutions”[5]

## High Definition (HD)

High definition video refers to higher resolution than that of standard definition. There is no specific definition for HD, but main phenomenon is that the video image with the horizontal lines larger than 480 in North America or 576 in Europe are evaluated as high definition. The specification from ITU-R BT.601-5 is listed in the table 3.2.

Parameter	30Hz frame rate	25Hz frame rate
Fields per second	60	50
Lines per complete frame	525	625
Luminance samples per line	858	864
Chrominance samples per line	429	432
Bits per sample	8	8
Total bit rate	216 Mbps	216 Mbps
Active lines per frame	480	576
Active samples per line (Y)	720	720
Active samples per line (Cr,Cb)	360	360

Table 3.2 Specification list from ITU-R BT.601-5 [9]

Obviously HD format needs larger storage than SD format and therefore it is vital to compress video for practical application.

## 3.4 Work design

### 3.4.1 Network emulator

Effects of packet loss on video are examined in this study. Packet loss happens when one or some packets of data travelling across a computer network fail to reach their destination. This is one of the main error types run into digital communications. This is caused by few factors like degradation of signal through the network medium due to multi-path fading, dropping of packets due to channel congestion, rejection of corrupted packets in-transit, faulty networking hardware, faulty network drivers, or normal routing sequence. Packet loss can be intentional due to network dissuasion technique for operational management purposes [3]. The packet loss acceptable rate depends on the data type that is sent. For example, in Voice over IP traffic, losing one or two packets per minute has no effect on quality of conversation but in the case of web page, a single dropped packet can result in missing part of the file. In general with TCP/IP protocols a packet loss less than 0.1% (1 loss packet per 1000 packets) could be tolerated and a higher rate will have more effect depending upon the type of services. Three different rates of packet loss are studied containing 0.5%, 1% and 2% that is applied by KauNet network emulator.

For the evaluation and examination of the response and performance of applications and transport layer protocols, network emulation is commonly used. Being able to study a wide variety of real implementations of the applications/protocols is an edge to emulation, which is not the case for simulations. KauNet emulation system has the advantage of being able to provide the feasibility to perform network emulation accompanied by wide degree of control and repeatability [7].

In the figure 3.7, a typical network emulator is shown. The Kaunet emulators run in machine in the middle to desire network or link characteristic. The application or protocol that will be evaluated are installed on host A and host B computers act as an end node. Emulator machine should have an operating system that can support Kaunet (Linux distribution such as Ubuntu). Host A and B can have any operating system and they should be in different subnets to route traffic through the network emulator.

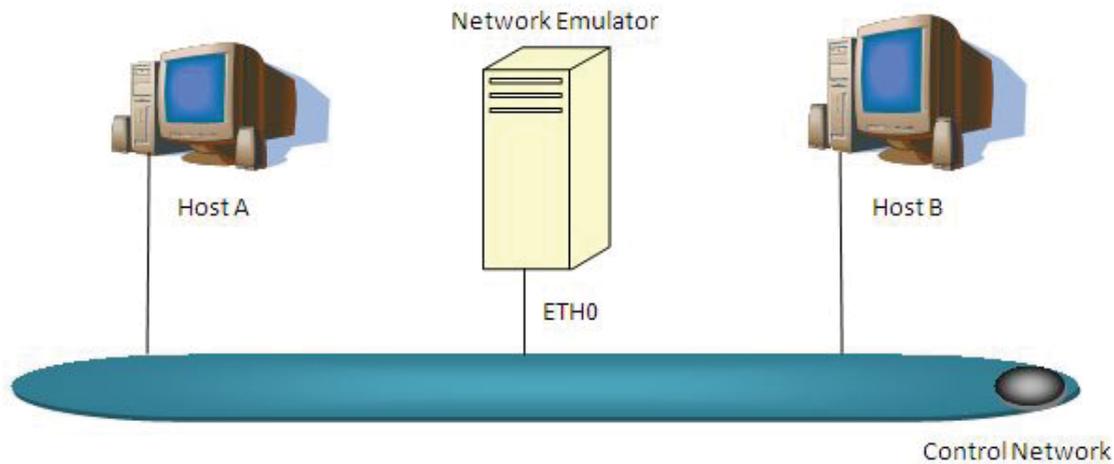


Figure 3.7 Typical network emulator

KauNet is designed by use of the well-known DummyNet emulator and some of extended patterns to improve better control over network. DummyNet has the ability of applying bandwidth restrictions, delay to the packets and packet loss, also desired link or network conditions. KauNet has some new features added, like bit-errors, reordering packets and triggers for sending information from emulator to outside observers. In other words, KauNet permits bit-errors, packet losses, packet reordering, triggers, delay and bandwidth variations to be controlled absolutely and reproducibly on a per-packet or per-millisecond basis using patterns [7].

### 3.4.2 Network emulator setup

In this study, packet loss pattern is used, by specifying the packets that must be lost in uniformly distributed random losses. Packet loss is applied in the dummyNet input/output function (`io()`) when the packet is received by the dummyNet layer for a specific pipe. This occurs before any queuing or bandwidth restrictions are applied or delays have been assigned to the packet.

Note that before applying emulator the pattern that will control network behavior should be created and it must be loaded into the emulator machine. Figure 3.8 shows the scenario that is used in this paper. As you see network emulator is running on the machine as a shaper to control all traffic that is coming from server to client computer.

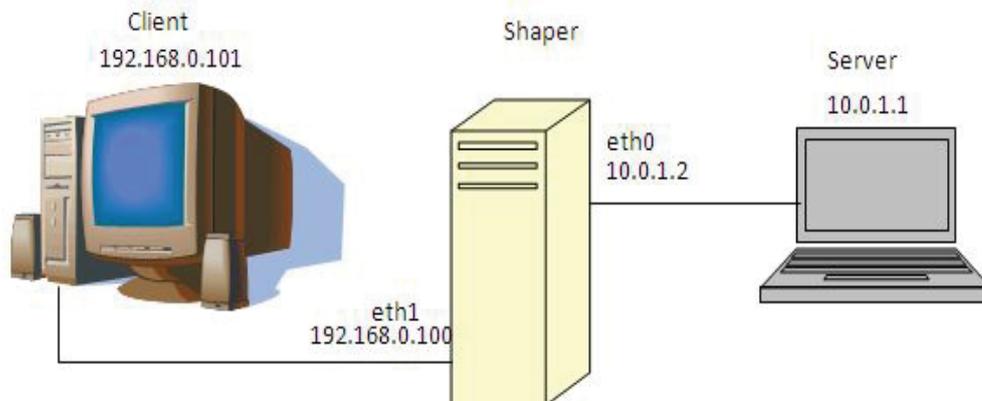


Figure 3.8 Network scenarios

Shaper has two interfaces. One of them is connected to the server and the other one connected to the client. In this scenario, shaper makes a pipe between interface eth1 and client machine and puts all traffic in the network 192.168.0.0/24 through the pipe. (Note: pipe is a logical connection between source and destination.) Executing the command below makes pattern:

```
home/KauNet/KauNet2.0Linux/patt_gen#./patt_gen -pkt -rand test1.plp data 5000 200 $1
```

The first part until # sign is the directory that Kaunet is active on. For writing command we should go through patt\_gen command line tool that is developed to create and manage patterns. The tool could generate patterns regarding to several parameters distributions. The syntax is illustrated in table 3.3. The value here is dependent on experiment setup.

Syntax	Explanation
-pkt	Packet loss
-rand	Uniform distributed
test1.plp	Pattern is saved in this file
5000	Number of packet
200	Random seed
\$1	Percentage of packet loss

Table 3.3 Illustrate syntax

```
home/KauNet/KauNet2.0Linux/patt_gen#ipfw -f flush
```

```
home/KauNet/KauNet2.0Linux/patt_gen#ipfw -f pipe flush
```

```
home/KauNet/KauNet2.0Linux/patt_gen#ipfw add allow all from any to any
```

```
home/KauNet/KauNet2.0Linux/patt_gen#ipfw add 1 pipe 100 ip from 192.168.0.101 to 10.0.1.1  
in
```

```
home/KauNet/KauNet2.0Linux/patt_gen#ipfw pipe 100 configbw 10Mbit/s pattern test1.plp
```

The first command removes all previous rules on the firewall and the second one omits the pipe, which is made between server and client through the shaper. The third line gives permission for all traffic. The fourth line makes a pipe 100 between client and server and applies rule to packets that come from server to client. The last line determines bandwidth for pipe 100.

Percentage changing of packet loss for each video randomly can be done using PHP code. Three rates of packet loss 0.5, 1 and 2 percent are applied by shuffling this value 12 times and the number is saved in text format that can be read by shaper program to assign value for \$1 in pattern.

### 3.4.3 Web design

Website is designed by the use of HTML5 header and features. The HTML5 sample code can be seen in Appendix A.

Two types of video content with two different resolutions (SD,HD) and three different bit rates 2,3 and 4 Mbps are planned to be tested and examined as to which of them is sensitive to packet loss from the user or subject viewpoint. Hence, twelve videos should be displayed to each subject. For finding better result, the packet loss rate to the videos shall be applied randomly. The order of video may affect user's score and therefore videos' order should be changed for different users. For this purpose, twelve videos are divided to two groups based on video resolution (SD, HD). Two websites with names videoqoe1 and videoqoe2 are created on the server. Each website consists of 6 pages containing one of the videos each.

A random number for 41 subjects are composed utilizing "rand" function in matlab and the random value is saved in text file which in turn is used to decide the website to be displayed first. With regard to exhibiting both websites to users, after the vision of first 6 videos, a 'continue' page is displayed asking the user to 'click' in order to continue. After the click, user is directed to the second website.

To obtain distinct order of video, PHP code places the page number in the shuffle and makes an order of webpage for each user. Random generator produces 125 different orders of session from the seed number that is chosen. There exists no duplication for 41 subjects.

Shaper should be called for applying packet loss rate concerning each session (each page of website). As explained in the section 3.3.2, the loss is applied on packet travelling from server to client. There are three different rates of packet loss that are to be applied randomly. PHP code helps call shaper and random generator is needed to generate random packet loss rates for each session and keep track of it. This value is saved in a table in database.

As this experiment is user based, quality score on each page of web given by user is recorded in database on client machine. All this data is helpful in analysis as mentioned in section 4. Saving of data on client machine is achieved by the usage of WAMP server v2.4. In mysql table, Information like user ID, session ID, rate of packet loss and quality score are saved.

PHP code for saving user information, user's score and page of continuation is run on client machine and is presented in appendix B. PHP code for control shaping is executed on the shaper machine and is under appendix C. The codes are mentioned under appendices to help enhance better understanding of work.

During the experiment, Google chrome's incognito windows are used for browsing to help make browser history free from videos shown to previous user. Incognito window does not save files that were opened or downloaded in browser or download history.

### 3.5 Implementation

When user arrives to participate in the experiment, experimenter running the experiment should explain the task to the user. There is a checklist referring the explanation to user before conducting the experiment.

- This experiment includes two parts of video presentations with different resolution. Each of them contains 6 videos and packet loss is applied to every video randomly. The first page contains user information form with the columns of name, age, email address and gender to be filled in by the user. Succeeding the form, the first webpage displayed includes first six videos followed by the occurrence of continuation page asking to press 'continue' button to carry on further with the experiment until the last page, thanking for the participation, appears.
- During experiment please do not stop or replay video but just let the videos show in their order.
- After the termination of each video, rate the video by providing score according to the quality of video seen.
- Next video appears after pushing next bottom on the page.
- The first page, start page is on the WAMP server on client machine but after the user clicks on ok button it goes to real server to show website content (video) but user is unaware of the server being changed because of PHP program. Also, the continue page is on the WAMP server on client machine. With programming the change of server has no effect on quality of experience.

## 4. Data analysis

Data from performance of experiment is collected in database. The data, in the form of observers' score should be summarized by statistical techniques to contribute to result in graphical form. For that purpose Rec. ITU-R BT.500-12 Annex 2 is used. Score value is an integer between 1 and 5. It has some variations based on observers' judgment on various conditions of the experiment for example, different rates of packet loss or different resolutions of video.

### 4.1 Calculation

Information from database contains number of presentations, L. Each presentation is one of test conditions, J that apply to one of the number of sequences, K. Here, test condition is rate of packet loss and number of sequences is the amount of video. In this study, combination of test conditions and sequence conditions is not repeated.

#### 4.1.1 Mean scores

Calculation of mean is the first step in analyzing the result of test. Mean of each presentation is calculated by the formula:

$$\bar{u}_{jk} = \frac{1}{N} \sum_{i=1}^N u_{ijk}$$

Where, 'u<sub>ijk</sub>' is the score of observer 'i' for test condition 'j' and sequence 'k'. N denotes the number of observers.

#### 4.1.2 Confidence interval

Confidence interval is an estimate interval that shows the limitation of real value. It is applied to exhibit reliability of experiment. There are various levels of confidence but most often 95% confidence interval is used and is suggested by Rec.ITU-R BT.500-12.[8]

95% confidence interval is [  $\bar{u}_{jk} - \delta_{jk}$  ,  $\bar{u}_{jk} + \delta_{jk}$  ]

Where:

$\delta_{jk} = 1.96 \frac{S_{jk}}{\sqrt{N}}$  and S<sub>jk</sub> is standard deviation for each presentation that is given by formula below

$$S_{jk} = \sqrt{\sum_{i=1}^N \frac{(\bar{u}_{ijk} - u_{ijk})^2}{(N-1)}}$$

## 4.2 Screening of the subjects

To discover if the distribution of scores for presentation in experiment is normal or not, Kurtosis coefficient ( $\beta_2$ ), proportionate relation of the forth order moment and the second order moment's squares to be calculated [8]. If  $\beta_2$  is between 2 and 4, distribution may be given as normal. Then, the process will continue in mathematical way to find out which observer's vote should be discarded in comparison to average operation.

$$\beta_{2jkl} = \frac{m_4}{(m_2)^2} \quad \text{with} \quad m_x = \frac{\sum_{n=1}^N (u_{njkl} - \bar{u})^x}{N}$$

For each observer,  $P_i$  and  $Q_i$  are calculated and these algorithms should be applied for overall presentation.

If,  $2 \leq \beta_{2jkl} \leq 4$

If  $u_{njkl} \geq \bar{u}_{jkl} + 2S_{jkl}$  then  $P_i = P_i + 1$

If  $u_{njkl} \leq \bar{u}_{jkl} - 2S_{jkl}$  then  $Q_i = Q_i + 1$

else:

If  $u_{njkl} \geq \bar{u}_{jkl} + \sqrt{20} S_{jkl}$  then  $P_i = P_i + 1$

If  $u_{njkl} \leq \bar{u}_{jkl} - \sqrt{20} S_{jkl}$  then  $Q_i = Q_i + 1$

If  $\frac{P_i}{J.K.L} > 0.05$  or  $\frac{Q_i}{J.K.L} > 0.3$  then reject observer i

Where:

N: number of observers

J: number of test combination of test condition and sequence

K: number of test condition

L: number of presentation

## 5. Result and conclusion

Two different video contents are tested in this thesis. Figure 5.1 shows the picture from video F that is presentation of electrical equipment. Figure 5.2 shows the picture from video T that is rugby games. Each video content encoded in two resolutions SD and HD that are presented in chapter 3.3 and three different bit rates. Therefore bit rates and contents are the differences between 12 videos, which are transmitted through the web page during experiment. Every user watches twelve videos with different rate of packet loss, which are applied randomly from network emulator. Users quality score for each video is saved in a database and it is imported to the excel sheet. Data analysis is done according to chapter 4. The data from user is presented in the figure 5.3 to figure 5.5. Obviously in both the video content, video T and video F, when the resolution and bit rate increase the quality of experience decreases. In other words, video with high resolution and higher bit rate value is more sensitive than the video with lower resolution and lower bit rate regarding packet loss. In normal situation without applying packet loss, the video with higher bit rate has a better quality score in the subjective test [17] but after applying packet loss, which is tested in this thesis, the result is the opposite. Video content is one of the factors that have affected the subject. In the case of video T, which shows a game, even the loss of 1 packet has bigger effect on the observer when compared to video F, which shows a presentation with not much motion.

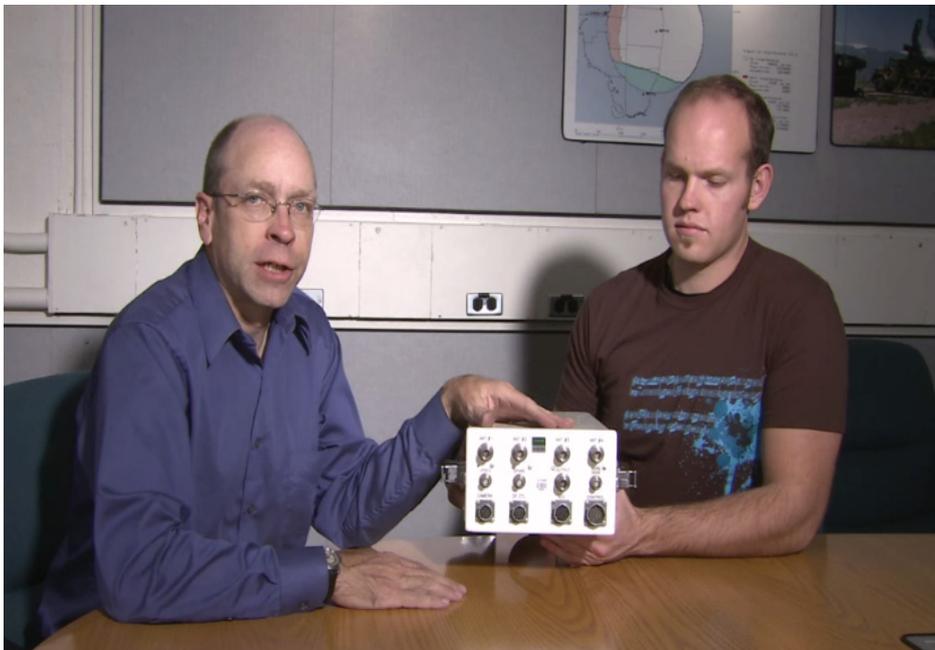


Figure 5.1 picture of video F

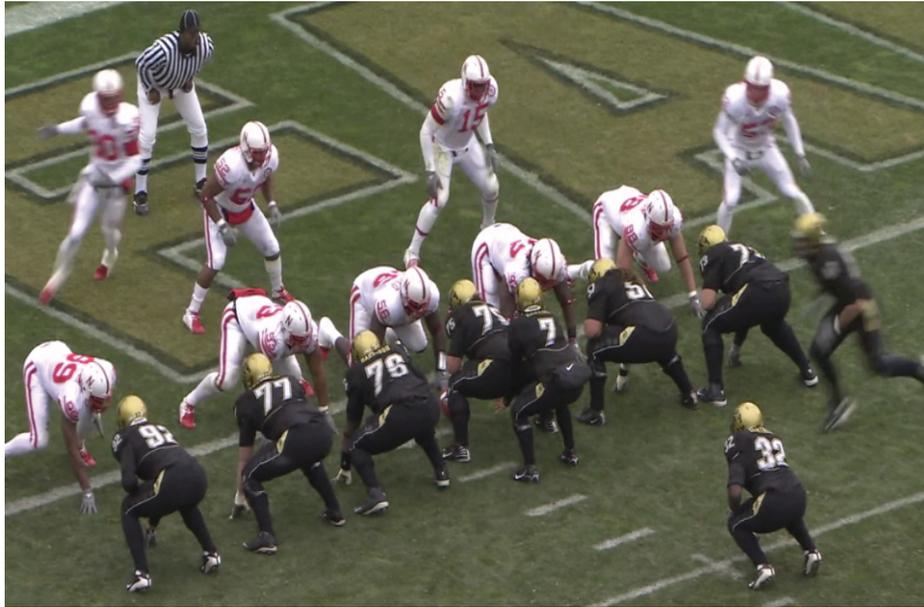


Figure 5.2 picture of video T

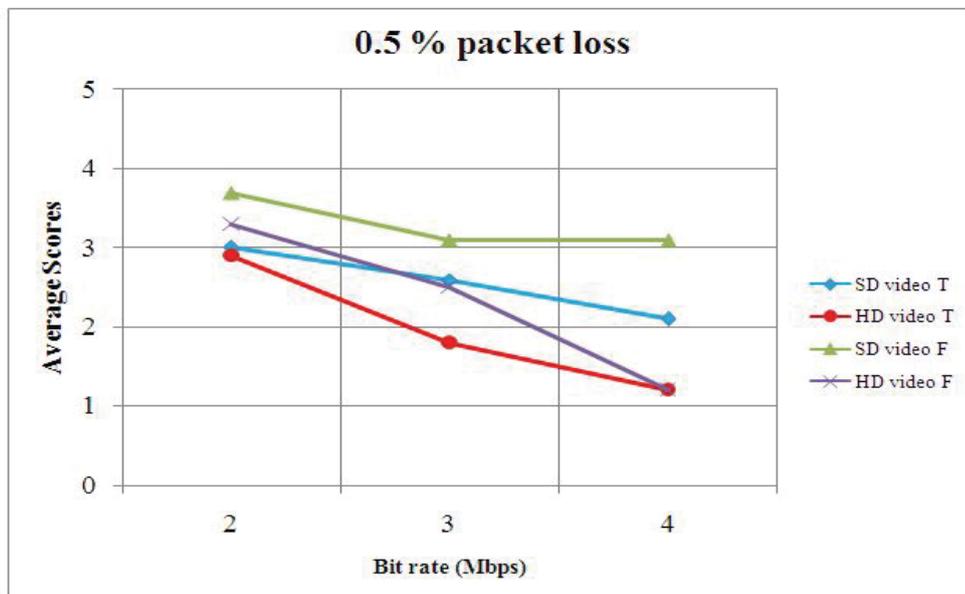


Figure 5.3 Videos with 0.5% packet loss

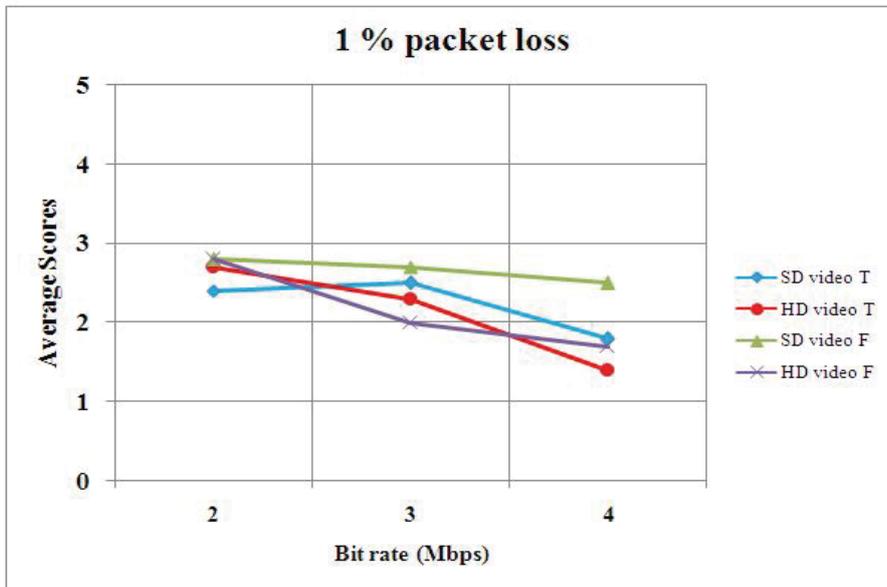


Figure 5.4 Videos with 1% packet loss

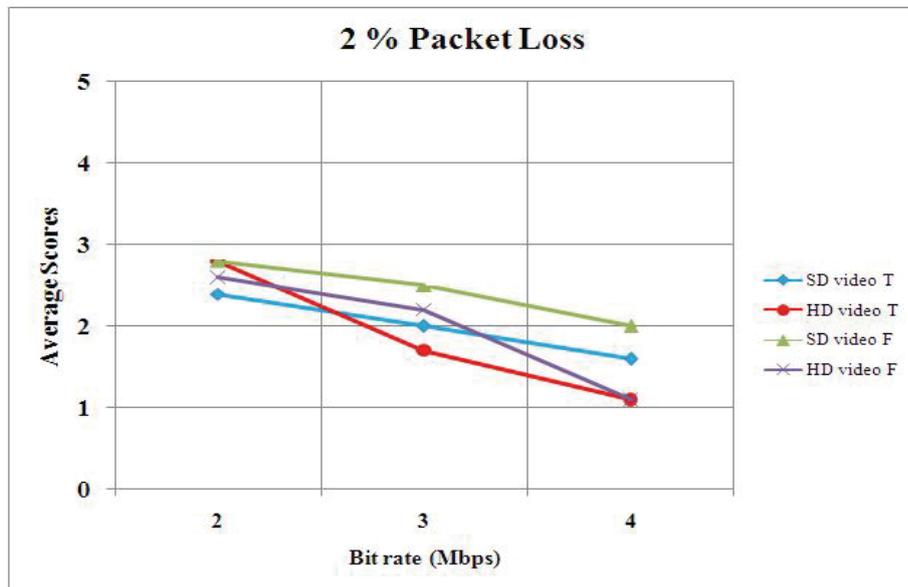


Figure 5.5 Videos with 2% packet loss

## Appendix A

**Index code (webpage):**This is code for webpage. All pages have a same code but the video name is different for each.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8"/>
<title>Video codecs test by HTML5</title>
<linkhref="style1.css" rel="stylesheet" type="text/css"/>
<script language="JavaScript" type="text/javascript" src="js/jquery-1.8.3.min.js"></script>
<script language="JavaScript" type="text/javascript" src="js/score.js"></script>
<meta http-equiv='cache-control' content='no-cache'>
<meta http-equiv='expires' content='0'>
<meta http-equiv='pragma' content='no-cache'>
</head>
<body>
<input type="hidden" id="page_id" name="page_id" value="1" />
<?php
include 'page_navigation.php';
randomGen(1,5,5);
$sess_id = getSessId();
$next_page_id = getNextPageld();
if($next_page_id != 'log_out.php'){
    $url = "index_".$next_page_id.".php";
} else {
    $url = "http://localhost/videoqoe/continue.php";
}
?>
<input type="hidden" id="sess_id" name="sess_id" value= <?php echo $sess_id;?> />
<h1></h1>
```

```

<p>
<div id="video"><video controls="controls" autoplay>
<sourcesrc="frontend720_2M.webm" type="video/webm"/>
    Your browser does not support the video tag.
</video></div></p>
<a href=?php echo $url;?>>Next</a>
<div id="score">
<center>
<p>What is your idea about video's quality?</p>
    <input type="radio" id = "idx_score" name = "score" value="5"/><span>Excellent (5)</span>
    <input type="radio" id = "idx_score" name = "score" value="4" /><span>Good (4)</span>
    <input type="radio" id = "idx_score" name = "score" value="3" /><span>Fair (3)</span>
    <input type="radio" id = "idx_score" name = "score" value="2" /><span>Poor (2)</span>
    <input type="radio" id = "idx_score" name = "score" value="1"/><span>Bad (1)</span>
<input TYPE="button" id = "score_submit" name = "score_submit" value="submit" />
</center>
</div>
</body>
</html>

```

**Page navigation code:** This is a php code for making different order of webpage to show to the users.

```

<?php
// $next_page_id = getNextPageId();
function randomGen($min, $max, $quantity) {
    $numbers = range($min, $max);
    shuffle($numbers);
    $page_arays = array_slice($numbers, 0, $quantity);
    file_put_contents("data/page_info.txt", "");
    foreach($page_arays as $line){
        $fh1 = fopen('data/'. "page_info.txt", "a+");
        fwrite($fh1, $line. "\n");
        fclose($fh1);
    }
}

```

```

    }
    $fh2 = fopen('data/'. "page_sess.txt", "w+");
        fwrite($fh2,1);
        fclose($fh2);
    $fh1 = fopen('data/'. "sess_info.txt", "w+");
        fwrite($fh1,1);
        fclose($fh1);
    return null;
}
functiongetNextPageId() {
    $page_info = file('data/'. "page_info.txt");
    $page_count_temp = file('data/'. "page_sess.txt");
    $page_counts = array();
    foreach($page_count_temp as $line){
        array_push($page_counts,trim($line));
    }
    $page_count = (int) trim($page_counts[0]);
    $page_infos = array();
    foreach($page_info as $line){
        array_push($page_infos,trim($line));
    }
    if($page_count != 6){
        $id = $page_infos[$page_count-1];
        $next_id = $page_count + 1;
        $fh1 = fopen('data/'. "page_sess.txt", "w");
        fwrite($fh1,$next_id."\n");
        fclose($fh1);
        return $id;
    } else {
        $fh1 = fopen('data/'. "page_sess.txt", "w+");
        fwrite($fh1,1);
        fclose($fh1);
        return "log_out.php";
    }
}

```

```

    }
}
functiongetSessId(){
    $page_count_temp = file('data/.'."sess_info.txt");
    $page_counts = array();
    foreach($page_count_temp as $line){
        array_push($page_counts,trim($line));
    }
    $page_count = (int) trim($page_counts[0]);
    if($page_count>= 6){
        $fh1 = fopen('data/.'."sess_info.txt", "w+");
        fwrite($fh1,1);
        fclose($fh1);
    } else {
        $next_id = $page_count + 1;
        $fh1 = fopen('data/.'."sess_info.txt", "w");
        fwrite($fh1,$next_id."\n");
        fclose($fh1);
    }
return $page_count;
}
?>

```

## Appendix B

**Index code:** This is a code that runs on a WAMP server on the client machine. It takes user information.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Video-QoE User Experiment</title>
<link rel="shortcut icon" type="img/x-icon" href="images/favicon.ico" />
<link rel="stylesheet" type="text/css" href="css/style.css" />
</head>
<body>
<h1>Welcome to the Video-QoE User Experiment</h1>
<form action="client_controller.php" method="POST">
<table class="forms" border="0" width="100%" cellpadding="0" cellspacing="0" align="center">
<tr>
<td colspan="2">Please fill out the form below</td>
</tr>
<tr>
<td width="47%">&nbsp;&nbsp;&nbsp;</td>
<td width="53%">&nbsp;&nbsp;&nbsp;</td>
</tr>
<tr>
<td>First Name:</td>
<td><input type="text" id="first_name" name="first_name"/></td>
</tr>
<tr>
<td>Last Name:</td>
<td><input type="text" id="last_name" name="last_name"/></td>
</tr>
<tr>
```

```

<td>Email:</td>
<td><input type="text" id="email" name="email"/></td>
</tr>
<tr>
<td>Age:</td>
<td><input type="text" id="age" name="age"/></td>
</tr>
<tr>
<td>Sex:</td>
<td><input type="radio" id="male" name="sex" value="Male"/><span>Male</span>
      <input type="radio" id="female" name="sex" value="Female"/><span>Female</span></td>
</tr>
<tr>
<td>What is your experience level in web browsing?</td>
<td><input type="radio" id="expert" name="site_type" value="Expert"/><span>Expert</span>
      <input type="radio" id="medium" name="site_type" value="Medium"/><span>Medium</span>
<input type="radio" id="naive" name="site_type" value="Naive"/><span>Naive</span></td>
</tr>
<tr>
<td>How often do you do web browsing?</td>
<td><input type="radio" id="day" name="visit_time" value="Every Day"/><span>Every Day</span>
      <input type="radio" id="week" name="visit_time" value="Once a Week"/><span>Once a Week</span>
<input type="radio" id="month" name="visit_time" value="Once a Month"/><span>Once a Month</span></td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
<tr>
<td>&nbsp;</td>
<td><input type="submit" class="buttons" value="Sign Up" /></td>
</tr>
</table>

```

```
</form>
```

```
</body>
```

```
</html>
```

**PHP code:**This code is used to save user score on the database on WAMP server on the client machine.

```
<?php
```

```
$score          = $_REQUEST["score"];
$page_url       = $_REQUEST["page_url"];
$page_id        = $_REQUEST["page_id"];
$sess_id        = $_REQUEST["sess_id"];
```

```
$date_created = date("Y-m-d H:i:s");
```

```
// storesession_info
$session_array = array();
$file = file("user_session.txt");
foreach($file as $line){
    array_push($session_array,$line);
}
$user_id = $session_array[0];
```

```
$shapping_array = array();
$file = file("shaping_info.txt");
foreach($file as $line){
    array_push($shapping_array,$line);
}
$shaping_model = $shapping_array[$sess_id - 1];
$sql_score      = "INSERT INTO mos_score (score,
page_url,user_id,page_id,date_created,session_id,shapping_model_id)
VALUES
($score,'$page_url','$user_id,$page_id','$date_created',$sess_id,'$shapping_model')";
```

```
$con1 = mysql_connect("localhost","root","",true);
if (!$con1)
{
    die('Could not connect: ' . mysql_error());
}
mysql_select_db("videoqoe", $con1);
$query_score = mysql_query($sql_score,$con1);
$score_id = mysql_insert_id();
```

```
echojson_encode($score_id);
```

```
?>
```

## Appendix C

**PHP code:** This code is run on shaper machine for changing the percentage of packet loss randomly.

```
<?php
functionconnection_close()
{
while (ob_get_level()) ob_end_clean();
header('Connection: close');
ignore_user_abort(1);
ob_start();
}
/**
 * Flush a closed HTTP connection from client and clean server context
 */
functionconnection_flush()
{
    $size = ob_get_length();
header("Content-Length: $size");
ob_end_flush();
flush();
ob_clean();
}

echo "start time".time()."<br>";

/*if(exec('echo EXEC') == 'EXEC'){
echo "Exec works";
}
*/

#$file_name = "test123.txt";
#echo $sess."<br>";
connection_close();
connection_flush();

$sess_id = $_REQUEST["sess_id"];
$shape_array = array();
$file = file("shapping_data.txt");
foreach($file as $line){
    array_push($shape_array,$line);
}
$exec = exec('sudo /home/videoqoe/KauNet/patt_gen/video_pipe.sh
/home/videoqoe/KauNet/patt_gen/temp.plp ' . $shape_array[$sess_id - 1]);
```

```
echojson_encode("status:Sucess");  
?>
```

## References:

- [1] [http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/VNI-Forecast\\_QA.html/index.html](http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/VNI-Forecast_QA.html/index.html)[Accessed on 23 April 2014]
- [2] [http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)[Accessed on 23 April 2014]
- [3] <http://www.webmproject.org/about/faq> [Accessed on 23 April 2014]
- [4] <http://www.webmproject.org/about/>[Accessed on 23 April 2014]
- [5] Iain E. Richardson Vcodex Limited, UK “THE H.264 ADVANCED VIDEO COMPRESSION STANDARD”, Second Edition
- [6] <http://blog.webmproject.org/2010/07/inside-webm-technology-vp8-intra-and.html> [Accessed on 20 December 2013]
- [7] Johan Garcia, Per Hurtig, Anna Brunstrom, Tomas Hall “KauNet 2.0: Design and Usage”, October 12, 2010, p5,6
- [8] Rec.ITU-R BT.500-12
- [9] Rec. ITU-R BT.601-5STUDIO ENCODING PARAMETERS OF DIGITAL TELEVISION FOR STANDARD 4:3 AND WIDE-SCREEN 16:9 ASPECT RATIOS
- [10] Iain Richardson ,Vcodex.com White Paper: “An overview of H.264 Advanced Video Coding “, 2007-2013
- [11] Christian Feller, Juergen Wuenschmann, Thorsten Roll and Albrecht Rothermel Ulm University “The VP8 Video Codec - Overview and Comparison to H.264/AVC”, 2011 IEEE International Conference on Consumer Electronics - Berlin (ICCE-Berlin)
- [12] Mu Mu, Roswitha Gostner, Andreas Mauthe, Gareth Tyson, Francisco Garcia,” Visibility of individual packet loss on H.264 encoded video stream – A user study on the impact of packet loss on perceived video quality”, Computing Department, Lancaster University, United Kingdom ,Agilent Laboratories, Edinburgh, United Kingdom
- [13] "Subjective assessment of standard definition digital television (SDTV) systems," ITU-R BT.1129-2, 1998
- [14] Zhang Hushi.”A Comparison of HTML4 and HTML5 with Respect to Server Response Time and Page Load Time”, Blekinge Institute of Technology, Master Thesis Electrical Engineering June 2012

[15] Saroj Sharan Regmi, Suyog Man Singh Adhikari, "Network Performance of HTML5 Web Application in Smartphone", Blekinge Institute of Technology, Master Thesis Electrical Engineering Nov2011

[16] OYEKANLU Emmanuel Adebomi, JOHN Samson Mwela, "Impact of Packet Loss on the Quality of Video Stream Transmission", Blekinge Institute of Technology, Master Thesis Electrical Engineering May 2010

[17] Decebal Constantin Mocanu, Antonio Liotta, Arianna Ricci, Maria Torres Vega and Georgios Exarchakos, "When does lower bitrate give higher quality in modern video services?", Department of Electrical Engineering Eindhoven University of Technology and Polytechnic University of Bari