

Evaluation and simulation of Video using EvalVid Tool

V.Sowmya Devi¹, Nagaratna P Hegde²

¹Dept of CSE, GITAM University, Hyderabad, India.

²Dept. of CSE, Vasavi College of Engineering, Hyderabad, India.

Abstract - Multimedia applications play very important role in today's world. Many applications are developed over wireless network based on Video transmission, video calling, video streaming etc. Video transmission over wireless network has many challenges like to overcome packet loss, delay and to increase throughput. In case of bad quality video, it will degrade the Quality of Experience (QoE) felt by the end user. So, evaluating the video transmission is necessary. In this paper, we have evaluated a video transmission using EvalVid tool in NS2, which is an open source video quality evaluation tool. We have compared the quality of performance evaluation metrics' like PSNR, SSIM and VQM.

Key Words: QoE, EvalVid, PSNR, SSIM, VQM.

I. INTRODUCTION

Video transmission over wireless environment has become a necessity for almost every internet user. Typically, for video transmission requires good internet connectivity with high bandwidth. Generally, multimedia applications need high quality of service. Previously, we used to store videos on hard disk which is wastage of memory and energy. Now we can directly watch online. There are two typing streaming, live streaming and offline streaming. In offline streaming, if video freezes at some point user cannot follow the content anymore. But, in live streaming, retransmission is no use, unless retransmission is finished before that part of content is scheduled to play. In our paper, the main problem is video transmission over mobile ad hoc networks and Quality of Experience (QoE) of end user due to bad quality of video. The poor quality of video is due to network congestion which may cause delay, packet loss, and jitter. So, in this paper we evaluate how much quality of video degrades from original video to distorted video.

For simulation, NS2 with EvalVid tool is used. NS-2 is the one of the most widely used network simulator. It is useful for simulating various networks like wireless LANs, ad hoc networks and Ethernet. First we create EvalVid environment in NS2 then analyze the results and performance by evaluating video quality by EvalVid tool. This tool can compare and evaluate videos only with equal number of frames.

EvalVid tool evaluates the video by considering Peak Signal to Noise Ratio (PSNR), Structural Similarity Index (SSIM), Video Quality Metrics (VQM). And this tool compares the

video frame by frame which helps in locating the position where distortion occurred.

II. INSTALLATION PROCEDURE

A. Prerequisites for using EvalVid

- A Linux operating system with Ubuntu 15.04.
- Download gpac files for ubuntu. GPAC files provide isomedia files for Linux (ubuntu).
- Download Network Simulator 2 (NS2) i.e. ns -2.35 from [1]
- Download the EvalVid 2.7 and its binaries from [2]
- To complete installation move gpac files to EvalVid 2.7 folder.

B. Basic Requirements of EvalVid tool are:

- The videos taken must be same size in term of frames.
- Both original video and distorted video should be same format i.e. raw format because EvalVid takes only raw format videos as input.
- Both video files taken should be of the same codec.
- The basic structure is shown in Fig.1.

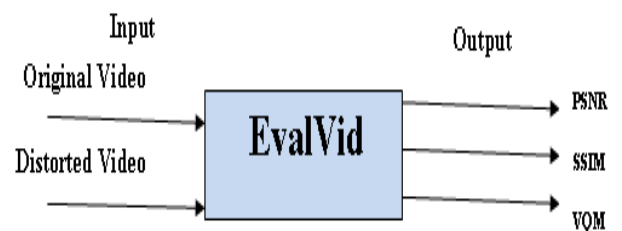


Fig.1. Illustrative figure for EvalVid

C. Integration of NS2 and EvalVid

The basic and integrated architecture is shown in Fig.2. To integrate NS2 and EvalVid, first modify code of packet.h, agent.h, agent.cc, tcl/lib/ns-default.tcl present in ns-allinone-2.35/ns-2.35 as shown in [3].

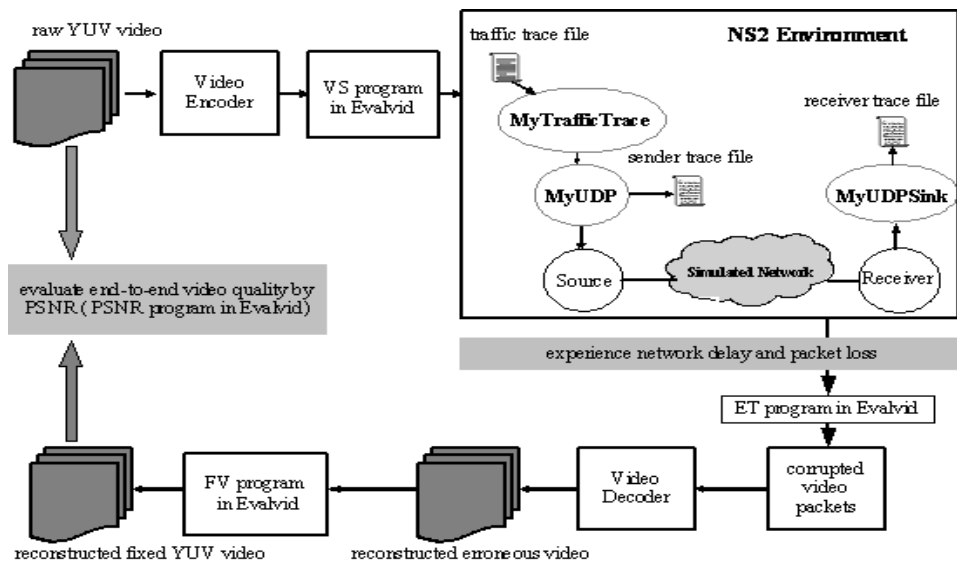


Fig.2 Integrated architecture of NS2 and EvalVid (Courtesy: csie.nqu.edu.tw)

Then create myEvalVid folder in ns-allinone-2.35/ns-2.35 and put myEvalVid.cc, myEvalVid_sink.cc, myEvalVid_sink.h, myudp.cc, myudp.h in it, shown in Fig.3.

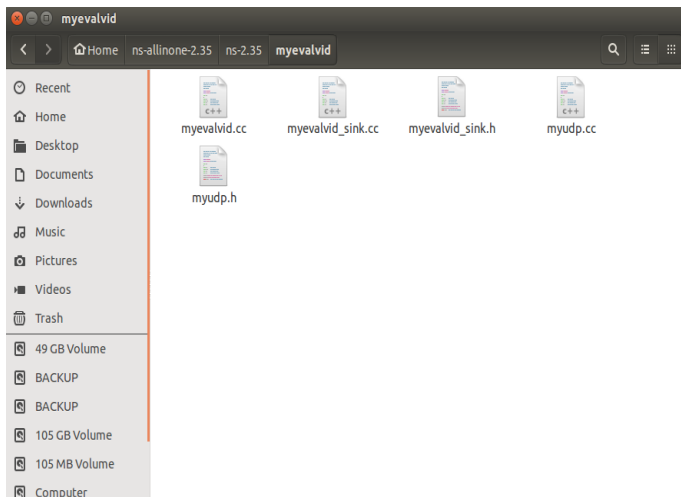


Fig.3. MyEvalVid folder

Modify the makefile present in /ns-allinone-2.35/ns-2.35 by putting myEvalVid/myudp.o, myEvalVid/myudpsink2.o, myEvalVid/mytraffictrace2.o in the OBJ_CC list.

Recompile NS2

```
$cd ns-allinone-2.35/ns-2.35
$ ./configure --with-tcl-ver=8.5.10; make clean; make
```

D. Installing tools for multimedia

Most of tools like psnr, MP4, fixyuv etc. work without installing any extra software's like codec. But FFmpeg, etmp4, MP4Box tools require special codec software and library files.

Download GIT-version of FFmpeg from [4], FFmpeg is a command line tool where FF means “fast forward” and MPEG is a video standard. It is mainly used to convert one multimedia format to another format. Installation steps are shown below:

```
$ cd ffmpeg-3.2.4
$ configure
$ ./configure
$ ./configure --disable-yasm
$ make
$ make install
$ ./install
$ sudo make install
$ cd ns-allinone-2.35/ns-2.35
$ sudo ./validate
```

Install gpac from command prompt by using command
\$ sudo apt-get install gpac

Then download YUV CIF reference video for akiyo_cif.264 from [5], H.264 is a lossless encoded video. The YUV model define color space in terms of brightness (luma) and color (chrominance). Previous black-and-white systems used only luma (Y) information and now color information (U and V) was added, so that black and white receiver would still be able to display a color picture as a normal black and white picture. Most commonly used YUV sampling ratios are 4:4:4 indicates no down sampling.

Step by step procedure

1. Then convert akiyo_cif.264 to akiyo_cif.yuv:
\$ffmpeg -i akiyo_cif.264 akiyo_cif.yuv
2. Encode the raw yuv file to m4v:

```
$ffmpeg -s cif -r 30 -i akiyo_cif.yuv -vcodec mpeg4 -b 64000 -bt 3200 -g 30 akiyo_cif.m4v
```

3. Convert m4v file to mp4 by using MP4Box:
 \$MP4Box -hint -mtu 1024 -fps 30 -add akiyo_cif.m4v akiyo_cif2.mp4

MP4Box is a multimedia package available in GPAC. It is mostly used for manipulating ISO media files. (e.g. MP4, 3GP) and encoding and decoding multimedia files (e.g. MPEG, AVI).

4. The MP4trace is able to send a hinted mp4-file per RTP/UDP/IP to a specified destination. The output of mp4trace will be needed later, so it is redirected to a file.

```
$/mp4trace -f -s 224.1.2.3 12346 akiyo_cif2.mp4 >Source_Video_Trace
```

Then it creates a Source_Video_Trace with the traces shown in figure. And shows output on screen as

```
Track 1: Video (MPEG-4) - 352x288 pixel, 300 samples, 00:00:10.000
Track 2: Hint (RTP) for track 1 - 300 samples, 00:00:10.000
```

5. Run NS2 script for wireless simulation scenario available in [6]. After simulation we will get sd_be for sender trace file and rd_be for the receiver trace file.

```
$ ns be.tcl //command for running NS2 script
```

6. Then etmp4 is used to generate corresponding receiver video file by using sender trace file, receiver trace file, source_video_trace shown in Fig.4 (this is generated by mp4 trace).

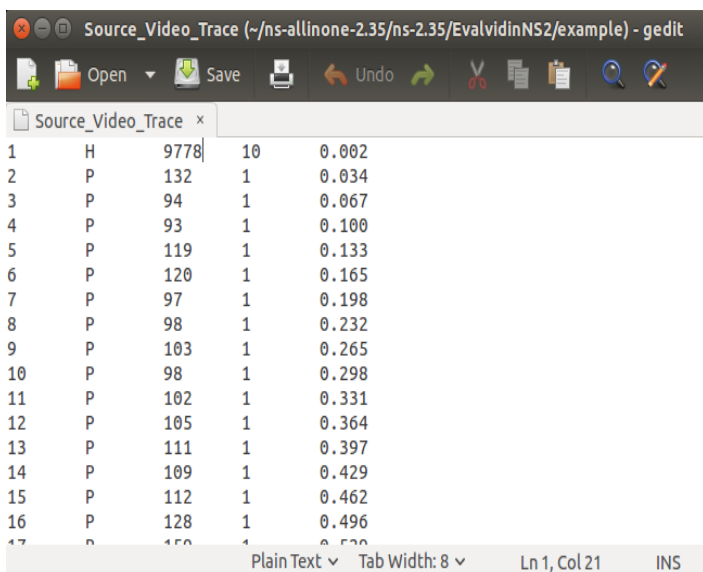


Fig. 4. Content of Source_Video_Trace

```
$ ./etmp4.exe sd_be rd_be source_video_trace akiyo_cif.mp4 akiyo_cife
```

Then distorted video is generated with name akiyo_cife.m4v and Etmp4 also creates some more files loss_akiyo_cife.txt, delay_akiyo_cife.txt, rate_s_akiyo_cife.txt, rate_r_akiyo_cife.txt as shown in Fig.5.

7. Next akiyo_cife.m4v is converted to akiyo_cife.yuv, which is a corresponding received video file in raw yuv format.
 \$./ffmpeg.exe -i akiyo_cife.m4v akiyo_cife.yuv

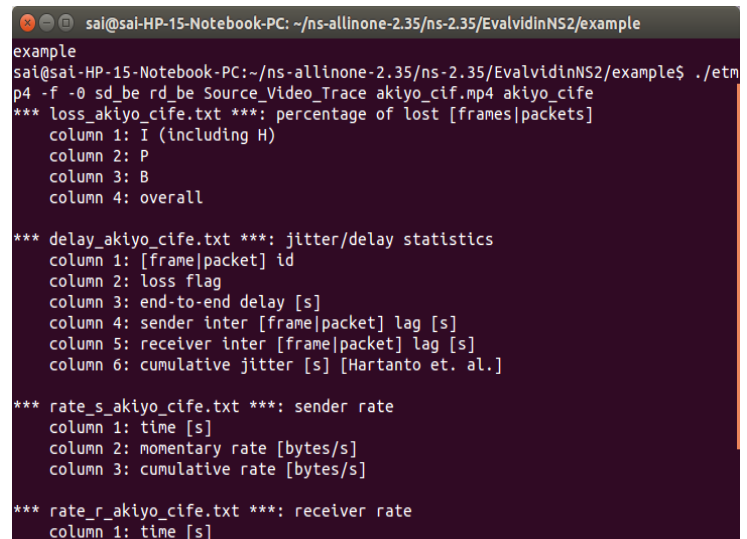


Fig. 5. Output files created by etmp4

III. PERFORMANCE ANALYSIS

A. PSNR: Peak Signal to Noise Ratio

The received and original video streams are compared frame by frame format. For distorted part of video we get PSNR as some non zero value and for similar frames values are zero.

Mean =

$$\frac{\sum \text{PSNR values of distorted video/frame obtained from tool}}{\text{Number of frames}}$$

The command for calculating PSNR:

```
psnr x y <YUV format> <source.yuv> <destination.yuv>
```

Here x is the frame width

y is the height width

YUV format: 444, 420, 422 etc.

Source.yuv is the original video

Destination.yuv is the received video (distorted video)

The PSNR of the original and received video is calculated by:

```
$ ./psnr 352 288 420 akiyo_cif.yuv akiyo_cife.yuv
```

The output for this is peak signal to noise ratio for original and received video along with mean and standard deviation shown in Fig. 6.
 psnr: 100 frames (CPU: 0 s) mean: 25.77 stdv: 2.17 (stdv = standard deviation)

```

sal@sal-HP-15-Notebook-PC: ~/ns-allinone-2.35/ns-2.35/EvalvidinNS2/example
26.237
25.184
24.842
24.899
24.791
24.822
24.909
24.838
24.661
24.685
24.858
24.825
26.128
25.812
25.761
25.718
25.637
25.585
24.495
22.483
22.509
22.471
psnr: 100 frames (CPU: 0 s) mean: 25.77 stdv: 2.17
sal@sal-HP-15-Notebook-PC:~/ns-allinone-2.35/ns-2.35/EvalvidinNS2/example$
    
```

Fig. 6. Output of psnr

Excellent values range from 30dB to 50 dB, where acceptable range in wireless transmission is around 25 dB.

B. SSIM: Structural Similarity Index

SSIM measures the correlation between sender and receiver end using distortion of video streams and includes contrast, luminance. Its value varies from 0 to 1. For a good quality video/ frame the values are 1.

The command for calculating SSIM:

```

psnr x y <YUV format> <source.yuv>
<destination.yuv>[ssim]
    
```

Here x is the frame width

y is the height width

YUV format: 444, 420, 422 etc.

Source.yuv is the original video

Destination.yuv is the received video (distorted video)

ssim is to be mentioned for calculating SSIM

C. VQM: Video Quality Metrics

It is video quality experienced by humans. It is concerned with how a video is observed by a viewer. It is also used to measure unwanted things present in video stream between source and destination. The unwanted factors include blurred effect, motion effect, different noise and distortion effect. If the result of those effects is close to 0 means the processing video quality is good.

IV. SIMULATION RESULTS

The Network Simulator (NS2) and EvalVid tool is used to simulate EMRP-QQ [7] and FQ-MP-OLSR [8] for source file akiyo_cif.yuv with 300 frames in 352 X 288 quality video. PSNR, SSIM, VQM is shown in Fig.7, Fig.8, Fig.9. We set up the simulation in an area of 1000 X 1000 sq.m. for a random way point mobility model with varying number of nodes and IEEE 802.11 MAC protocol is used for our implementation. The two ray ground transmission model is used with the transmission range 250m and 11 Mbps bandwidth.

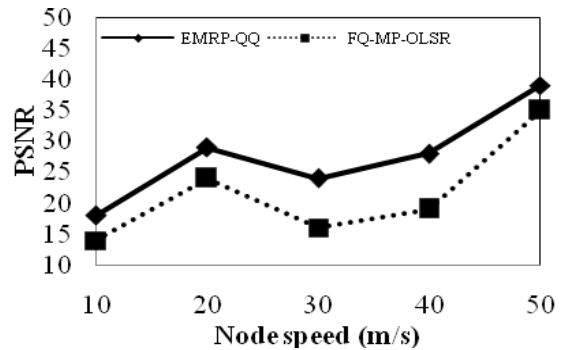


Fig.7. PSNR for 352X288 quality video

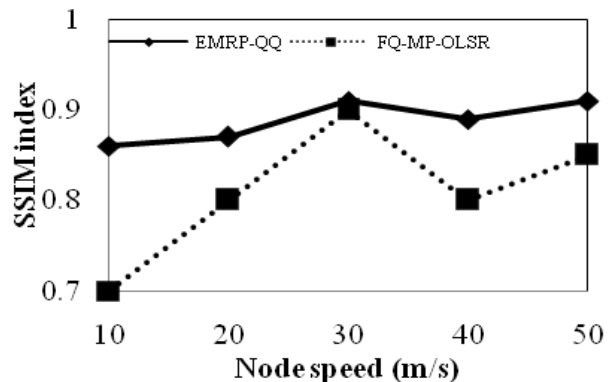


Fig.8. SSIM for 352X288 quality video

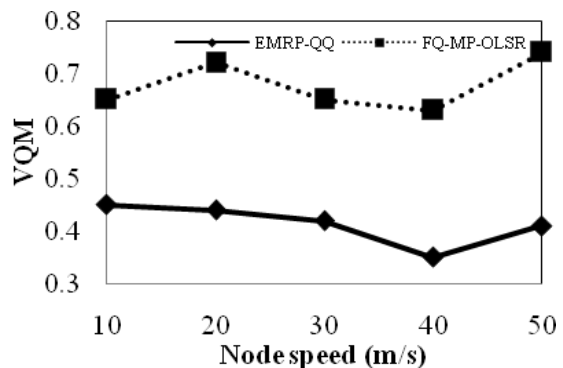


Fig.9. VQM for 352X288 quality video

V. CONCLUSION

In this paper, we presented a step by step procedure for creating source and destination trace files which are useful to compare frame by frame of original and distorted videos. Calculated and compared PSNR, SSIM and VQM for values for two existing algorithms.

VI. REFERENCES

- [1] <http://www.isi.edu/nsnam/ns/ns-build.html>
- [2] <http://www2.tkn.tu-berlin.de/research/EvalVid/fw.html>
- [3] http://csie.nqu.edu.tw/smallko/ns2_old/EvalVid_in_NS2.htm
- [4] <https://ffmpeg.org/download.html>
- [5] <http://www2.tkn.tu-berlin.de/research/EvalVid/cif.html>
- [6] <http://csie.nqu.edu.tw/smallko/ns2/myEvalVid.htm>
- [7] V. Sowmya Devi and Dr. Nagaratna P Hegde "Energy Efficient Multipath Routing Protocol for Enhancing QoS and QoE in Multimedia Applications for MANETS", International Journal of Communication Networks and Information Security (IJCNIS), Volume 8, No. 3, December 2016, Page no. 158 – 170.
- [8] A. Boushaba, A. Benabbou, R. Benabbou, A. Zahi and M. Oumsis, "An intelligent multipath optimized link state routing protocol for QoS and QoE enhancement of video transmission in MANETS", Computing, vol. 98, no. 8, pp. 803-825, 2015.