# MITIGATING THE EFFECT OF PACKET LOSSES ON REAL-TIME VIDEO STREAMING USING PSNR AS VIDEO QUALITY ASSESSMENT METRIC

**Anietie Bassey, Kufre M. Udofia & Mfonobong C. Uko**
Department of Electrical/Electronic
& Computer Engineering, University of Uyo, Uyo
**NIGERIA**

## ABSTRACT

Real-time video streaming refers to content delivered live over the Internet. It requires a form of source media (e.g. a video camera, an audio interface, screen capture software), an encoder to digitize the content, a media publisher, and a content delivery network to distribute and deliver the content. Real-time video streaming (live streaming) is complicated due to a number of factors including bandwidth, jitter, and packet losses as well as many additional issues such as how to fairly share the network resources amongst many flows and how to efficiently perform multicast (one-to-many) communication for popular content. In our previous paper, we proposed an Adaptive Media Play-out (AMP) algorithm and a mathematical model to reliably transmit packets, while reducing jitter and improving video quality at the receiving end. The proposed AMP algorithm and model enabled a valuable trade-off between quality of service (QoS) parameters and received video quality. Through simulation experiments, the performance of the AMP was compared with the existing technique of buffering - on the basis of that research, it was found that the AMP algorithm outperforms buffering to a large extent. In this paper, we use the peak signal-to-noise ratio (PSNR) as video quality assessment metric to mitigate packet losses on real-time video streaming. We have shown that by using the PSNR as the video quality assessment parameter, the effect of packet loss on real-time video streaming over the internet can be mitigated. The video frame rates for the PSNR analysis were compressed at 15, 20, 25, 27, 28, 29 and 30fps respectively. The result of our analysis shows that the higher the average frame rate received, the higher the PSNR, the lower the loss rate, and the better the video quality.

**Keywords:** Packet loss, Real-time video streaming, PSNR.

## INTRODUCTION

Packet losses are usually caused by an insufficient bandwidth, noise on the transmission channel, network congestion, and out-of-time delivery of packets due to the latency. In this research, PSNR metric is used to make video bit streams more resilient to packet loss and to produce better quality video to the viewer. We have also validated our PSNR analysis through simulation experiment and shown how the PSNR metric provides performance benefits in terms of packets latency and how it affects frame rate.

Real-time video streaming is an important component of many internet multimedia applications, such as distance learning, digital libraries, church services, conferences, etc. Real-time video streaming applications can tolerate some packet loss but require timely delivery of information. Although TCP is standardized for RTP, it is not normally used in RTP application because TCP favours reliability over timeliness (Postel, 2012) [1]. Therefore, the majority of RTP implementations are built on top of UDP, which is particularly suitable for transmission of real-time traffic. RTP consists of a data part which supports applications with real-time properties. Specifically provides timestamps for synchronization, sequence numbers for packet loss and reordering detection, security and content identification. Generally, RTP enables to identify the type of data being transmitted, determine what order the packets of data should be presented, and synchronize media streams from different sources (Frederick and Jacobson, 2003) [2].

However, RTP itself does not provide any mechanism to ensure timely delivery or provide other QoS guarantees. RTP is usually implemented within the application, because control mechanisms such as congestion control and loss recovery have to be implemented in the application level. The control part of RTP is called RTCP that can support real-time applications between unicast users of multicast groups with any size over the Internet. It offers QoS feedback from the receivers to the multicast group as well as support for the synchronization of different media streams. RTSP is more of a framework than a protocol that allows control of multimedia streams over the Internet. RTSP has been designed to run on the top of RTP to control and deliver real-time video packets.

The delivery mechanisms are usually based on RTP. RTSP controls over RTP which includes absolute positioning within the media stream, recording and device control. It also supports interoperation between clients and servers from different vendors. RTSP session is not tied with a transport-level protocol such as TCP connection; however, it could be used with a connectionless transport protocol such as UDP. RTSP does not typically deliver the continuous streams itself; RTSP merely acts as a network remote control for multimedia streaming servers (Wu et al., 2001) [3].

## RELATED WORK

Researchers had gathered empirical data on the effects of transmitting MPEG video over the internet (Boyce and Gaglianello, 1998) [4]. Pinson and Wolf (2004) had proposed protocols that use selective retransmission for recovering from bit errors [5]. Nick Feamster and Hari Balakrishnan, in their work in 2002 – “Packet Loss Recovery for Streaming Video” proposed a mechanism for recovering data packets, after selective retransmission. They quantified the effects of packet loss on the quality of MPEG-4 video, developed an analytical model to explain the effects, presented a system to adaptively deliver MPEG-4 video in the face of packet loss and variable internet conditions and evaluated the effectiveness of the system under various network conditions [6].

Prior studies have also gathered experimental results that describe packet loss characteristics for MPEG video and suggest the need for better error recovery and concealment techniques (Boyce and Galianello, 1998). In our previous research, we proposed an Adaptive Media Play-out (AMP) algorithm and a mathematical model to reliably transmit packets, while reducing jitter and packet losses when streaming video in real time (Bassey, Ozoumba and Udofia, 2015) [7].

## VIDEO QUALITY ASSESSMENT USING PSNR METRIC

The ability to successfully decode a compressed bit stream with inter-frame dependencies depends heavily on the receipt of reference frames (i.e., I-frames and to a lesser degree P-frames). While the loss of one or more packets in a frame can degrade video quality, the more problematic situation is the propagation of errors to dependent frames. If we assume that the viewer can only tolerate frames that are at least a certain PSNR quality, and that frames below such a quality are not pleasing to view, then we can show how packet losses degrade video quality (Bassey, 2015) [8]. The effects of packet losses on different frame rates at the receiver are analysed here. Using these results, we submit that under certain PSNR conditions (20dB), packet loss can degrade the quality of the received bit-streams. Therefore, the analysis is based on two premises.

First, that packet loss will result in degradation of quality of the video stream at the receiver, i.e., that there is some signal loss caused by the loss of a packet during transmission. This is true in general, except on the condition that packet-level FEC or erasure codes are extensively used, but since FEC and other retransmission mechanisms do reduce the effective bit rate of transmission, it is complicated and often introduces additional end-to-end delay in the system. The second premise is that, below a certain PSNR level, frames are not viewable by users. PSNR values smaller than 20 dB are generally not viewable, meaning that the individual frames are not particularly useful without a correction mechanism at packet loss rates larger than $2^{-8}$ (Feamster and Balakrishnan, 2002) [6]. Since the threshold of the viewing experience varies from one frame rate to another, the analysis were performed for several frame rates of the same video using the PSNR as the video quality metric.

PSNR is a prominent performance indicator of video quality that is derived from the root mean squared error (RMSE). The PSNR for a degraded $N_1$ x $N_2$, 8 bit image f' from the original image f is computed by the formula:

$$PSNR = 20\log_{10}\frac{255}{\left(\frac{1}{N_1N_2}\sum_{x=0}^{N_1-1}\sum_{y=0}^{N_2-1}[f(x,y)-f\prime(x,y)]^2\right)^{1/2}} \qquad \text{---} \qquad \text{---} \qquad (1)$$

The aim of this analysis is to derive a relationship between the packet loss rate $P$ and the "observed frame rate" $f$. When calculating the frame rate, $f$, we assume that if the quality of a frame (PSNR) falls beneath a certain threshold, then the frame is "dropped". We then express the observed frame rate $f$ as $f_0\,(1-\theta)$, where θ is the "frame drop rate", the function of frames dropped, and $f_0$ is the frame rate of the original bit-stream in frames per second (fps).The frame drop rate θ is a sum of conditional probabilities:

$$\theta = \sum_i P(f_i)\cdot P(\bar{F}/f_i) \qquad \text{---} \quad \text{---} \quad \text{---} \quad \text{---} \quad \text{---} \quad \text{---} \qquad (2)$$

Where $i$ runs over the three possible frame types (I, P, and B) respectively, and $\bar{F}$ represents the event that a frame is "useless" because it falls below a certain quality threshold. $f_i$ is the event that the corresponding frame is of type $i$. $P(f_i)$ can be determined directly from the fractions of bit-stream data of each frame type. We then deduce that:

$$P(\bar{F}/I) = 1-(1-P)^{S_I} \qquad \text{---} \quad \text{---} \quad \text{---} \quad \text{---} \quad \text{---} \quad \text{---} \qquad (3)$$

Where $S_I$ is the number of packets on average in an I-frame, and P is the packet loss rate. The conditional probabilities for P and B frames are somewhat more involved, and require an understanding of the inter-frame dependencies in MPEG video. These dependencies are shown in figure 1.

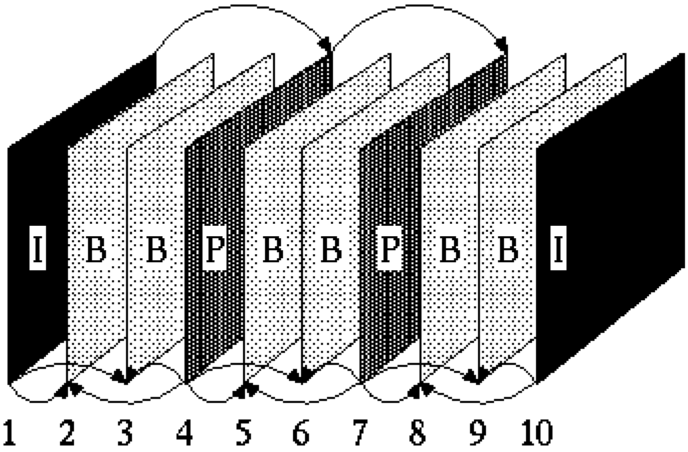

Figure 1: Frame dependencies in an MPEG bit stream

Every P-frame depends on the preceding I or P frames in the "group of video object planes", and every B-frame depends on the surrounding two reference frames (the closest two I or P frames that surround it). Thus, the successful decoding of a P-frame depends on all I and P frames that precede it in the GOVP, and the successful decoding of a B-frame depends on the successful decoding of the surrounding reference frames, which implies the successful decoding of all preceding I and P frames and the succeeding I or P frame. These dependencies can be expressed in the following relationships:

$$P(\bar{F}/P) = \frac{1}{N_p}\sum_{k=0}^{N_p}[1-(1-P)^{S_I+KS_P}] \quad --- \quad --- \quad --- \quad --- \quad (4)$$

$$P(\bar{F}/B) = \frac{1}{N_p}\sum_{k=0}^{N_p}[1-(1-P)^{S_I+(K+1)S_P+S_B}] \quad --- \quad --- \quad --- \quad (5)$$

Here, $S_p$ is the average number of packets in a P-frame, $N_P$ is the number of P-frames in a GOVP, and $S_B$ the number of packets in a B-frame. These equations can be simplified to the following closed form expressions:

$$P(\bar{F}/P) = 1 - \frac{(1-P)^{S_I}}{N_p[1-(1-P)^{S_P}]}[1-(1-P)^{S_P N_P}] \quad --- \quad --- \quad --- \quad (6)$$

$$P(\bar{F}/B) \leq 1 - \frac{(1-P)^{S_I+S_P+S_B}}{N_p[1-(1-P)^{S_P}]}[1-(1-P)^{S_P N_P}] \quad --- \quad --- \quad --- \quad (7)$$

An expression for θ using equations 2 and 3, equations 6, and 7 can be obtained. Given this expression for θ, we can determine $f = f_0\ (1 - \theta)$, given values of $N_p$, $S_I$, $S_P$, $S_B$, and $f_0$. This result can be extended to derive analytical results for lower PSNR thresholds, assuming that there is a relationship between the rates of packet losses in a particular frame. Again, instead of performing the calculations so that one packet loss results in a "useless" packet, we can generalize to allow for *n* losses, with a larger value of *n* corresponding to a lower threshold PSNR.

## DESCRIPTION OF THE VIDEO STREAMING PROCEDURE

In order to stream the videos, a streaming server to stream and a client to receive the stream is needed. Both the server and client should have a Video LAN Client (VLC) player installed. The client should be connected to the streaming server via a wired/wireless LAN connection.

After the server and client systems are connected, the configuration for server and client is presented separately. A more detailed explanation with screenshots of how the video used for this study was streamed is presented below.

**Configuration at the Streaming Server:**

(i) Start the VLC player and then on the main Menu press “Media” option
(ii) Then click on “Streaming”
(iii) In the next menu click on “Add” to add the video files and then click on “Stream” button can be found at the end of menu, this can found in figure 3.

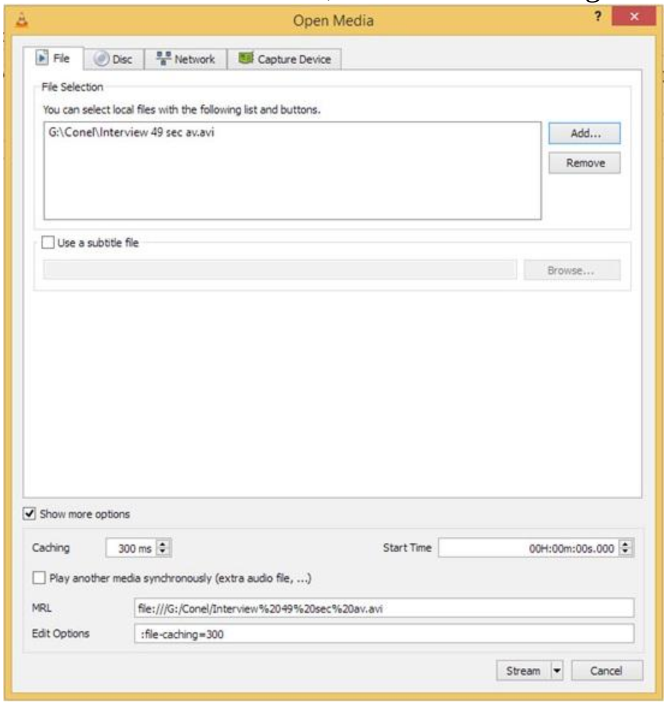


Figure 3: Adding a file to VLC to stream

(iv) In the next menu as in figure 4, under New Destination select UDP and click on Add button.

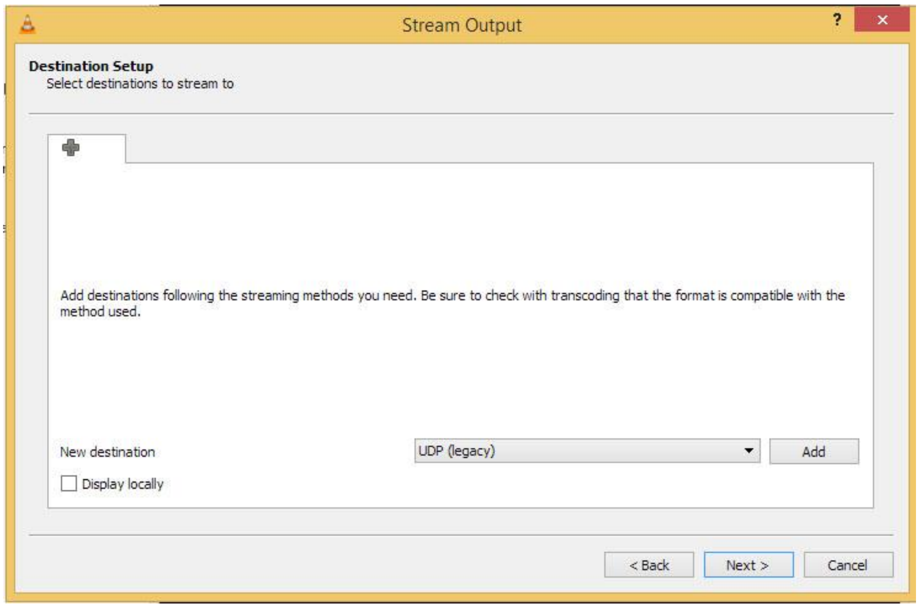


Figure 4: Adding new destination to stream the video

(v) In the next menu from figure 5, fill in the "Address" field with the IP address of the destination or client and leave the port number as "1234".

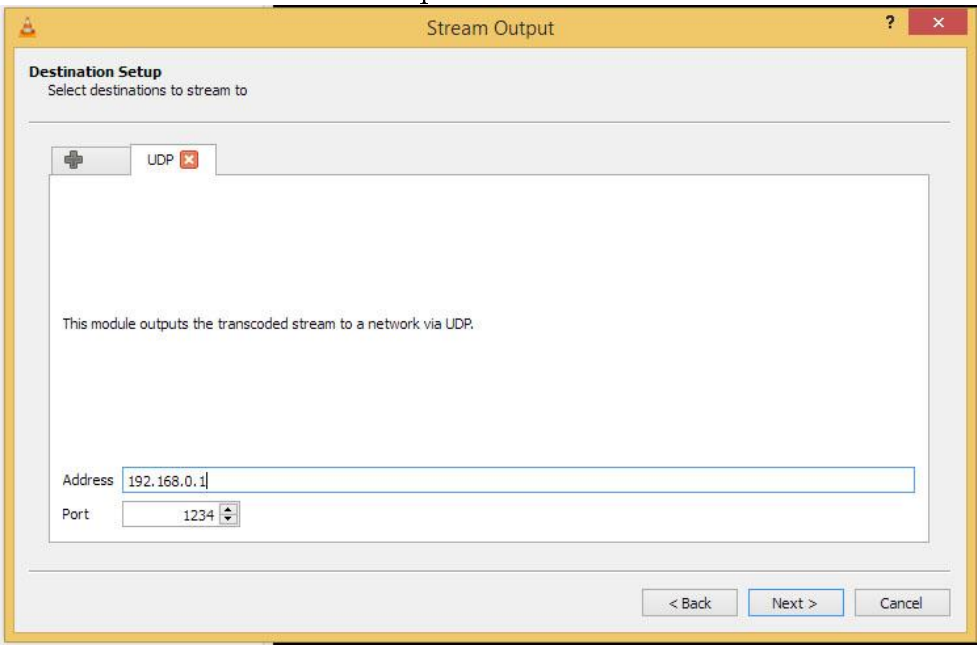


Figure 5: Adding destination address

(vi) Then click on "Edit selected profile" in the profile menu and then in video codec keep the bit rate at 800 kb/s which is a default value and then the frame rate to 25fps.

(vii) Click on save, then click on "Next" and then in the next screen just click on "Stream"

**Configuration at the Client (User) Side**

(i) At the client side open VLC player and click on "Media" and then click on "Open Network Stream"

(ii) In the next screen select the protocol as "UDP" and leave the Address as blank and port number as 1234 as shown in figure 6.

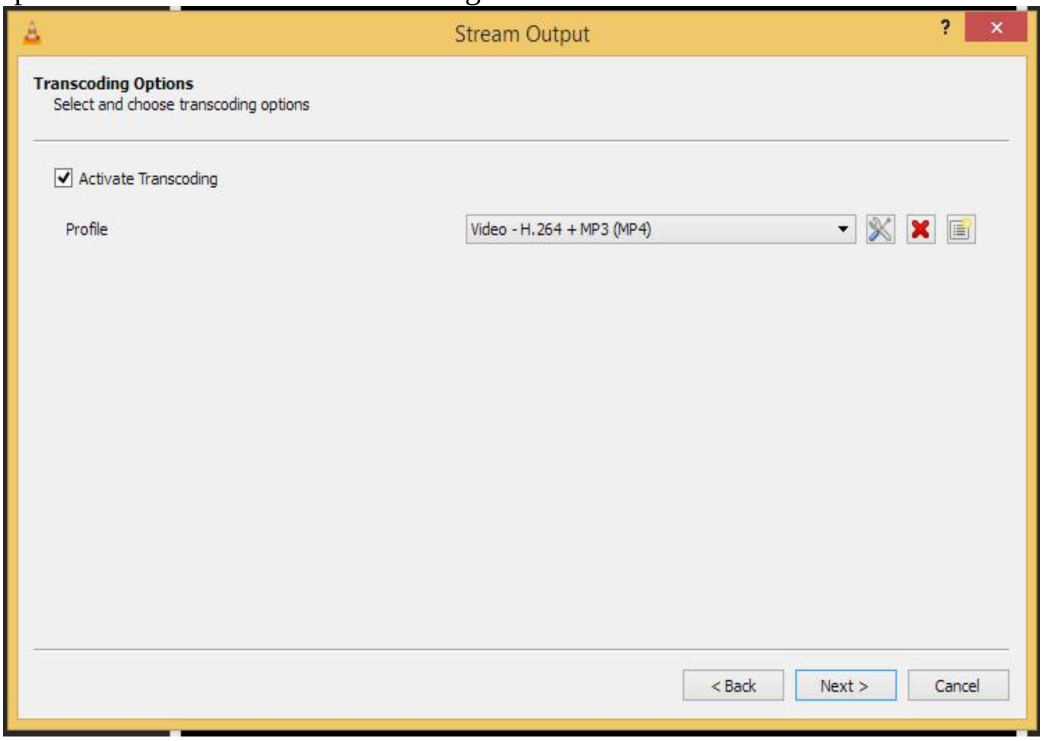


Figure 6: Receiving video stream

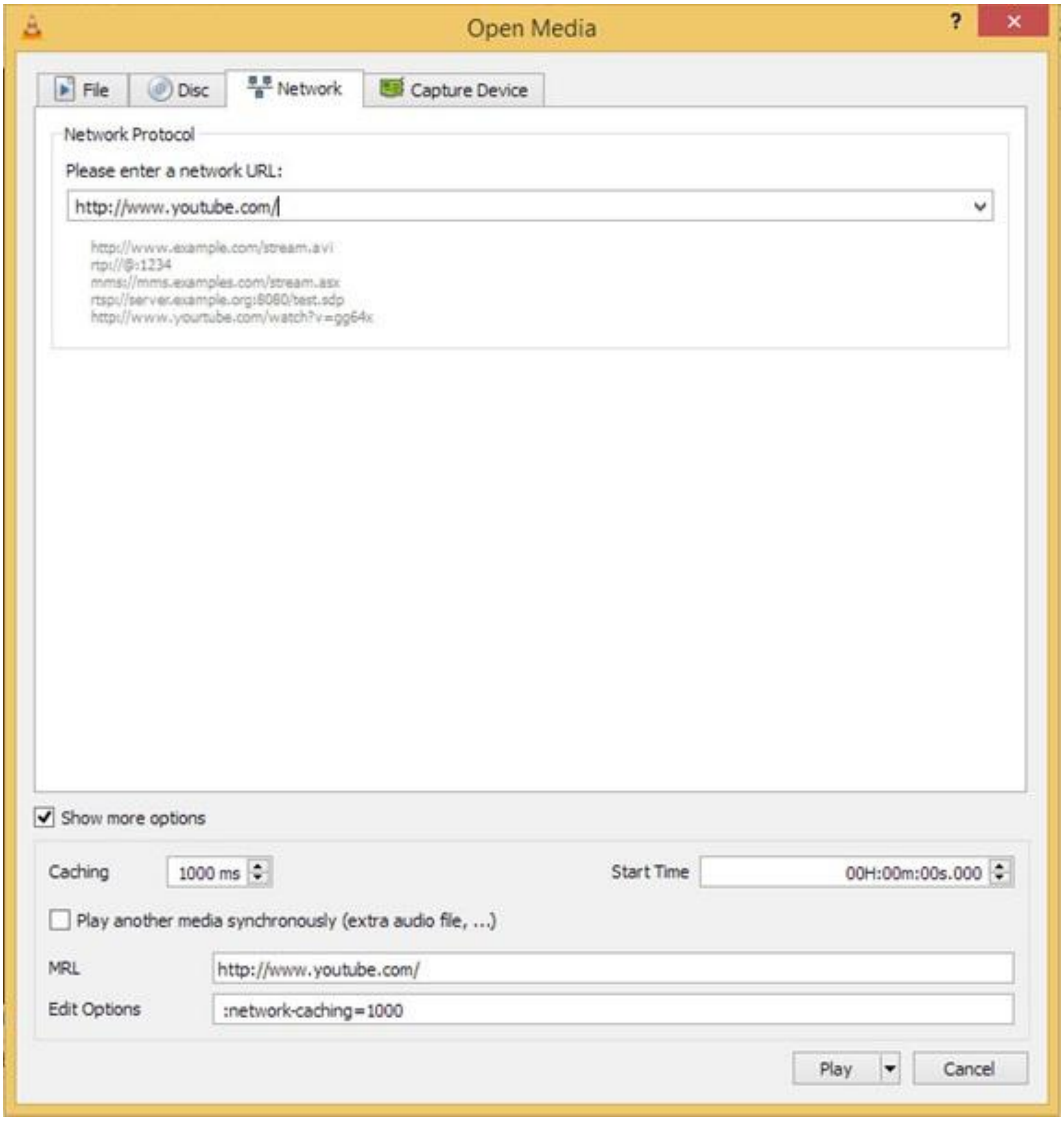

Figure 7: Playing video stream

(iii) Then click on "Play" and the video will be streamed to the client and will be played.

**Recording and Saving of Streaming Video**

VLC media player can convert and save different videos that are being streamed over a network. This option is much useful for the users to save the streaming videos. In this section the configuration for saving on streamed videos is presented.

(i) Click on "View" in the VLC"s main menu and then select "Advanced Controls"
(ii) When playing a network stream video at the client end, click on "Record" button in the advanced controls and the video is saved to default location.
(iii) If the video needs to be saved then in "Media" click convert/save then select the Network option that comes on the next window, select the destination where the file needs to be saved.
(iv) Check mark on Dump Raw I/P and click play.

## DESCRIPTION OF THE SIMULATION SETUP

The characteristics of the video are shown in the video source from the video camera. The video (MPEG 2) is compressed from a Chroma sub-sampling ratio of 4:4:4 to 4:2:0 by the MPEG encoder and then sent to the RTP-UDP/IP modules. After transmission from the network channel (Internet) the received video streams are then decoded (compressed) from a ratio of 4:2:0 to the original sampling ratio of 4:4:4 before it is sent to the buffer at the receiver, and then to the embedded Simulink function to determine the PSNR. The signal is then displayed on the video display – computer devices (laptops, tablets, smart phones, etc.) as the received video.

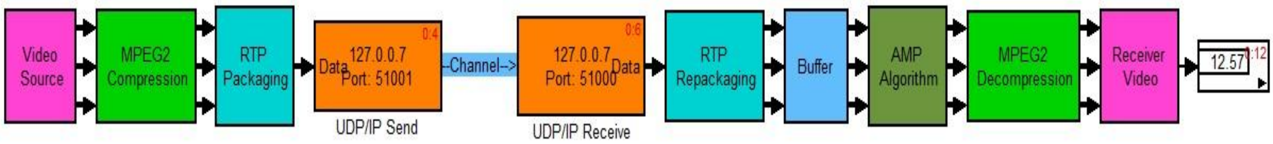


Figure 8: Block diagram of simulation and streaming procedure

The audio stream used in this work was compressed using an audio codec of Media Player 3 (MP3) standard, the video stream is compressed using an improved video codec of H.264 standard, and the encoded audio and video streams were assembled in a container bit stream of Audio Video Interleave (AVI) format.

According to ITU-R (ITU-R Recommendation, 2002) [7], the length of the video should be at least 5sec. Therefore, the selected video is of 15 seconds duration. The shorter duration videos are used so that the viewer wouldn't get bored to watch each video and rate them. It is interesting to notice that the video is used because of its characteristics – high movements. The video frame rates for the PSNR analysis were compressed at 15, 20, 25, 27, 28, 29 and 30fps. The simulation graphs of PSNR vs Time are as shown in figures 9 to 15. The video sequences of YUV file format were converted to.avi format using an open source software tool called Adobe Premiere Pro CC (Windows) encoder.

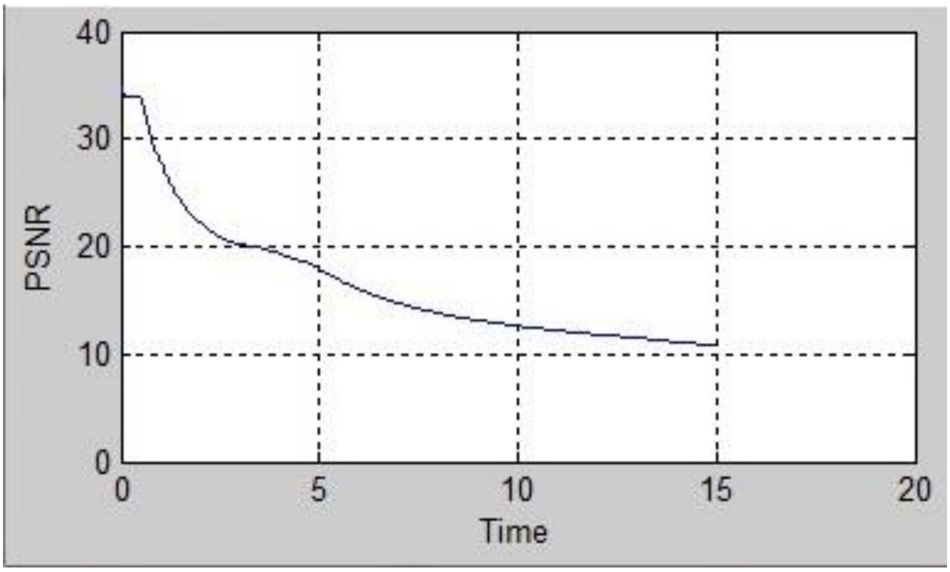


Figure 9: Peak Signal-to-Noise Ratio (PSNR) value (10.83) at 15fps

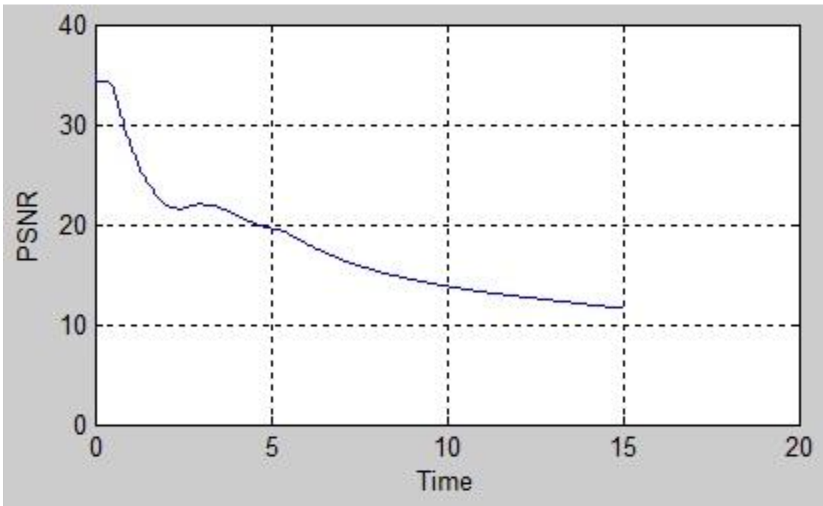


Figure 10: Peak Signal-to-Noise Ratio (PSNR) value (11.62) at 20fps

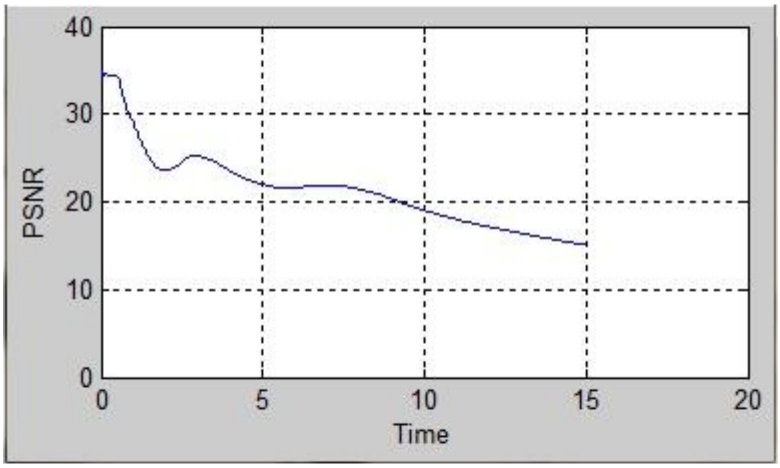


Figure 11: Peak Signal-to-Noise Ratio (PSNR) value (15.1) at 25fps

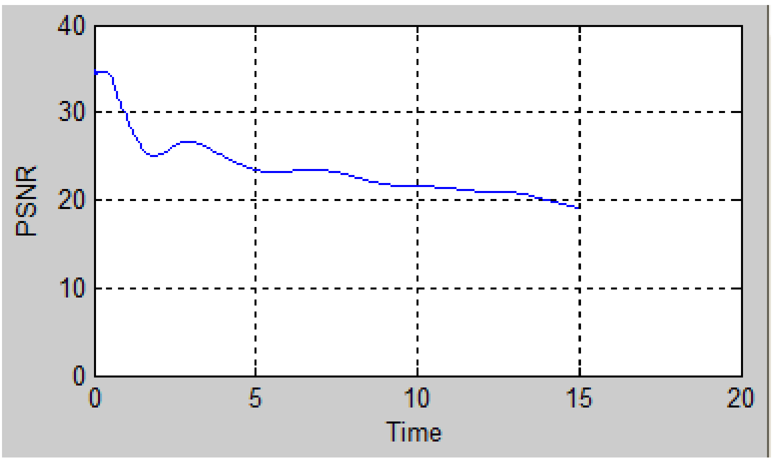


Figure 12: Peak Signal-to-Noise Ratio (PSNR) value (19.11) at 27fps

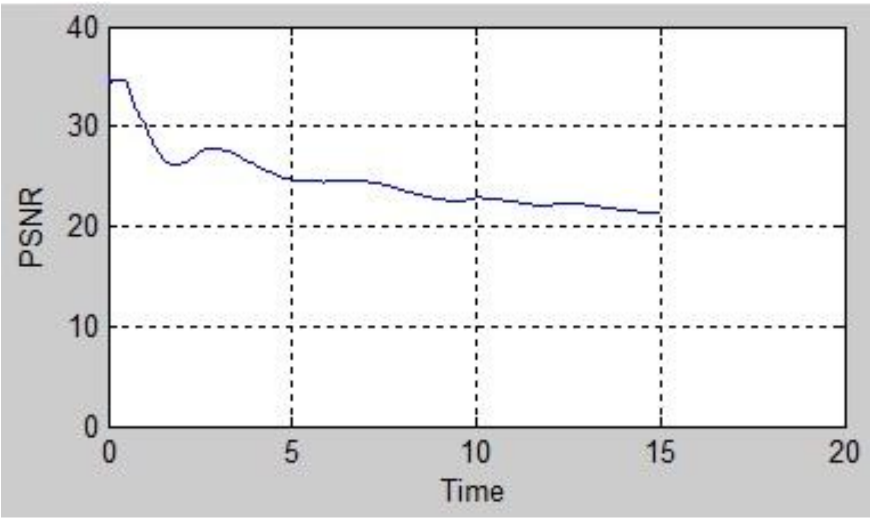


Figure 13: Peak Signal-to-Noise Ratio (PSNR) value (21.31) at 28fps

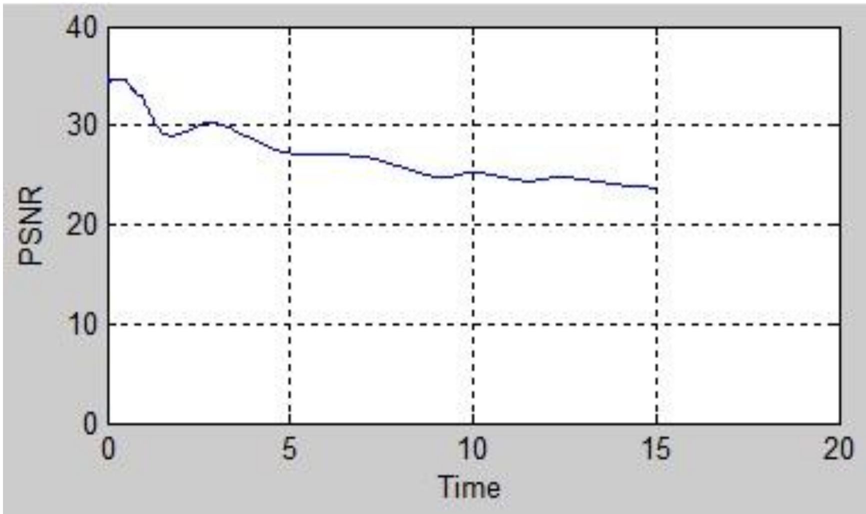


Figure 14: Peak Signal-to-Noise Ratio (PSNR) value (23.76) at 29fps

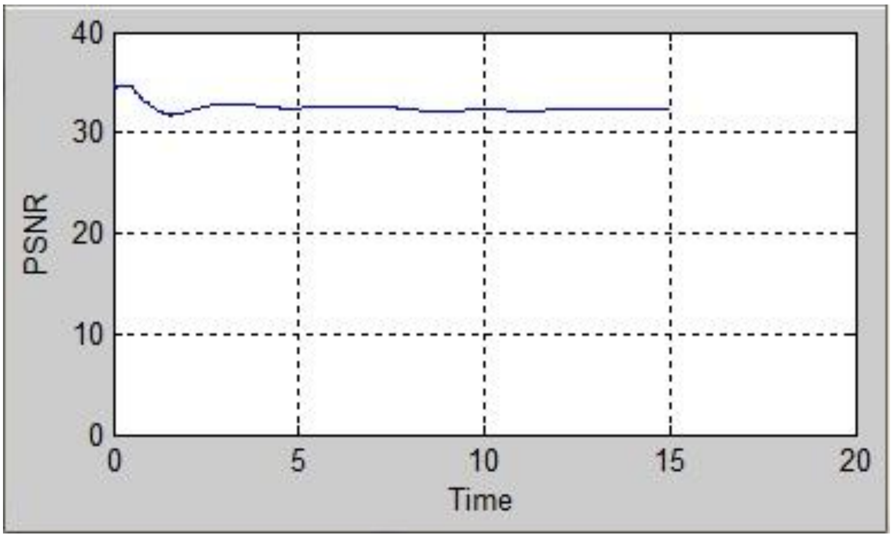


Figure 15: Peak Signal-to-Noise Ratio (PSNR) value (32.31) at 30fps

## SIMULATION RESULTS

The peak signal to noise ratio, PSNR, is still the most widely used visual quality metric. It is recommended that PSNR values of smaller than 20 dB are generally not viewable and unacceptable in communications (IETF, 2008) [8]. This fact has been confirmed in this research with various simulation experiments at varying video frame rates as seen in table 1.

Table 1: Relationship between frame rate and peak signal-to-noise ratio (PSNR)

| Frame Rate (fps) | PSNR (dB) |
|---|---|
| 15 | 10.83 |
| 20 | 11.62 |
| 25 | 15.1 |
| 27 | 19.11 |
| 28 | 21.31 |
| 29 | 23.76 |
| 30 | 32.31 |

Figure 16 shows the simulation results of the frame rates as a function of packet loss rate, with one dot per PSNR value. The result of our analysis shows that the higher the average frame rate received, the higher the PSNR, the lower the loss rate, and the better the video quality.

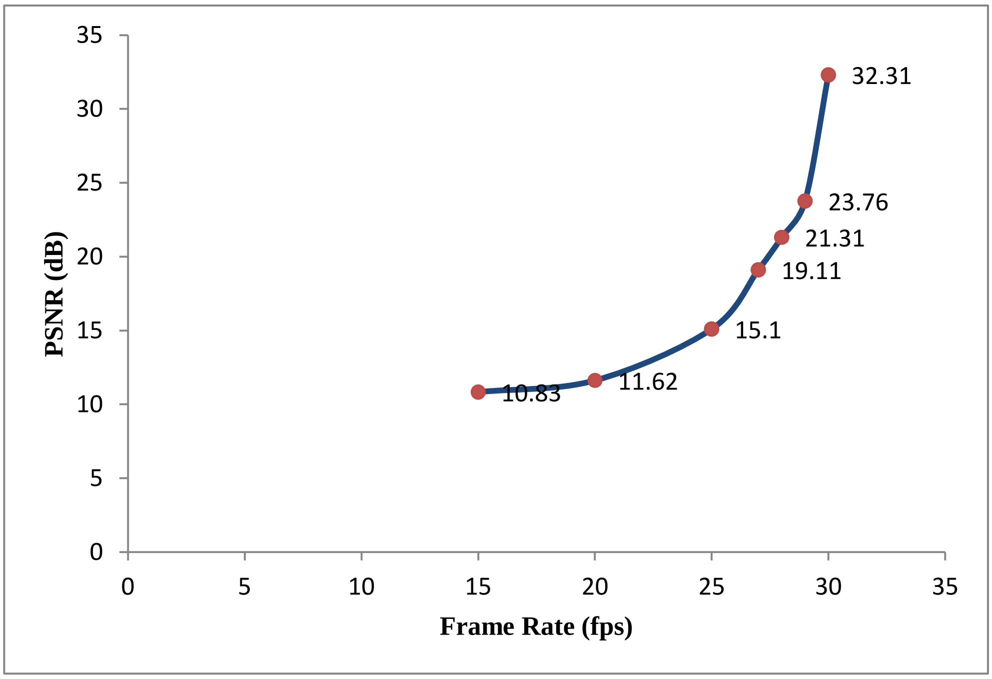


Figure 16: PSNR as a function of frame rate

## CONCLUSION

The peak signal to noise ratio, PSNR, is still the most widely used visual quality metric. It is recommended that PSNR values of smaller than 20 dB are generally not viewable and unacceptable in communications. This fact has been confirmed in this research with various simulation experiments at varying video frame rates. In order for video streaming to succeed on the internet, systems must account for the anomalies of packet loss which makes the delivery of live video streaming via the internet challenging.

We have shown that by using the PSNR as the video quality assessment parameter, the effect of packet loss on real-time video streaming over the internet is mitigated. The result of our analysis shows that the higher the average frame rate received, the higher the PSNR, the lower the loss rate, and the better the video quality. Based on this, it is observed that the PSNR values of video frame rates from 28fps to 30fps are above the recommended 20dB – meaning they are the viewable frames at the receiver.

## REFERENCES

[1] Postel J. (2012). Transmission Control Protocol. *Available: http://tools.ietf.org/html/rfc793.* Accessed: 28 September, 2012.

[2] Frederick R. and Jacobson V. (2003). RTP: A Transport Protocol for Real-Time Applications. *IETF RFC3550*

[3] Wu D., Hou Y. T., Zhu W., Zhang Y. Q., and Peha J. M. (2001). Streaming Video over the Internet: Approaches and Directions," *IEEE Transactions on Circuits and Systems for Video Technology*, 11(3), 282–300.

[4] Boyce J. and Gaglianello R. (1998). Packet Loss Effects on MPEG video sent over the Public Internet. *ACM Multimedia*

[5] Pinson, M. and Wolf, S. (2004). A New Standardized Method for Objectively Measuring Video Quality, *IEEE Transactions on Broadcasting*, 50(3): 312-322

[6] Feamster N., and Balakrishnan H. (2002). Packet Loss Recovery for Streaming Video. *12th International Packet Video Workshop.* Pittsburgh, PA.

[7] Bassey, A., Ozoumba, S. and Udofia, K. (2015). An Effective Adaptive Media Play-out Algorithm for Real-time Video Streaming over Packet Networks. *European Journal of Basic and Applied Sciences (EJBAS)*, 2(4): 1-14.

[8] Bassey, A. (2015). Development and Performance Analysis of Protocol for Multimedia Networks that Support Video Streaming. M. Eng Thesis, Department of Electrical/Electronic & Computer Engineering, University of Uyo, Uyo, Nigeria.