AN EFFECTIVE ADAPTIVE MEDIA PLAY-OUT ALGORITHM FOR REAL-TIME VIDEO STREAMING OVER PACKET NETWORKS

Anietie Bassey, Simeon Ozumba & Kufre Udofia
Department of Electrical/Electronic
& Computer Engineering, University of Uyo, Uyo
NIGERIA

ABSTRACT

The increasing demand to deliver rich multimedia content over the network has made video streaming an interesting area of research. Consequently, developments in protocols, communication and network infrastructure technologies have continued to change the concept of video streaming. However, there are a number of challenges that affect video streaming over packet networks (eg. the internet) such as bandwidth, jitter, packet losses, as well as how to efficiently perform multicast communication for multimedia content. In this paper, we propose an Adaptive Media Play-out (AMP) algorithm and model to reliably transmit packets, while reducing jitter and packet losses when streaming video in real time. The proposed AMP algorithm and model enables a valuable trade-off between the quality of service parameters and received video quality. Through simulation experiments, the performance of the AMP has been compared with the existing and conventional technique of buffering. On the basis of this study, it has been found that the AMP algorithm outperforms buffering to a large extent, in terms of buffer size, jitter, packet losses, and received video quality.

Keywords: Video streaming, AMP algorithm, Packet losses, Jitter.

INTRODUCTION

Video streaming is the delivery method used to transmit video and audio via the internet from one location to another [1]. Over the past 20 years, video streaming has evolved as an important component of many internet multimedia applications such as distance learning, digital libraries, home shopping, video on-demand (VoD), IPTV, IP telephony (VoIP), online games, video conferencing, industrial control systems, network operation support systems, etc [2]. The most significant problems in video streaming over packet networks are the unpredictable nature of wireless medium and mobile networks in terms of bandwidth, end-to-end delay and losses [3].

In this paper, analytical and simulation approaches are used to develop an Adaptive Media Play-out (AMP) algorithm and model to minimize the packet loss and jitter effects in real-time video streaming over packet networks. The analysis, as verified and validated with simulation experiments using Simulink/MatLab (in multicast scenario), shows a valuable trade-off between the quality of service (QoS) parameters and received video quality. Comparative study of the performance of buffering with the developed AMP algorithm is conducted in this paper and it has been seen from the simulation result that the proposed AMP performs better than buffering.

RELATED WORK

In previous research works, researchers have used different approaches to enhance the performance of video streaming systems in order to reduce packet losses, long delays and congestion in transmission channels. In 2004, Kalman, Steinbach and Girod [4] designed and proposed a streaming system model that buffers media at the client, and combats packet losses with deadline-constrained automatic repeat request (ARQ). For the channel, they defined a two state Markov model that features state-dependent packet loss probability.

Nick Feamster and Hari Balakrishnan [5], 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.

Furthermore, prior works 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) [6]. However, despite the aforementioned techniques for reducing and minimizing jitter and losses, the drawback of these video streaming protocols are quite enormous and therefore cannot be ignored. Whereas buffering introduces additional end-to-end delay due to large buffer size and additional storage requirements at the receiver; selective retransmission, error concealment, forward error correction (FEC), and error resilient video coding are quite cumbersome and complicated to deploy. Hence the AMP Algorithm and model (which is preferable in terms of performance, simplicity, and the QoS parameters) is proposed in this paper.

THE AMP-BASED MODEL AND ALGORITHM DESCRIPTION

The proposed Adaptive Media Play-out-Based Model is designed to allow the receiver to buffer less data, reduces buffer size and latency for a given buffer condition. It is recommended that slowing the play-out rate of video and audio up to twenty five percent (25%) is unnoticeable to the viewer, and time-scale modification is subjectively preferable compared to halting play-out or errors due to missing data. It is also found out that fast and slow play-out allows a mean latency that is smaller than the constant delay that the user would experience in the case of fixed play-out speed, for a given play-out condition. This application was explored for the case of two-way voice communication [8], and has been extended to video in this research.

The AMP Model for Video Streaming is a real-time protocol developed to work upon existing User Datagram Protocol/Internet Protocol (UDP/IP). Since UDP/IP is widely used and is a hardware independent protocol, the real-time video protocol developed can be used for communications between different systems. Figure 1 shows the block diagram of key modules in the Model. Some special hardware devices are required to construct the total system: a Moving Pictures Experts Group (MPEG) encoder, an MPEG decoder and a video camera. As soon as the input video signal comes from a video camera, it is compressed by the MPEG encoder (where MPEG is a recently established standard for full-motion video compression). The Real-time Transport Protocol (RTP) and User Datagram Protocol (UDP) are transport layer protocols that interface with the Internet Protocol (IP), a network layer protocol to process the encoded bit streams as explained below. After network transmission through the communication channel (Internet), the signal is then buffered at receiver, before it is sent to the AMP module – the module performs the AMP algorithm based on the calculation in equation 1. The received signal (optimal quality) is then decompressed by the MPEG encoder before it is sent to the video player with a pleasant viewing experience.

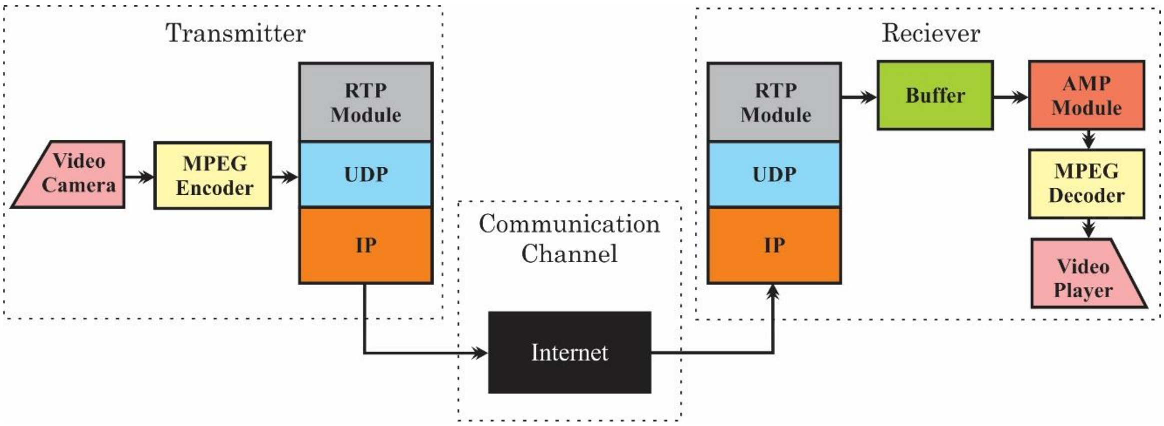

Figure 1: Block diagram of key modules in the proposed AMP model

During the network transmission, there are two software modules involved – the Real-time Transport Protocol (RTP) and User Datagram Protocol (UDP). These modules sit on top of the Internet Protocol to perform the following functions:

Real-time Transport Protocol (RTP): RTP is a new protocol for real-time video communications running on top of UDP/IP. Jitter reduction and packet loss control schemes (using buffer) are introduced to minimize their effects on the displayed stream. RTP is designed for end-to-end, real-time transfer of data streams. Real-time video streaming applications can tolerate some packet loss but require timely delivery of information. 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, RTP provide timestamps for synchronization, sequence numbers for packet loss and reordering detection, security and content identification. RTP helps 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. 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.

(i) **Sequence Number:** The frame number and the block number fields in the RTP packet header together form a RTP sequence number. The frame number is used to represent one video frame in the sequence. Since it has a size of two bytes, it permits a cyclic sequence up to 64 frames. The RTP sequence number can be used to detect frame loss, out of sequence, and duplication. Therefore, it is important to reproduce a correct sequence video at the receiver.

(ii) **Timestamps:** A solution to the variation in packets delay and synchronization of signal problems is use of timestamps whereby each packet has a time of arrival at the receiver with respect to the first packet and so on. This time is measured at the sender's end as shown in figure 3. Timestamp removes jitter with play-out buffers and provides inter-media synchronization between audio and video.

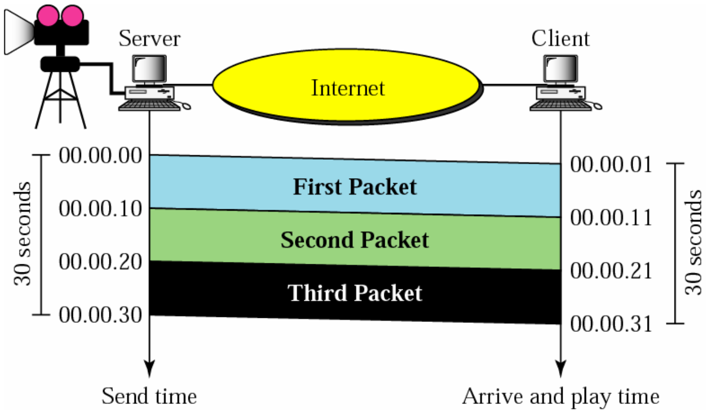

Figure 2: Time relationship in a jitter and packet loss free system

Assumption: Each packet holds 10 seconds of video information. The transfer delay equals 1 second and total duration of video transmission is 30 seconds. If this is true, then receiver sees the video with the same speed as it is transmitted; the constant transfer delay does not matter since it falls within the tolerance threshold of 1 to 15 seconds as recommended by IETF (2008).

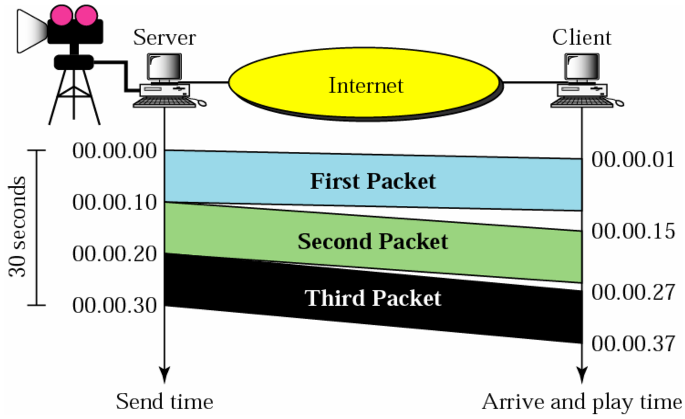

Figure 3: Variation in packets delay at the receiver

(iii) **Packet Types:** There are four packet types, described as follows:
a. XREQ (Transfer Request): Connection setup request
b. ACCEPT: Connection accept acknowledgment
c. DATA: Video data
d. FINAL: Connection tear-down request

(iv) **Connection Setup and Tear-down:** A video connection is set up by an exchange of two RTP packets between the sender and receiver. In implementation, the sender behaves as an active entity (client) and the receiver behaves as a passive entity (server). To start, the active end first sends an 'XREQ' packet. After arriving at the destination, the passive end responds with an 'ACCEPT' packet. Once the active end receives the acknowledgement, the transfer can begin. After data transmission is finished, the connection can be torn down by sending a 'FINAL' packet from the active end to the passive end.

(v) **Flow Control:** There is no flow control employed in the RTP protocol. Instead, from the video transmission model the transmission rate is fixed at frame rate. The Real-time Control Protocol (RTCP) provides the flow control instead.

(vi) **Error Detection and Control:** Error detection allows the receiver to detect the corrupted packets and drop them. Since RTP is based on the UDP/IP, error detection is simply done by the UDP checksum mechanism, which covers both the RTP packet header and data. Since we are concerned with real-time video transmission, there will be no retransmission for error detected packets.

User Datagram Protocol (UDP): UDP is preferred to the Transmission Control Protocol (TCP), since it does not require acknowledgement and is thus fast in transmission. Like TCP, it is designed to work with IP. UDP traffic enjoys a high-priority status on the internet, making it fairly smooth and can be used for low bandwidth networks. Other advantages of using UDP include, congestion control, frame rate control, multiplexing etc. UDP is used for real-time streaming due to datagram service it provides which reduces packet latency.

Since the video receiver must receive, decode, and display video frames at a constant rate, any late frames resulting from jitter produces serious problems in the reconstructed video and frames whose arrival time falls below the minimum time range will be lost. Hence there is a need to introduce jitter buffer at the receiver. Besides the additional storage requirements at the streaming receiver, buffering also introduces additional delay before playback can begin, which is undesirable in real-time communications. Based on the aforementioned problems associated with buffering, an Adaptive Media Play-out (AMP) algorithm is developed so as to minimize the buffer size and delay before playback; this in turn also reduces packet loss.

The AMP algorithm hangs on the fact that slowing the play-out rate of video up to 25% is often unnoticeable. The pre-roll delay is reduced by beginning play-out with fewer frames of media stored in the buffer. Using slowed play-out, the receiver reduces the initial frame rate so that the buffer occupancy increases over time. When a sufficient amount of frames is buffered, play-out can continue at normal rate. The AMP equation enables the receiver to start playback at the rate of 75% of the rated video frame rate (f) which is the threshold, and automatically increases playback frame rate as the buffer size (x) increases until the buffer size reaches the minimum size (x_m) that will enable the video receiver to play at its normal frame rate. The playback frame rate is given in equation 1.

$$\boldsymbol{AMP\ Playback\ frame\ rate} = \begin{cases} \left(0.25\dfrac{x}{x_m} + 0.75\right)f, & x < x_m \\ f, & x \geq x_m \end{cases} \qquad (1)$$

ALGORITHM AND FLOW CHART FOR THE PROPOSED AMP

Below is a description of the algorithm that performs AMP at the receiver. To implement this, the following two premises are adopted.

(i) A buffer is used for jitter control;
(ii) A separate hardware device for real-time MPEG decompression is assumed to be available. This allows the software to process only the protocol.

With the above description, the AMP algorithm can be studied as follows:
1. Initialize the receiver and wait for incoming frames.
2. Store arrived frames in the buffer.
3. Calculate the playback frame rate using equation 1
4. Playback video frames at the calculated frame rate in step 3.
5. Repeat steps 1 to 4 until all transmitted frames are playback.

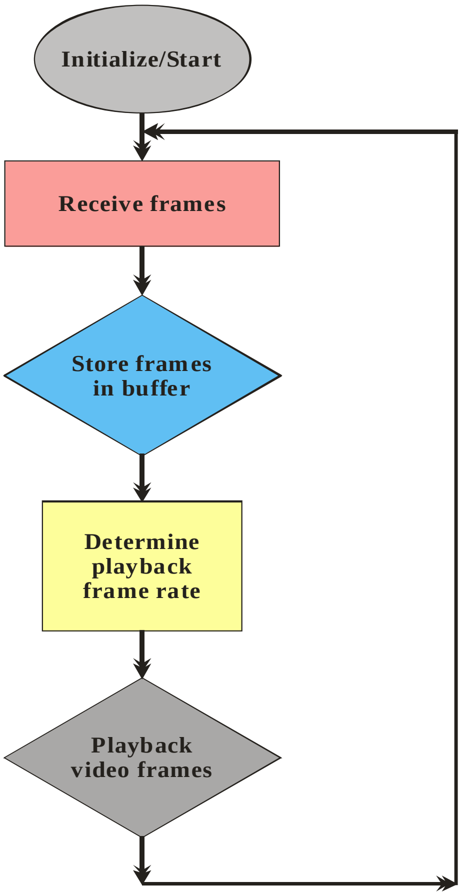

Figure 4: Flow chart for the proposed AMP algorithm

It is observed that the proposed AMP-Based Model has the following advantages:

(i) With the use of a buffer, jitter and latency has been reduced, hence, packet loss is minimized.

(ii) AMP reduces the buffer size needed for acceptable video quality. This is critical in real time communications.

(iii) The buffering latency most noticeable to the viewer is pre-roll delay (i.e the time it takes for the buffer to fill up with data and for play-out to begin after the viewer makes a request). The AMP algorithm can reduce pre-roll delays by beginning play-out with fewer frames of media stored in the buffer. Using slowed play-out at 25% (enabled by the algorithm equation); the viewer will not notice the play-out delay.

(iv) With this algorithm, faster play-out can be used during good channel periods to eliminate any excess latency accumulated with slowed play-out.

IV DESCRIPTION OF THE SIMULATION SETUP

Table 1: Simulation Parameters

Parameters	Value
Simulator	Simulink/MatLab
Protocol	AMP
Application Type	Video Streaming
Simulation Time	20 secs
Frame Rate	15fps, 20fps, 25fps
Resolution	320 x 240 pixels
Performance Parameter	Ave. frame rate, buffer size, latency, playback delay

Simulink/MatLab was used to validate the AMP Algorithm. The video (in MPEG format) was simulated at a short duration of 20 seconds at a resolution of 320 x 240 pixels at different frame rates. The MPEG video was 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 as shown in the transmitter block of the proposed AMP model in figure 1. After transmission via the channel (internet), the received video streams were then decoded (compressed) from a ratio of 4:2:0 to the original sampling ratio of 4:4:4 before it was sent to the buffer at the receiver. The signal is usually displayed on the video display – computer devices such as laptops, tablets, smart phones, etc as the received video. The effects of packet losses and jitter at various frame rates are as shown in figure 5, 6 and 7.

Figure 5 shows the transmitted and received videos at a frame rate of 15fps. The received video shows the effect of jitter and packet losses – a very poor video is seen after the 20 seconds duration of streaming due to variation in delay and packet losses. Same poor video quality is seen in figure 6 at 20fps, but better than the received video in figure 5 with less delay and packet losses

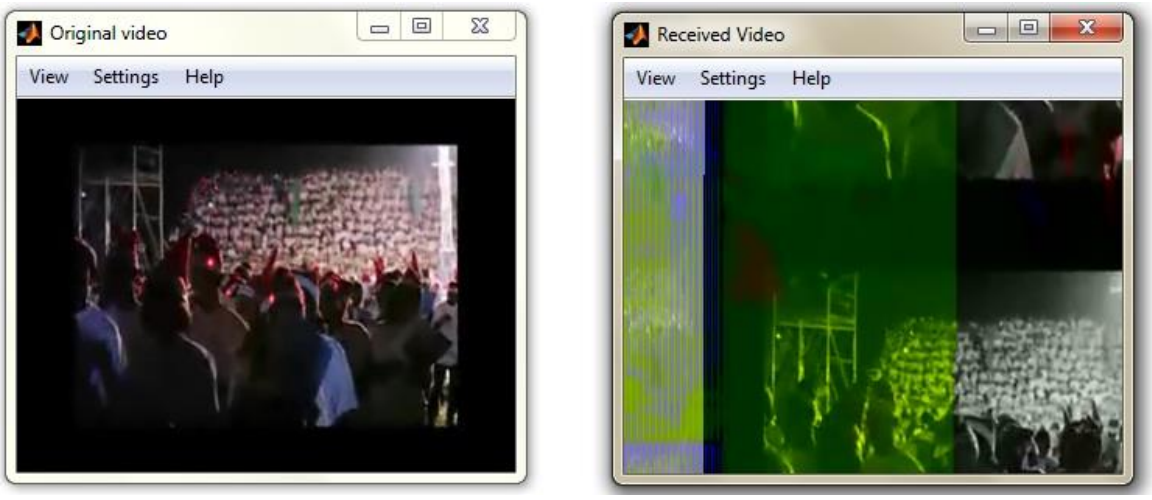

Figure 5: Transmitted and received videos at 15 fps showing jitter & packet loss effect

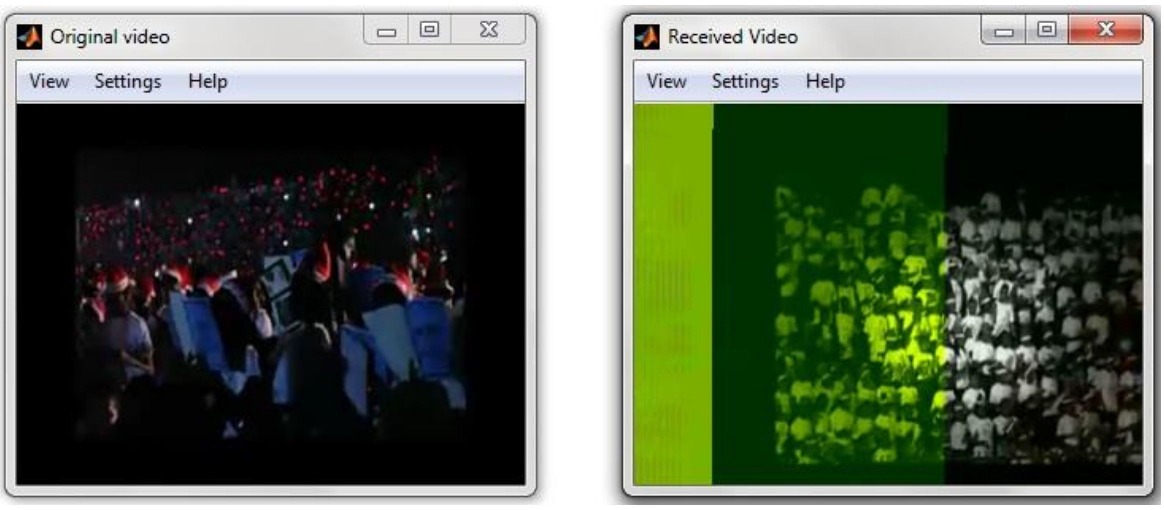

Figure 6: Transmitted and received videos at 20 fps

Figure 7 shows the transmitted and received videos at 25fps with exact resemblance; meaning there was neither packet losses nor jitter. Based on the simulation experiment, it was concluded that packet loss results in degradation of video quality at the receiver. More packet losses amounts to a lower threshold PSNR, and poor video quality – the higher the PSNR, the lesser the packet loss rate, the better the video quality.

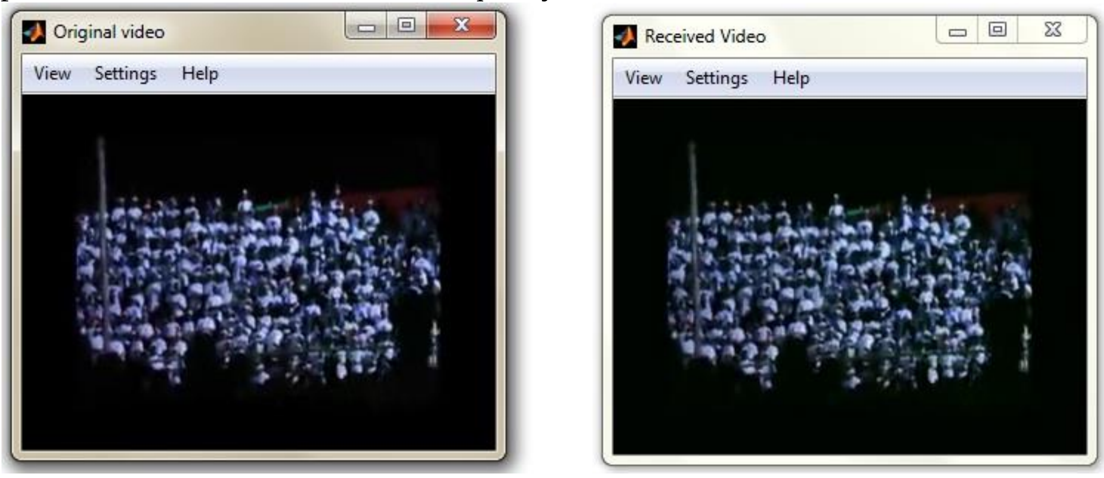

Figure 7: Transmitted and received videos at 25 fps

SIMULATION RSULTS

The description of the proposed AMP protocol and simulation setup are explained considering various QoS parameters such as average frame rate, buffer size, latency, and playback delay. The data obtained from the simulation experiments were gathered and graphs plotted to demonstrate the effects of these parameters, their relationships, and solutions provided in this paper. The results of the analysis as validated with several simulation experiments have shown tremendous improvement on video quality experience.

Table 2: Summary of Results for the proposed AMP Algorithm

Jitter (%)		Playback Delay (secs)	Latency (secs)	Buffer Size (Frames)	Ave. Frame Rate (fps)
0	Without Buffer	0.2	0.1	0	25
	With Buffer	0.2	0.1	3	25
	With AMP	0.2	0.1	1	25
5	Without Buffer	0.2	10	0	10.4
	With Buffer	3.6	3.5	83	25
	With AMP	0.8	1.6	14	24
10	Without Buffer	0.2	10	0	10.4
	With Buffer	5.4	5.3	117	25
	With AMP	0.9	2.4	17	22.6

ISSN 2059-3058

15	Without Buffer	0.2	10	0	10.4
	With Buffer	7.3	7.2	155	25
	With AMP	1.2	3.0	19	21.4
20	Without Buffer	0.2	10	0	10.4
	With Buffer	9.5	9.4	191	25
	With AMP	1.9	3.9	24	20.1

From table 2, it is observed that the values for the system without buffer remain the same irrespective of the increasing effects of jitter; hence we shall only consider systems with buffer and AMP in explaining our results since our proposed system is only utilizing the concept of buffering and Adaptive Media Play-out.

Average Frame Rate

Average frame rate is the mean playback frame rate at the receiver. The higher the average frame rate received, the better the video quality. Table 3 shows the relationship between average frame rate in a system with buffer and that of AMP compared to the threshold of 18.75 (i.e. 75% of the transmitted video frame rate of 25 fps).

Table 3: Average frame rates in a system with buffer and AMP

Jitter (%)	With Buffer	With AMP	Threshold
0	25	25	18.75
5	25	24	18.75
10	25	22.6	18.75
15	25	21.4	18.75
20	25	20.1	18.75

As shown in figure 8, the result indicates that AMP is a better is system compared to a buffered system since it begins play-out at a rate of 75 percent of the transmitted video during bad network conditions instead of waiting for the buffer to fill up as in buffering (which introduces additional delay that may halt play-out at the receiving end). However, both systems do not fall below the threshold as shown in the graph.

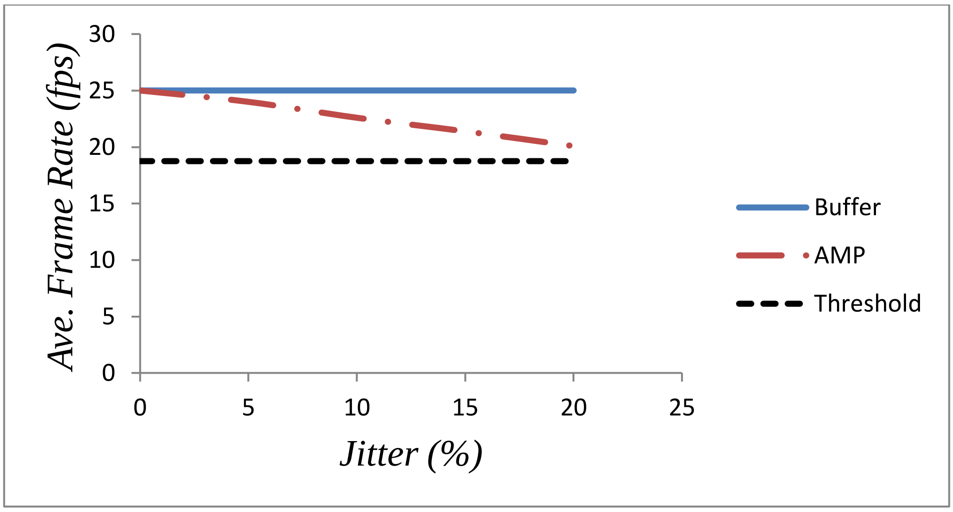

Figure 8: Average frame rate and jitter in a system with buffer and AMP

Buffer Size

Besides the additional storage requirements at the streaming receiver, buffering also introduces additional delay due to large buffer size (as evident in table 3) which must always be filled up before playback can begin – this is undesirable in real-time communications. Based on the aforementioned problems associated with buffering, an Adaptive Media Play-out (AMP) algorithm is developed so as to minimize the buffer size and delay before playback; this in turn also reduces packet loss.

Table 4: Buffer size in a system with buffer and AMP

Jitter (%)	With Buffer	With AMP
0	3	1
5	83	14
10	117	17
15	155	19
20	191	24

The AMP-Based algorithm hangs on the fact that slowing the play-out rate of video up to 25% is often unnoticeable. The pre-roll delay is reduced by beginning play-out with fewer frames of media stored in the buffer. Using slowed play-out, the receiver reduces the initial frame rate so that the buffer occupancy increases over time. When a sufficient amount of frames is buffered, play-out can continue at normal rate. AMP enables the receiver to start playback at the rate of 75% of the transmitted video frame rate, reduces buffer size by so doing, and automatically increases playback frame rate as the buffer size (x) increases gradually until it reaches the minimum size (x_m) that will enable the video receiver to play at its normal frame rate.

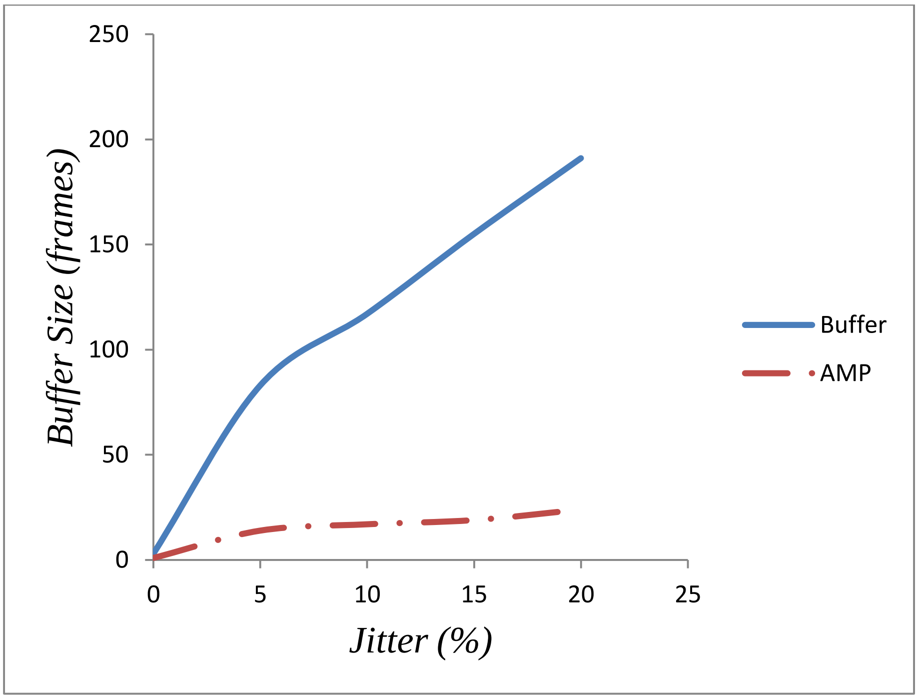

Figure 9: Buffer size in a system with buffer and AMP

Latency

In video streaming, latency is the amount of time between the instant a frame is captured and the instant that frame is displayed. Applications and network devices can cause latency. Latency is expressed in terms of time units (in seconds or milliseconds), but the biggest contributors to video latency are the processing stages that require temporal storage of data, i.e. short term buffering with a smaller buffer size which the proposed AMP algorithm enables. Because of this, video streaming system engineers tend to measure latency in terms of the buffered video data.

Table 5: Latency in a system with buffer and AMP

Jitter (%)	With Buffer	With AMP
0	0.1	0.1
5	3.5	1.6
10	5.4	2.4
15	7.2	3
20	9.4	3.9

Converting from frames to time depends on the video's frame rate. For example, a delay of one frame in 25 frames-per-second (fps) video corresponds to 1/25 of a second (0.04secs). Again, converting from video lines time requires both the frame rate and the frame size or resolution. A 720p HD video frame has 720 horizontal lines, so a latency of one line at 25fps is 1/(25x720) = 0.00014secs which is quite low and unnoticeable to the human eyes. Low latency is a design goal for any real-time system such as live video streaming which is the focus of this dissertation. This is shown in the table 4 when comparing the latency in a buffered system to the proposed AMP-based system at various percentages of jitter. From table 5, the graph of Latency against Jitter is as shown in figure 10.

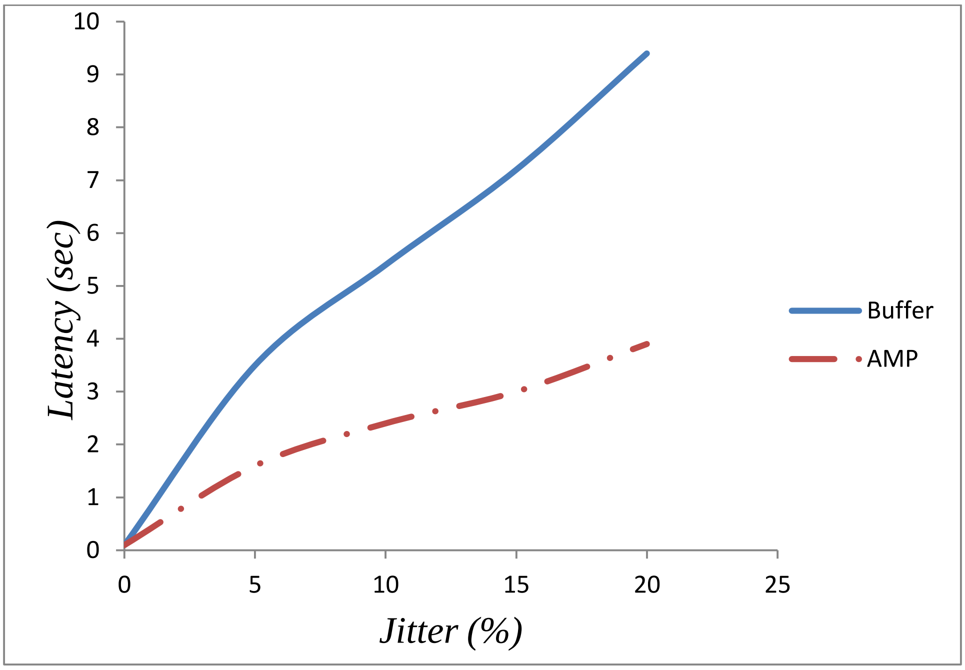

Figure 10: Latency in a system with buffer and AMP

Playback Delay

Playback delay is the time between the generation of a packet until it is played at the receiver. It is a natural delay metric for in-order playback, in-order in the sense that the video frames are played back in the order that they are received. In other words, the higher the playback delay, the more the video quality is degraded. Table 6 shows the playback delay values at different percentages of jitter from 5 to 20%.

Table 6: Playback delay in a system with buffer and AMP

Jitter (%)	With Buffer	With AMP
0	0.2	0.2
5	3.6	0.8
10	5.4	0.9
15	7.3	1.2
20	9.5	1.9

As a solution, we have shown how the proposed AMP protocol is utilized to reduce playback delay in video streaming system compared to the technique of buffering as shown in figure 11.

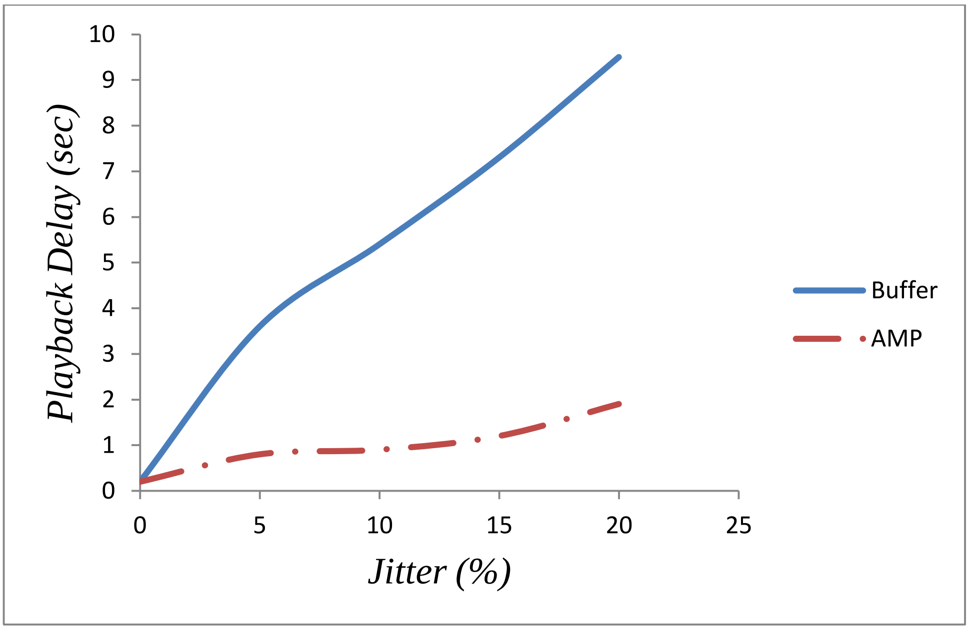

Figure 11: Playback delay and jitter in a system with buffer and AMP

CONCLUSION

In this work, we have analysed the performance of AMP-based model and algorithm as it has been developed and used to address packet loss and jitter in real-time video streaming over packet networks of which the internet is a concrete example. Four quality of service parameters (average frame rate, buffer size, latency, and playback delay) have been used to evaluate system performance compared with existing techniques. Based on the various simulation experiments done in this study, it has been found that the AMP algorithm outperforms buffering to a large extent, specifically, in terms of buffer size, jitter packet losses, and received video quality.

Therefore, the proposed AMP is recommended as an effective real-time video streaming protocol.

RECOMMENDATIONS FOR FUTUREWORK

The work presented in this paper can be furthered in a number of future research directions as recommended below:

1. In this research, the results of the analysis and evaluation through simulation experiments can be used to make video streams more resilient to packet losses and to reliably transmit MPEG video in real time instead of using the techniques of FEC, and Selective Retransmissions which are quite complicated. Further work should also take into account to study the channel characteristics for the available bandwidth and end-to-end delay to predict the expected packet loss at the receiver. In addition, other controllers could be studied to improve the number of discarded frames and decrease the wasted header packets differences with various FEC parity adaptation algorithms.
2. The implementation with one specific set-up is not enough to provide definitive conclusions. This applies to the work that has been presented in this research. Further research from this work includes an implementation of more advanced test scenarios than that presented in this work. In addition, future work could also be able to explore advanced FEC adaptation schemes against a real-world experimental test-bed with a wide variety of network setup.

REFERENCES

[1] Khalifa N. and H. Elmahdy (2009). The Impact of Frame Rate on Securing Real-Time Transmission of Video over IP Networks. *IEEE ICNM,* pp. 57 – 63.

[2] Girod B, Chakareski J., Kalman M., Liang Y., Setton E., and Zhang R (2002). Advances in Network-Adaptive Video Streaming.*2002 Tyrrhenian Inter. Workshop on Digital Communications*

[3] Joshi, Bais, and Dubey (2013). Analysing Video Streaming Quality over Different Routing Protocols on Mobile Ad-hoc Network. *IJARCSSE, Vol. 3, Issue 10, October 2013.*

[4] Kalman M., Steinbach E., and Girod B. (2004). Adaptive Media Play-out for Low Delay Video Streaming over Error-Prone Channels. *Preprint, to appear in IEEE Trans.Circuits and Systems for Video Telephony.*

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

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

[7] Bolot J.C. & T. Turletti (1996).Adaptive Error Control for Packet Video in the Internet. *Proceedings of International Conference on Internet Protocols.*

[8] Liang Y. J., Färber N., and Girod B. (2001). Adaptive Play-out Scheduling Using Time-Scale Modification in packet voice communication. *Proc. IEEE Int.Conf. on Acoust., Speech and Sig. Proc.*, Salt Lake City.

[9] Ramjee R., Kurose J., Towsley D., and Schulzrinne H. (1994). Adaptive Play-out Mechanisms for Packetized Audio Applications in Wide Area Networks. *IEEE Computer and Communications Societies on Networking for Global Communication, pages 680 – 688,* Los Alamitos, CA, June, IEEE Computer Society Press.

[10] Zhang Q., Wang G., Zhu W., and Zhang Y. Q. (2001). Robust Scalable Video Streaming Over Internet with Network-Adaptive Congestion Control and Unequal Loss Protection. *Proc. Packet Video Workshop,* Kynongju, Korea.

Authors Biography

Anietie Bassey is an Electrical Engineer at Ministry of Works, Akwa Ibom State, Nigeria. He obtained his bachelor of engineering (B. Eng) degree in Electrical/Electronic Engineering from University of Uyo, Akwa Ibom State, Nigeria in 2010 and a master of engineering (M. Eng) degree in Electrical/Electronic Engineering in 2015 with specialization in Electronic/Communication Engineering in the same University. His research interests include, Video Streaming and Multimedia Applications, Satellite and Mobile Communication, Wireless Communication and GSM Technology, Broadcast Technology, and Microwave Engineering.

Simeon Ozoumba is a senior lecturer in the Department of Electrical/Electronic & Computer Engineering, University of Uyo. Simeon received his B. Eng degree in Computer Engineering from Lagos State University, Lagos State, Nigeria and the M. Eng and Ph. D (Computer Software) degrees from the Federal University of Technology (FUTO), Owerri, Imo State, Nigeria. His current research interests are in the areas of software engineering, computer programming, computer architecture and systems security, protocols, and QoS Support for Multimedia over wireless internet. He has published many papers in both local and international journals in Computer and Software Engineering.

Kufre Udofia is a lecturer in the Department of Electrical/Electronic and Computer Engineering, University of Uyo. Kufre received his B. Sc degree in Physics from the University of Uyo, Akwa Ibom State, Nigeria. He has two Masters' degrees to his credit, M. Sc in Mechatronic Systems Engineering and Electronic Mobile Communication from the Lancaster University and South Wales University (formally the University of Glamorgan) respectively in the United Kingdom. He has a Ph. D in Satellite Communication from South Wales University, Pontypridd, United Kingdom. He is presently the Postgraduate Co-ordinator in the department of Electrical/Electronic & Computer Engineering, University of Uyo. His research interests include, broadband satellite services and antennas and radio-wave propagation, electronic mobile communication, and mechatronic systems engineering. He has published many papers in both local and international journals in Communications Engineering.