ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 12, December 2013

Improving a performance of MPEG video streams with different UDP variants

Prof. Rajesh K. Ahir

Assistant Professor, Department of Computer Engineering, G. H. Patel College of Engineering & Technology,

Vallabh Vidyanagar, Anand, Gujarat

Abstract : The Internet Protocol is becoming a very dominant in computer network technology for video transmission. Video streaming is becoming an increasingly significant component of IP network traffic. Video streaming refers to the real-time transmission of stored video. Initially video was captured and transmitted in analog form i.e. MPEG4 frames are transmitted. Normally UDP (User Datagram Protocol) is used for Media Streaming. With increased deployment of services such as IPTV and Video on Demand (VoD), we have to take care about Quality of Service (QoS). This research work mainly putting the improved performance of different UDP variants with current UDP performance for MPEG video streams. It is found that UDP variants are far performs better as compare to UDP.

Keywords: UDP, QoS, GOP, UDT

1. INTRODUCTION

The best effort model for current Internet becomes inadequate to face the requirement of best multimedia streaming[1] performance. Many of streaming applications are based on the delivery of continuous media content. Thus, the key point to success of such applications is the delivery quality of video and audio streams. In video transmission over the network, both encoding and transmission process affect the performance of streaming applications. Video streams are first compressed and then it will transfer to the network. Compression is needed to reduce the storage space and bandwidth requirements of digital video. UDP[4][6] is an unreliable and connection-less protocol because it is unidirectional unlike Transmission Control Protocol (TCP)[3] which is connection-oriented and bi-directional in nature. Since UDP is unidirectional, so our aim was to compile a comprehensive set of data describing the performance of UDP variants over networks in terms of packet loss analysis, packet delay, jitter, and throughput. There is absolutely no guarantee that the datagram will be delivered to the destination host. Not only the datagram can be undelivered, but it can be delivered in an incorrect order. On the basis of the foregoing, there is the need to carry out a critical analysis of UDP and its variants like MUDP[7], RBUDP[10], UDT[9] and Tsunami[11] performance on the network.

2. Background on video streaming or transmission

MPEG[5] encodes video ass a sequence of frames. Usually video has a high degree of redundancy, i.e. information in successive frames is highly correlated. Standard MPEG encoders generate three types of compressed frames: **I** (Intra-coded), **P** (Predictive-coded) and **B** (Bi-directional-coded) as shown in fig 1. An **I** frame is intra coded, having no dependency on any other frames. MPEG uses motion prediction and interpolation techniques to reduce the size of intermediate frames. Two types of motion prediction are used: Forward prediction, where the previous frame is used as a reference for decoding the current frame, and Bi-directional prediction, where both past and future frames are used as a reference. This last technique provides a better compression. The encoding of **P** frames uses Forward prediction and encoding of **B** frames are uses Bi-directional prediction. Normally **I** frames are larger in the size, followed by **P** frames and finally **B** frames.

The video sequence may be decompressed into smaller units which are coded together. Such units are called GOP (Group of Pictures) as shown in fig 1. Each GOP holds a set of frames or pictures that are in a continuous display order. GOP is a set of consecutive frames that can be decoded without any other reference frames. Usually in GOP there are 12 to 15 frames are combined.

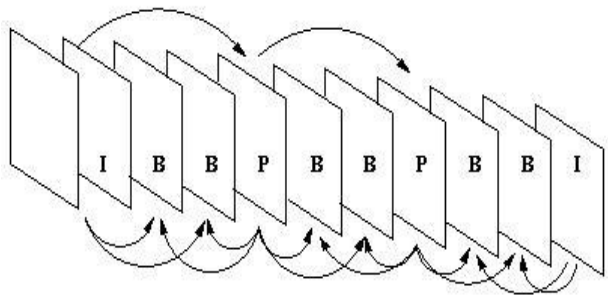

Fig. 1 GOP structure for MPEG4[4]

Video delivery by video streaming attempts to overcome the problems associated with file download, and also

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 12, December 2013

provides a significant amount of additional capabilities. The basic idea of video streaming is to split the video into parts, transmit these parts in succession, and enable the receiver to decode and playback the video as these parts are received, without having to wait for the entire video to be delivered. Video streaming can conceptually be thought to consist of the follow steps:

1) Partition the compressed video into packets
2) Start delivery of these packets
3) Begin decoding and playback at the receiver while the video is still being delivered

Video streaming enables simultaneous delivery and playback of the video. This is in contrast to file download where the entire video must be delivered before playback can begin. In video streaming there usually is a short delay (usually on the order of 5-15 seconds) between the start of delivery and the beginning of playback at the client. Video streaming provides a number of benefits including low delay before viewing starts and low storage requirements since only a small portion of the video is stored at the client at any point in time. The length of the delay is given by the time duration of the buffer, and the required storage is approximately given by the amount of data in the buffer.

3. Simulation Experiments

In this section, we discuss the different transport protocols to evaluate their performance for media streaming applications.

1. Packet delay (in %)

Delay refers to the time taken for a packet to be transmitted across networks from source to destination.

$$\textbf{Packet Delay} = T_d - T_s$$

Where T_d: time at which the packet is arrived at the destination

T_s: time at which the packet is transmitted at the source

2. Jitter

Jitter is a fluctuation of end to end delay from one packet to the next packet of connection flow. If the transmit time between two packets are not same then jitter effect is present in the transmission.

$$\textbf{Jitter } J = | D_{i+1} - D_i |$$

Where D_{i+1}: Delay of $(i+1)^{th}$ packet

D_i: Delay of i^{th} packet

3. Packet loss

Packet loss is where networks traffic or packet fails to reach its destination in timely manner.

$$\textbf{Packet loss} = (P_s - P_r) / P_s$$

Where P_s: no. of packets sent by the sender

P_r: no. of packets successfully received at the receiver.

4. Throughput (Kbps)

Throughput is the rate at which networks sends or receive the data.

Throughput = total transmitted bits / observation duration

Where duration: timeEnd – timeBegin

- The time corresponds to the first matching records (timeBegin); i.e., the time that the first data packet leaving node fromNode arrives the node toNode.
- The time corresponds to the last matching records (timeEnd); i.e., the time that the last data packet leaving node fromNode arrives the node toNode.

3.1 Research plan

Analysis of topic, data collection and scheduling method applied in accordance with the need of this study. Systematic Simulation Study was customized in order to perform simulation for the UDP and MUDP protocol. In first step generate the media or video traffic at application layer in NS2[6]. In second step apply that media traffic to UDP and MUDP and finally measure the performance in terms of packet delay, packet loss, jitter and throughput.

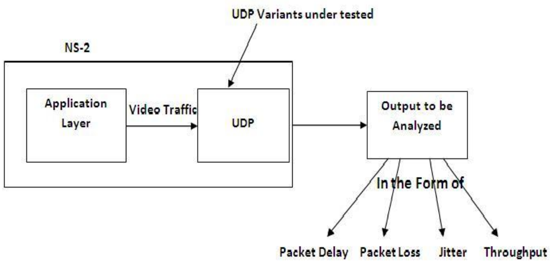

Fig.2 Implementation plan

3.2 Experimental Evaluations

As mention in the introduction, the goal of our study is to compare performance of UDP and its variants in the form of jitter, packet delay, packet loss and throughput from the user perspective. Therefore, in our evaluation we considered only user-perceived performance indices for the live streaming media applications (MPEG4).

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 12, December 2013

(i). Packet Delay:

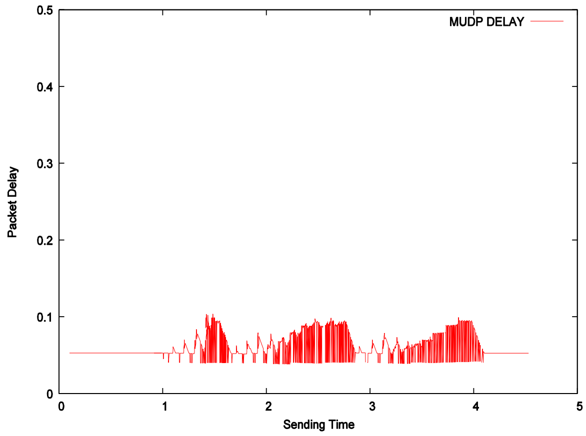

Fig. 3 Packet Delay of Modified UDP

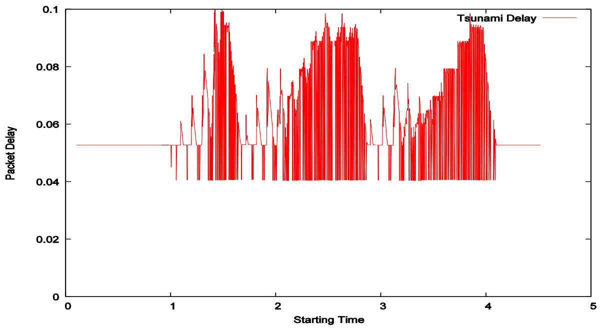

Fig. 7 Packet Delay of TSUNAMI

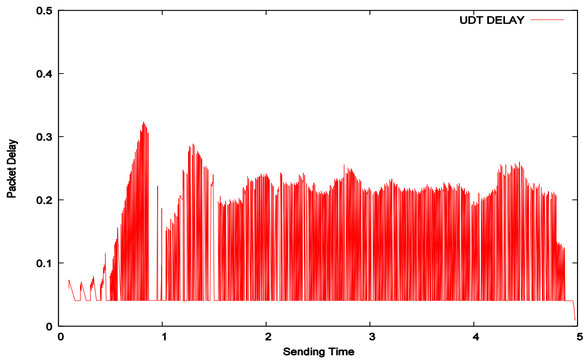

Fig. 4 Packet Delay of UDT

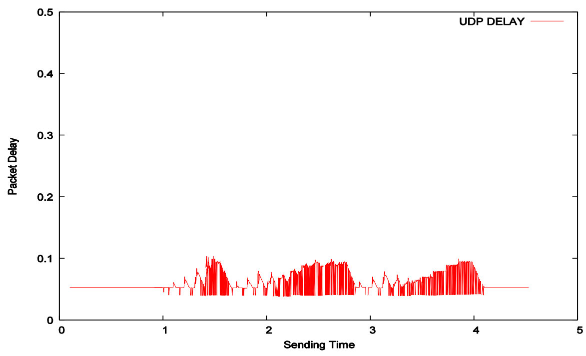

Fig. 5 Packet Delay of UDP

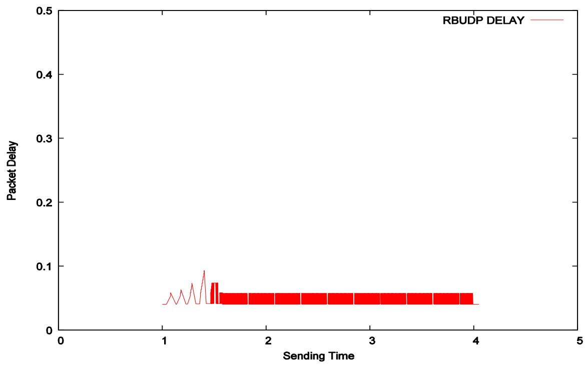

Fig. 6 Packet Delay of RBUDP

(ii). Jitter:

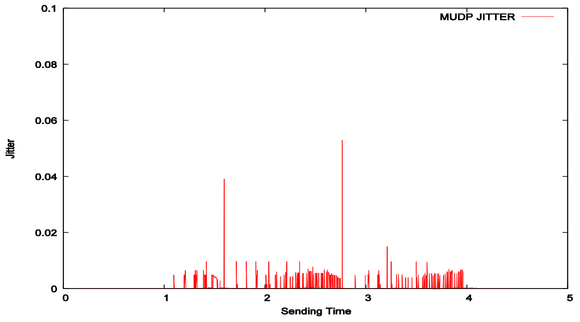

Fig. 8 Jitter Effect of Modified UDP

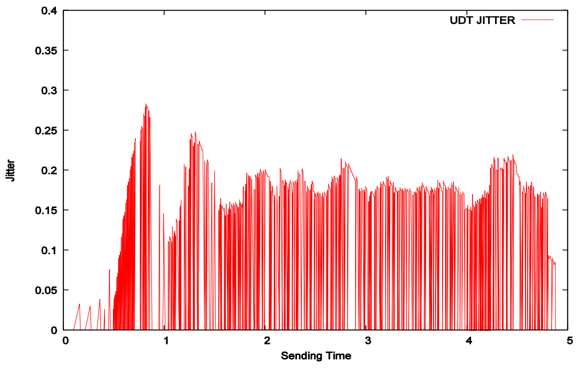

Fig. 9 Jitter Effect of UDT

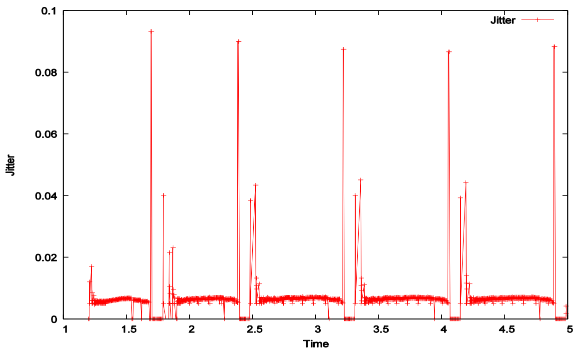

Fig. 10 Jitter Effect of UDP

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 12, December 2013

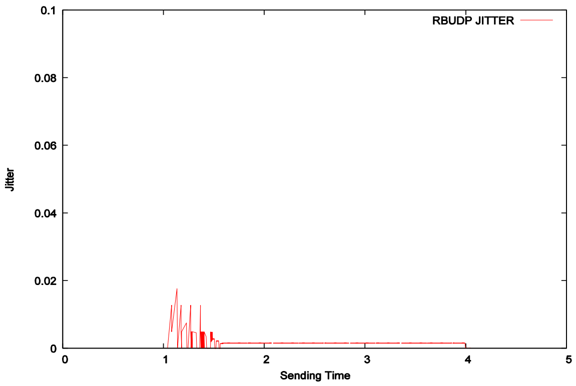

Fig. 11 Jitter Effect of RBUDP

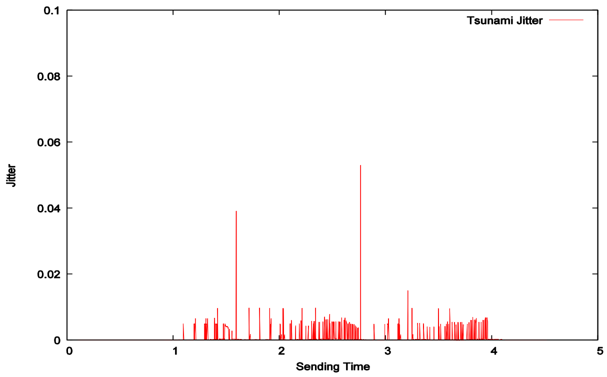

Fig. 12 Jitter Effect of TSUNAMI

(iii). Packet Lost Ratio (in %):

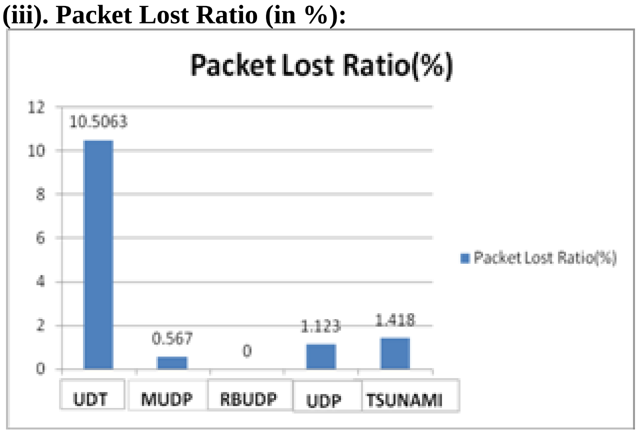

Fig. 13 Packet Lost Ratio of all UDP Variants

iv). Throughput (Kbps):

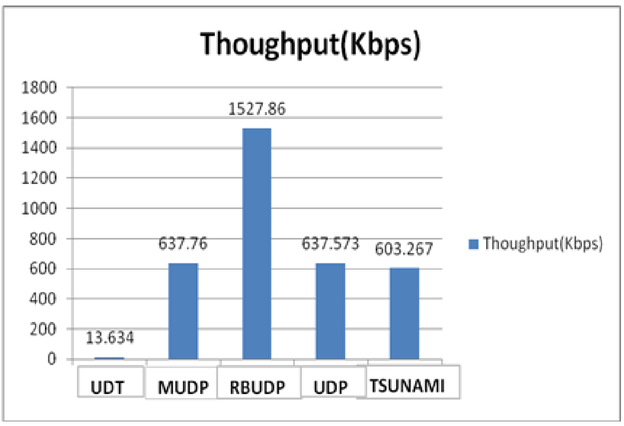

Fig. 14 Throughput (Kbps) of all UDP Variants

3.3 Analysis of Results

In this section, we present the experimental results for various UDP variants to evaluate the effectiveness of protocols for media streaming applications. We compare result of all UDP variants in the form of packet delay, packet Loss, jitter and throughput. A figure 3 to 7 expressed the packet delay for all UDP variants. From these figures we can observe that there is more packet delay for MUDP, UDP, UDT and Tsunami as compare to the RBUDP. So, RBUDP provides the better performance in terms of packet delay for media streaming applications. As we analyzing the figure 8 to 12 there are jitter effect variations for Standard UDP, Modified UDP, Reliable Blast UDP, UDT and Tsunami protocols. From the figure we can observe that jitter for UDT and standard UDP is more than the RBUDP MUDP and Tsunami.

A measurement of the packet loss is in term of percentage. From figure 13; we can observe that the packet loss is more for UDT protocol as compare to the MUDP and UDP protocol. We got the best result for the RBUDP protocol and Tsunami. So, RBUDP provides the less or none packet loss for the media streaming applications. We can calculate the network throughput in form of Kbps. From the figure 14, we can observe that the throughput for RBUDP is more than the other UDP variants. Throughput for Tsunami protocol is also average as compare to other variants.

REFERENCES

[1] Boaz Ben Moshe et al. “Analysis and Optimization of the Live Streaming for Over the Top Video”, IEEE Workshop on Emerging Densely Connected Networks (EDCN), 2011 IEEE.
[2] Chi-Fai Wong et al. “TCP Streaming for Low-Delay Wireless Video”, 2nd Int’l Conf. on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine 2005) 0-7695-2423-0/05, 2005 IEEE.
[3]“User Datagram Protocol”, R*FC768*, http://en.wikibooks.org/wiki/Computer_Networks/UDP
[4] MPEG4 http://www.chiariglione.org/mpeg/standards/mpeg-4 [Online]
[5] Stephan Brumme, "An Improved UDP Protocol for Video Transmission Over Internet-to-Wireless Networks", Advanced Data Communications at the University of Technology, Sydney, 2004.
[6] Marc Greis‘Tutorial Network Simulator NS2 [Online] Available http://www.isi.edu/nsnam/ns/tutorial/
[7] MUDP http://140.116.72.80/~smallko/ns2/tool_en.htm [Online].
[8] Al-samarraie hosam1 et. al. “Comparative Study of Media Streaming Protocols over Wireless IP Network”, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.7, July 2010.
[9] Danilo Valeros Bernardo et. al., “A Pragmatic Approach: Achieving Acceptable Security Mechanisms for High Speed Data Transfer Protocol-UDT”, International Journal of Security and Its Applications Vol. 4, No. 4, October, 2010.
[10]A Report, “High Performance International Internet Services between Research and Education Institutions in the United States and Europe/Israel”, National Science Foundation, Directorate for Computer Information Science and Engineering Division of Advanced Networking Infrastructure & Research (ANIR), NSF Cooperative Agreement No. ANI-9730202 Annual Report (April 1, 2002−March 31, 2003).
[11] Mark R. Meiss et. al. “Tsunami: A High-Speed Rate-controlled Protocol for File Transfer”, Indiana University.
[12] Yunhong Gu et. al. “Optimizing UDP-based Protocol Implementations”, Third International Workshop on Protocols for Fast Long-Distance Networks (PFLDNet 2005), Lyon, France, February 3 – 4, 2005.