Impact Factor (SJIF): 4.542

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

Volume 4, Issue 4, April-2017

Pre-Synchronization Method With ViRED Queue Management Technique For Better Result At Client Side

Krutika A Shukla
Student, L.J.I.E.T, Surkhej, Ahmedabad, Gujarat, India

Abstract- Real time multimedia traffic is increasing day by day, online streaming is more beneficial in every term of video management system. Synchronization between audio and video is must for good result. Ever we have to take buffer and, buffer management must done. Live video streaming should have less jitter, less packet loss ratio for better output, plus improve signal to noise ratio for better resolution. To add some more features in video streaming RTP/RTSP/RTCP protocol works together. RTSP protocol used on client side to establish connection. Data transmission done by RTP protocol and the RTCP protocol used to send and receive reports about the network between client and server. Pre-synchronization algorithm used for intra-media synchronization at client side. Client side used to modular and try to support all available cameras and open source sever. So, by using this method we can get better QoS of live and on demand streaming at any computer platform. With Pre-synchronization method we are using ViRED queue to managing buffer. After receiving packets it's very much necessary to storing and processing packets done in some particular manner.

I. INTRODUCTION

Before developing video streaming when people want to watch some video from difference place or want to communicate through video they just send whole video file, and end user have to first download whole file then and then he/she could watch whole video.

That was very time consuming. Even don't support real-time. So, at that time real-time video streaming concept established. In real-time video streaming sender compress the whole video and send it to receiver, and at the time receiver receive the video packets it start playing, receiver don't have to wait till whole video get download.

There are lots of multimedia streaming applications or services, like online internet calling, live video conferencing and transmitting only audio over internet. These all became very popular now a day in between millions of users over internet. There are many protocols work on it. But still to provide much better result of real-time applications we have to do some sort of modifications or extend current real-time protocols are necessary. To minimize end-to-delay, and to provide accurate synchronization between audio and video media packets and give feedback to get details about quality of service are very important for better result of video streaming. RTP is a real-time transport protocol. RTP provides end-to-end communication. RTP most probably used UDP protocol, so misleading packets during transmission can possible. RTP doesn't have any guarantee of on time packet delivery; even it doesn't transmit packets in original sequence. Because it is real-time multimedia applications, it needs appropriate timing for data transmission and for video playing. So RTP provides time-stamping for each and every packet, sequence no, and to solve every kind of timing issues use some other mechanisms. Even we can say RTP is used only when user send request to play video and video will play from starting point to ending point without any additional feature. User can only do play request at the stating and tear down the video before it get stop. [9]

RTSP protocol uses TCP for communication. In RTSP acknowledgement received after every single packet or message delivery. RTSP supports some operations and services that are OPTIONS, DESCRIBE, ANNOUNCE, SETUP, PLAY, PAUSE, TEAR DOWN, GET_PARAMETER, SET_PARAMETER, REDIRECT, and RECORD. Even RTSP server needs to keep up all above states. In RTSP sever or client any one can issue request. RTSP provides interoperability between any client and server [7].

RTCP is real time control protocol; it generates the report about network information and other useful information about sender. It sends five types of reports: receiver report, report, source and destination items, BYE packets and application specific functions [2].

II. RELATED-WORK

Streaming is the process of playing a file while it is still downloading. Streaming technology, also known as streaming media, lets a user view and hear digitized content such as video, sound and animation s it is being download. Various media are used in internet, not only for obvious purpose of entertainment application, but for other applications such as video conferencing, video archives and libraries, remote learning, multimedia presentations and of course, video on demand. We will focus on streaming video which is a sequence of "moving images" that are sent over the internet and are seen by the viewer as they arrived [9]

Wang, Weiqiang Wu, Qinyu Zhang, Changjian Zhang have discover pre-synchronization technique in Pre-synchronization of AVS Audio-Video Based on Real-Time Transport Protocol in 2012, a new pre-synchronization method for synchronizing audio and video in this paper. We realize the intra-media rapid synchronization and eliminate the inter-media skew without increasing extra end-to-end delay before unpack RTP packet. Limitations in the current HDFS in-memory caching implementation is that users should manually specify what files should be cached and uncached. For an input file of each application that users want to cache, they must use a command to add the file to the in-memory cache. [1]

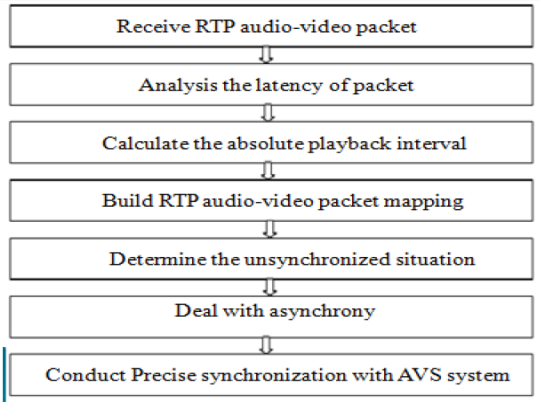

Fig: Architecture of Pre-Synchronization Method [1]

And they have conclude There were less work done on buffer design, characteristic AVS system and functions of RTP header. The pre-synchronization method is able to strike a satisfactory synchronization effect and without increasing the end-to-end delay. In view of the increasing popularity of internet applications, the results in this study are very timely and important for achieving better QoS in many multimedia and networking applications [1].

III. PROPOSED METHOD

In below figure, we can see the actual process is done at client side.

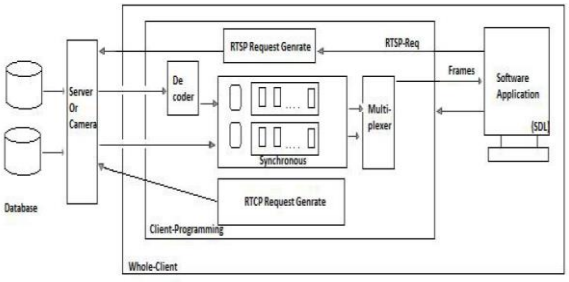

Fig: Sub-level Architecture

When renderer send request for video, client generate request and send it to server. And server and client will communicate with each other, this whole process done by RTSP protocol. In other words we can say the establishment of connection phase done by RTSP protocol.
After that, server starts to send all packets without wasting any time. Because our system is real-time system, and can't tolerate any kind of delay so, for packet transmission RTP protocol used at server side. And client used RTP protocol to receive packets.
After receiving packets client store it in buffer and do synchronization and create frames of that packet and at last send that ready video at renderer.

After RTP protocol starts working, RTCP also get active. RTCP simply takes information from client regarding network and also get all information of packet like delivery time, sequence no and etc. And generate reports based on that after every particular time-interval to get better result at renderer side

Process at RTP side:

We are working with 2-threads when RTP protocol starts.
1) One threads for data receiving- This one received data packet and find out its probability of discarding. So, it works at the starting point.
2) Other thread for processing data- In this, video frames are creating. Than mapping audio and video happens. At last transmit that data at the end application.

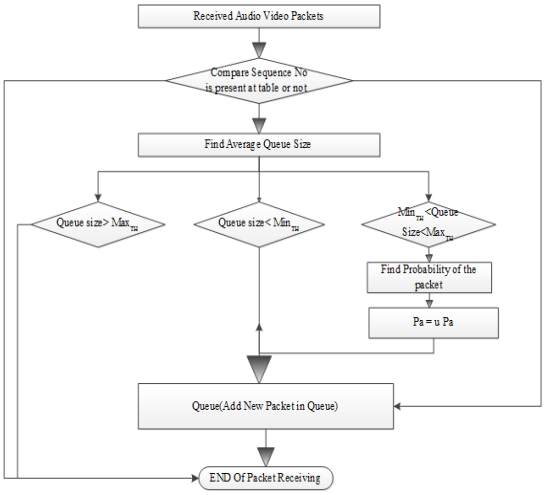

Fig: Queuing at Buffer

Process:

First of all we have to received packets, and find out that we are able to synchronize it or not.

If packet is not late, than find out packet's probability to get in the queue.

$$p_b = \max_p \frac{(k - \min_{th})}{(\max_{th} - \min_{th})};$$

$$p_a = \frac{p_b}{(1 - count * p_b)};$$

Equation: Probability of Packet [6]

If packet is video packet than:

$$p_a = u.\ p_a\ ;$$

Equation: Probability of Video Packet [6]

$$u(\hat{k}) = 1 - \alpha \frac{\hat{k}}{L}$$

Equation: Function of video packets [6]

Than based on their probability put packets in queue or discards it.

And then add packet in the queue-buffer. If other packets are already present in the buffer-queue, when new packet reaches at buffer, that time thread find-out its position where the packet can place.

Now, go to starting point if we still have to receive packet.

Or end the process if no packets remaining.

Further-Process:

When first packet reached at buffer-queue, after that thread two starts processing on it. And converting it in to frame.

And if we are having audio and video both media packets, that time in mapping in between of them done in this thread.

After all, client sends ready frames and audio packets which are already mapped with that frames to renderer to play the video.

Work-Flow:

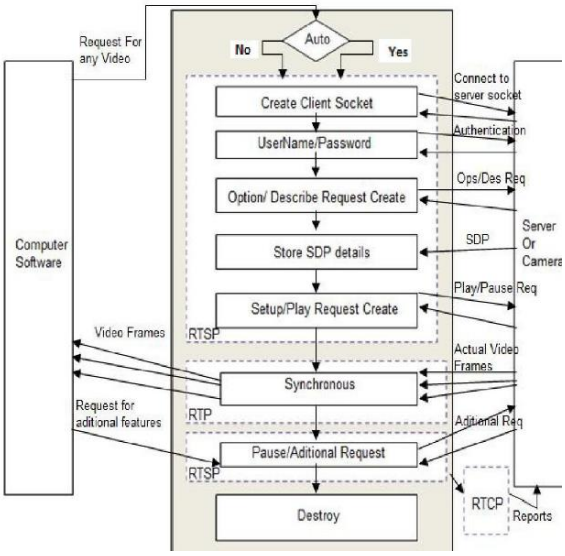

Fig: Work-Flow

Process:

First client receives request from the computer application, to play one video.

And then, client checks the request is for auto connection or not. After that client starts RTSP protocol to establish the connection.

If the request is required auto-connection; that time all the further process's functions execute by itself in one proper sequences. But there is some kind of disadvantage of this feature, that is if we received negative ack, that time we are not able to find out the reason or we can say which request fail.

In other request, when computer application don't want to follow auto-connection, that time we have to follow bellow steps one by one till PLAY response.

Create Client done- In that socket and header list create.

After that Username and Pass sets; and send connection request to server by RTSP protocol.

First OPTION request prepared and send and then received response of it and add required information in the data-base.

Then DISCRIBE request prepare and send receive response and SDP from the server. Then client debug SDP information stores it in database.

Then send SETUP request. Get respond.

At last PLAY request send. Get respond.

After getting positive ack of PLAY req from the server, client informs the system and switch protocol from RTSP to RTP.

Received RTP packets from server. And store that packets in buffer, and process on the packet packets, then convert packets in to frames, at last RTP do mapping in between audio and video packets.

When first packet received, that time RTCP starts, and get information from client .Create report. Send it to server regarding bite rate, jitter, delay and etc. Even RTCP periodically generates the report and send report to server.

During video playing process if user wants to pause the video, that time client simply send pause req by RTSP protocol. And when client wants to play again, that time client simply send play req to server. And display further frames to computer applications

After whole video get play or computer applications want to stop the video, that time client send tear-down request to server.

At last destroy client and in other words memory free at client side done.

Calculate MTU so we can use it for next video.

III. RESULT

We have done our coding in c++ language, in visual studio 2013. And we have tested our project in wireshark tool. They have given us some result in consents of packet loss.

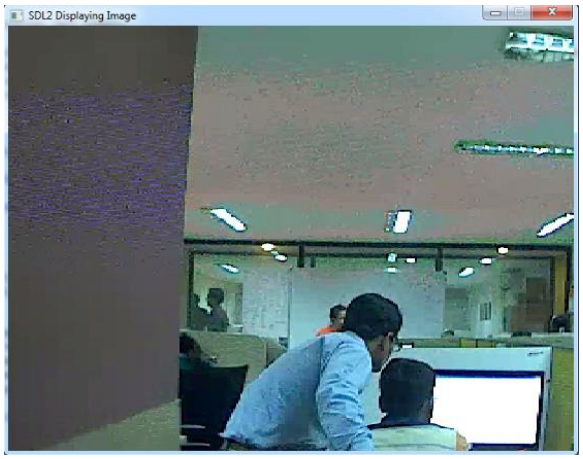

Fig: Result in SDL with pre-synchronization and ViRED queuing technique

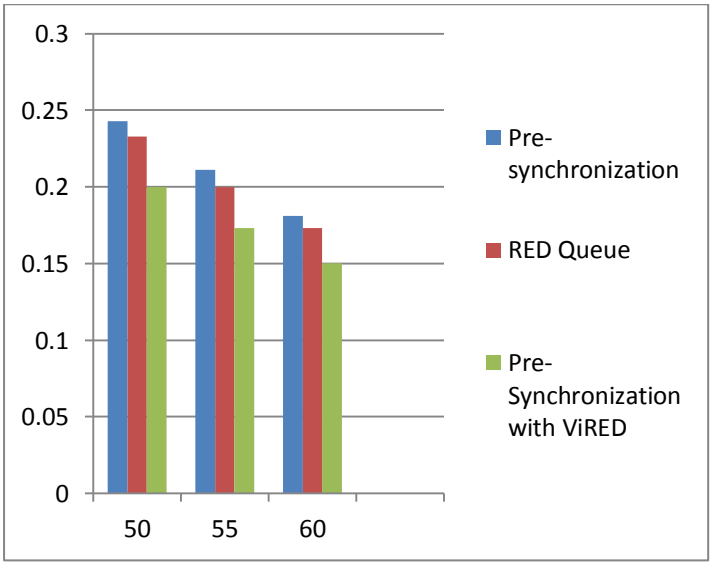

Fig: result graph present relation between bandwidth and packet-loss

IV. CONCLUSION AND FUTURE-WORK

RTSP/RTP/RTCP protocols can work nicely together; By using ViRED queue technique at buffer in pre-synchronization algorithm we are able to get proper and fast synchronization at client side. So, now client doesn't have to wait for system to increase the buffer size, when buffer gets full. So, automatically our latency time has been decreased. We are using two threads at client side one for receiving packets and another for processing and transmitting packets so we are able to decreased the packet loss. ViRED queue technique helps to increased PSNR.

We are planning to validate all available standard cameras and open source severs. So, client can provide VOD features for all of them.

V. REFERENCE

[1] Lei Wang, Weiqiang Wu, Qinyu Zhang, Changjian Zhang; Pre-synchronization of AVS Audio-Video Based on Real-Time Transport Protocol; @2012 IEEE; The national science and technology major projects of china.np.
[2] Iffat Ahmed, Leonardo Badia, and Khalid Hussain; Evaluation of Deficit Round Robin Queue Discipline for Real-time Traffic Management in an RTP/RTCP Environment; 978- 0-7695-4308-6@ 2010 IEEE, DOI 10.1109/EMS.2010.87. pp. 484-488
[3] Tejmani Sinam, Irengbam Tilokchan Singh, Pradeep Lamabam, Ngasham Nandarani Devi, Sukumar Nandi; A Technique for Classification of VoIP Flows in UDP Media Streams using VoIP Signalling Traffic; 978-1-4799-2572-8/14@ 2014 IEEE International Advance Computing Conference(IACC), pp. 354-359
[4] Du Xiao-dan, Hu Qing and Liu Yong-hong,Yu Hong; The System Construction and the Implementation of QOS Control Mechanism in Intelligent Streaming Media; International Conference on Solid State Devices and Materials Science; 1875-3892 @2012 Elsevier B.V. Slection;; doi: 10.1016/j.phpro.2012.03.161. pp. 808-813
[5] DENG Huaqiu ; A Real-time Embedded Video Monitoring System; 978-1-4799-3/24- 0/14@ 2014 IEEE.np.
[6] Tommi Koistinen; Protocol overview: RTP and RTCP; Nokia Telecommunications
[7] Chunlei Liu; Multimedia Over IP: RSVP, RTP, RTCP, RTSP.
[8] Arjan Durresi, Raj Jain; RTP,RTCP and RTP – Internet Protocol for Real-Time Multimedia Communication ; 2005 by CRC Press LLC
[9] Margarita Esponda Streaming Databases for Audio and Video Theory and Praxis Oct, 2007
[10] http://www.rosebt.com/blog/RTP/RTSP/RTCP/Video%Streaming/ (15-11- 2016)
[11] Cao Diep Thang, Nguyen Thuc Hai and Nguyen Linh Giang, "Enhance Quality of Video Transmission Based on Improving the RED Algorithm", VOL.13 No.10, October 2013, 26-32