Performance Evaluation using Adaptive Playout Buffer Algorithm in Voice Communication Over IP (VOIP)

Mansoor Qadir[1], Ghassan Husnain[1], Waqas Ahmad Imtiaz[2], Waqas Ahmad[1]

[1]IQRA National University, Peshawar, Pakistan
[2]Abasyn University, Peshawar, Pakistan

***Abstract*—Real time voice communication is a major challenge in today's internet demand for better Quality of Service. The communication quality is reduced by unnecessary delay and jitter. This paper describes an understanding of threats in Voice communication over IP (VOIP) and how Real Time Protocol (RTP) works in conjunction with Real Time Control Protocol (RTCP). RTCP packet header and delay measurement issues are analyzed with transmission interval and delay of the packets at the receiver end. Further, the end-to-end control of the timestamp in RTP and RTCP is evaluated. An adaptive playout buffering scheme is proposed to improve real-time voice communication over IP networks. Simulation results shows that the tradeoff between buffering delay and late loss can be improved significantly.**

Keywords: Delay, Jitter, RTCP, RTP, VOIP.

I. INTRODUCTION

Voice communication over Internet Protocol (VoIP) for end client perspective is an advanced technology that comes forth as a replacement of the conventional analog telephone allowing making calls using a broadband connectivity. VoIP empowers you to send any kind of information at higher speed of Internet. It is progressively turning into an awesome choice for private and business clients. Numerous suppliers offer boundless plans that will give the endorser boundless long distance, free calling with VOIP. VOIP guarantees less cost over customary telephone lines and along these lines enables you to save money on your monthly telephone costs. VoIP enables you to make voice calls over the web, utilizing your PC, to basically anywhere on internet [1].

Real-time Transport Protocol (RTP) for audio/video data delivery over web. RTP works in combination with the RTCP. It is used to examine Quality of Service (QoS) and transmission statistics of the data. RTP and RTCP utilizes the User Datagram Protocol (UDP). There are various issues that need to be addressed when it comes to performance and quality of service. Due to the varying data flow over the internet and increasing demand of more and more better quality from the customer. Some of the issues addressed in this paper are the discussed mainly on the data loss and delay/jitter. So as to give great quality limiting the delay is vital. Other than jitter the fundamental danger to VoIP QoE is congestion which leads of latency which again translates into jitter and packet loss. Congestion in the network is made up of various traffic types. All these various types of traffic is trying to transmit at the same time regardless of the application protocol's nature [2].

II. BACKGROUND

VOIP transmission utilizes UDP for data delivery as transport layer protocol. VoIP information arrives as the receiver, the application layer translates it before presenting it to the client. SIP and RTP are commonly used application layer protocols for VOIP other than H.323. Real Transport Protocol (RTP) is the Internet protocol which translated real time information, for example, sound and video. Real-time data delivery cannot be ensured by RTP protocol rather it provides mechanism for the sending and accepting applications to support data stream. UDP notifies about the delivery of the data however, it has no mechanism for notification of application regarding any loss in transmission. In time delivery of packets can only be ensured by using lower layer services such as routers and switches. RTP gives functionality suited for conveying real time content like timestamp, control mechanism and reordering of data into correct format before entering the destination is done by the RTP as well. Where there is a problem that RTP doesn't likely cover, packet loss and error recovery [3].

Timestamp shows the current time of an event occurred recorded by a system. The timestamp is used for variety of synchronization techniques such as if there is failure in multi-event exchange it can be canceled. There is another technique that a timestamp is used to record time for initial stage. RTP assigns sequential timestamps to voice packets such that they can be protected using the buffer at receiver, reassembled, and

ISSN 2229-5518
delivered without mistake. The transmitted RTP packet is used to identify failure with the help of time stamp and seq. no. This is proportional to the packet time as well. However, for video it is divided across several RTP packets which means a few packets may have the same timestamp and RTP can identify the absolute time of a specific sample in data stream.

VoIP jitter occurs due the variation in sent and received frames of the data. This variation in packet delay affects the quality of the communication, referred to as the variable delay in the packets transmission. Recipient will see the delays in the communication affected by jitter. Therefore, various service providers now target higher jitter levels as the performance metric.

VoIP packet loss occurs during high traffic influx causing higher packet loss in the network. It for the most part shows itself as dropped discussions or tinny sounds. Packet loss should remain as low as 1%. Mostly service providers guarantee up to 0.5% or lesser packet loss. A packet loss of 1% or 0.25% can be interpreted as error occurring every 50 minutes [3].

A. RTP/RTCP and Delay Measurement

The RTP timestamp in combination with another timestamp called the Network Time Protocol (NTP). Timestamp value can be used to find the absolute time for a particular packet sample. Suppose the RTP time stamp has a value of 65 and NTP timestamp 15:02:00:20 meaning sample 65 in the media stream occurred at time with respect to hr: min: sec: ms.

RTP timestamp can be calculated in various ways. Sampling rate along with frame assembly time, the next timestamp for audio stream can be obtained. Supposedly packets with 20 milliseconds of sample audio at 8 kHz data rate, the timestamp for each block increments by 160, regardless of whether the block is sent due to idle cancelation. In case of video the timestamp frame number is considered, for a 30 f/s video, timestamps would increment by 3,000 per frame, for a 25 f/s video by 3,600. If a frame is sent by many RTP frames, then the timestamp will remain same for all the packets. Moreover, the frame number cannot be calculated if they are sample a-periodically. This normally occur in CODEC applications, the network system clock will calculate timestamp. RTCP packets are generally sent after every 5 sec. The RTCP shows the quality of service at the receiver side. They are used even for point-to-point connections. It can be helpful in maintaining the synchronization of data and also informs if some packets are not received due to silence suppression [4].

The RTP header has the following information;

V=2	P	X	CC	M	Seq # 16
Time stamp 32 bits					
Identification of Synchronization Source (SSRC)					
Identification of Contribution Source (CSRC)					

$\longleftarrow$ ***32 bits header*** $\longrightarrow$

Figure.1 RTP header Format [4]

Succession number field in RTP header in figure. 1 is 16 bits in length, its initial value starts randomly which is incremented by one for next occurring packets. There is another 32 bits field, timestamp, demonstrates when the first RTP is recorded. There is a linear iterative mode of clock which enables synchronization and jitter calculation at the receiver side.

The RTCP has 32 header contains the following information see figure. 2;

V	P	Count	Type	Length
Data				

$\longleftarrow$ ***32 bits header*** $\longrightarrow$

Figure.2 RTCP Header Format [4]

RTCP header fields has padding field in the last byte. The number of reports in the packet can be obtained from the count field. There are sub-fields in the data field. NTP timestamp field is 64 bits. NTP timestamp resolution should not be changed while taking account the estimation precision of a receiver. RTP and NTP have similar values. Sources for inter and intra media synchronization as well as for media independent destination can be ensured by NTP and RTP clock frequency.

Source count field and no. of RTP for the data frames are sent by the sender between the creation and transmission interval. SSRC will be changed once the counter is reset. Source packet byte field, extended highest sequence number receiving field are 4 Bytes long each. There is one more field, Inter-arrival jitter field. This field is calculated by time interval for individual RTP packet. It is an estimation of the timestamp in the form of whole numbers. This can also be referred to as the comparative transmission time between two frames.

The expression for measuring is;

$$J = \frac{J + (|D\,(i-1, i)| - J)}{16} \quad [4]$$

Where 'i' shows first packet, D is the difference between two packets and J is the second packet.

We take a case of timestamp and the difference between the jitters computation. Suppose the first packet is sent at time 15:02:02:00. There is 20miliseconds of time difference between the frames comprising of 160 octets of audio data. Sampling rate is 8 kHz. The clocks needs to be synched between the source and destination. Below are few frame seq. no. mentioned arriving at the destination.

Pck seq. # 1: 15:02:01:20 ms
Pck seq. # 2: 15:02:01:43 ms
Pck seq. # 3: 15:02:01:63 ms
Pck seq. # 4: 15:02:01:84 ms
Pck seq. # 5: 15:02:01:109 ms
Pck seq. # 6: 15:02:01:130 ms
Pck seq. # 7: 15:02:01:150 ms
Pck seq. # 8: 15:02:01:170 ms
Pck seq. # 9: not received
Pck seq. # 10: 15:02:01:230 ms

Here we assume the initial timestamp at the sender and the receiver begins with 0. Timestamp is computed by the formula;

Timestamp = (sampling frequency) x (time-interval between two packet) (1)

Timestamp for 1st packet will be;
Pck # 2: 0.02*8000=160.

Similarly at the receiver side, the timestamp is computed;
Pck # 2: (43ms-20ms)*8000=184
Pck # 3: (0.063-0.02)*8000=344
As for inter-arrival jitter calculation, we can refer to the two formulas mentioned:

$$D(i,j) = (Rj - Sj) - (Ri - Si) \tag{2}$$

$$J(i) = \frac{J(i-1) + (|D\,(i-1,i)| - J(i-1))}{16} \tag{3}$$

The inter-arrival jitter calculated. See Table. 1

S#	D(i-1,i)	S_i	R_i	J(i)
1	0	0	0	0
2	24	160	184	1.5
3	0	320	344	1.406
4	8	480	512	1.818
5	140	640	712	4.204
6	8	800	880	4.442
7	0	960	1040	4.164
8	0	1120	1200	3.904
9	None	1280	lost	none
10	160	1440	1680	13.66

Table. 1 Jitter calculation [4]

B. RTCP Transmission Interval and Report Generation

The RTCP transmission interval is computed on the basis of states of sender and receiver packets [1].
Sender report (SR) are sent continuously by the source. Whereas the receiver reports (RR) are also sent containing all RTP information. In SR there is a type of timestamp, absolute timestamp which gives the information about the time passed in seconds. This timestamp is used by the receiver side RTP. RR is sent for inactive member which are not not transmitting data. The report generated is used for Quality of the service.

V=2	P	RC	PT/SR	Length L
SSRC of sender				

Figure. 3 RTCP sender Report [1]

Figure. 3 shows some of the fields are important to note, for example RC field is a 5-bits field shows the RRs per frame. Moreover, Packet Type (PT) its constant value is 200 which show that it is RTCP's SR packet. More or less other fields are the same as in the RTP frame.
There is not much difference between the SR and RR other than some of the sender frames information is discarded such as information related to the NTP and RTP time stamps. Report generation is useful both at the sender and receiver end. The sender may can improve data transmission depending upon the receiver feedback whereas receivers can identify the state of the problem whether problems are local or outside the domain. Third party also plays a critical part. While managing the networks profile independent monitors may be used that receive frames for RTCP [5].
A field value utilized in both the sender and recipient packets are aggregate counts for various kind of time period difference. Furthermore, take some preventive measures in case of report failure. This last report can be used to analyze the performance of the system however, NTP time is added for data rate calculation from these differences over the interval between two reports. It is possible to compute encoding and performance monitors since the time stamp has no relation with the encoding on the basis of clock rate.
The average payload data rate can be calculated from the sender information when the data is not received the average packet rate over an interval. We can find the average payload size by taking the ratio of the two. Supposing the packet loss to be not dependent upon packet size then the throughput available at the destination can be calculated by number of packets received multiply by the packet size.
The data loss rate between SR and RR can be used for the data frames. The next arriving frame can be calculated with the help of last received seq. no. If the two reports are sequential then the ratio can be calculated by the variance of interval between two frames and frames lost. Whereas the variance in the timestamp of Network Time Protocol and loss in frames will give the fraction of the frame failure rate. Moreover, the variance of the loss frames and next receiving frames will give the total no. of frames that reach at destination. The next receiving frames to reach at destination can also be used to analyze the statistical performance of the failed frames. For example, one out of ten frames has lesser importance as compared to the 300 frame loss in 1500 frames [5].

III. THREAT TO VOIP: JITTER

Subscribers aren't keen in underlying technologies, the intricate details of the customer or convention, or the specialized reasons why their association is performing great or ineffectively; their technical reasons or protocols when they get their telephone. There is a dial tone; and that when they dial, the call gets connected and that the voice is clear and has no recognizable delays. Guaranteeing these things isn't a luxury- it is an unfit need for any specialist service provider seeking after a VoIP deployment. There are two primary network issues threaten to relegate VoIP quality in PSTN range; unmanaged network blockage/congestion and malicious traffic.
There are a numerous factors that affect the voice quality i.e. latency, which can be understood from the fact that when voice data arrives and is in queue for router or any other network element and unable to reach to its destination is direct result of network congestion [6].
Jitter is one reason referred to as delay variation. It is not necessary for the packets to arrive at the destination on pre-defined paths. In such case it arrives at the router with variable time lengths between the sender the destined router, and therefore it is received at the target destination with variable rate. It is imperative that sufficient packets arrive at the destination for the speech data to be recognizable. However, it is not certain that 100% of all packets received must reach the desired destination. In actuality, a voice data stream can have 5% packet loss and still be recognizable. A higher failure-rate will result in an incomplete and indecipherable message.
Packets are dropped when a router is congested and has its queue overflow. It has no place to put extra received packets and therefore discards them. Voice traffic for the most part utilizes the UDP protocol, since it doesn't require extra state communication. UDP packets are sent once, without waiting for affirmation of a guaranteed reception. UDP is utilized as opposed to a conventional retransmission component (i.e. TCP) since these protocols introduce latency since they negotiate the connection with the destination socket.

ISSN 2229-5518

This reliance on UDP can be ambiguous since it provides the lowest possible latency, it means that there is no means to ensure that enough packets are received for an intelligible data message. Thus the congestion that leads to dropped voice packets also results in incomplete VoIP [6].
Jitter is calculated in timestamp units. Suppose an audio stream sampled at 8 kHz, the time at which the frames reach the destination can be calculated by the resident time x 8000.
Delays can be categorized as constant and variables. Components for static delay segments add to direct delay however, egress trunk buffer produce queueing delay. They are controlled through de-jitter buffer at the delivery end which can be a router or a gateway.

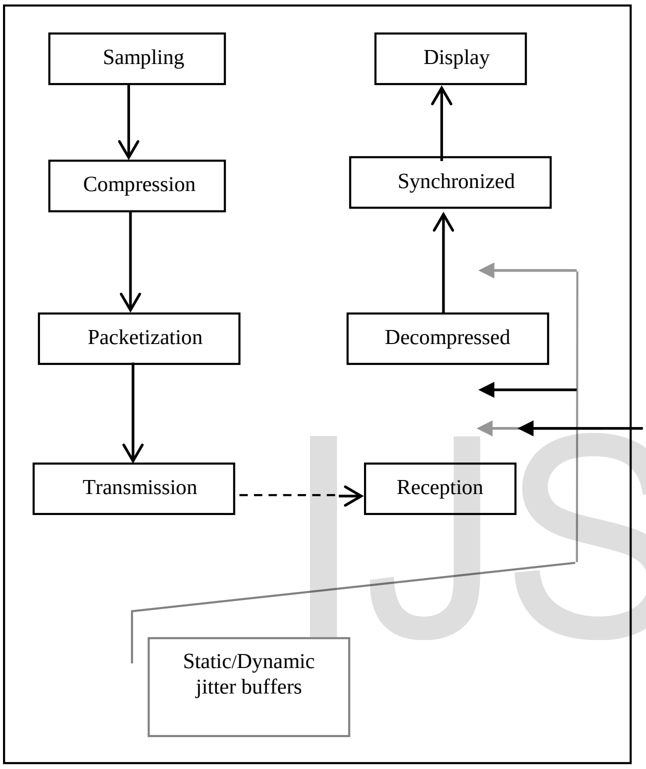

Figure. 4 Jitter Buffer [6]

Before understanding the jitter it's important to know about the different types of delays. Delays that are measured from end-to-end are referred to as one way delays as well**.** It is the measure of elapsed time by data to move across the network. One way delays consists of four components propagation, transport, packetization and jitter buffer delays.
Propagation is the delay that takes time for a data packet to move end-to-end across the network. Propagation delay depends upon the distance. Transport delay is the time between the network devices, including routers, gateway, firewall, WAN etc.
Packetizing is the delay it take time for codec to digitize the analog signal and encodes sends till it is received by the receiver. Jitter buffer is the delay introduced by the receiver to hold one or more datagrams at a time [6].
A jitter buffer is introduced by the receiver as said before, just before the data packets are synchronized and displayed. The purpose of jitter buffer is to remove the transmission delays. When a packet receives this buffer adds a small amount of delay so that the receiver doesn't seem to be receiving data at varying time intervals. There are two types of jitter buffers. A static jitter which is hardware based and configured by the manufacturer. A software-based jitter buffer is called a dynamic jitter buffer and can be configured by the system or network administrator. A static jitter buffer is used before the decompression and a dynamic jitter buffer is used after the decompression of data [6]. See Figure. 4.

A. Static Vs Dynamic Jitter Buffer:

To solve the jitter problem, systems use a dejitter buffer. With regard to the input, voice packet data arrives at the de-jitter buffer 40 in an unsynchronized fashion. That is, every time a packet arrives, it is received by the dejitter buffer and stored therein. Initially dejitter buffer is reset. During initialization, the dejitter buffer is centered to a minimal delay. The nominal delay is equal to the amount of de-jitter that the system can handle. For example, if each voice packet represents 10 ms of voice, and the nominal delay is set at 50 ms, the de-jitter buffer will not send a packet out before there are at least five packets stored in the buffer. With regard to the output, after the de-jitter buffer has been initialized, packets are read from the de-jitter buffer in constant time intervals, such as one packet every 10 ms, wherein exactly every 10 ms, the local receive procedure pulls a packet from the de-jitter buffer. If one or more packets have been delayed in the network, the fact that five 10 ms packets have been stored in the de-jitter buffer provides that the receive procedure can pull packets for the next 50 ms from the de-jitter buffer without degradation of the voice quality. When packets arrive faster from the network that they are pulled from the dejitter buffer, they accumulate in the dejitter buffer, and are not discarded [7].
A special procedure associated with de-jitter buffers allows packets that arrive out of order to be sorted so that these packets will be played through for example a telephone in the correct order. The bigger the de-jitter buffer and the nominal delay, the better the de-jitter buffer can handle delay jitter. However, the bigger the buffer, the more delay the de-jitter buffer introduces to the system. Generally, total delay exceeding 150–250 ms degrades the quality of the conservation over the network. Thus, the characteristics of a de-jitter buffer must be tuned to the characteristics of the network delay. The characteristics of the network delay might change constantly, especially in a packet switching network. Many adaptive algorithms have been suggested to achieve the best nominal delay for a given state of the network. Most of the algorithms which have been formulated have been based on a complex statistical analysis of the characteristics of the network. These techniques suffer from several drawbacks. If the technique is used, it is important to properly analyze the characteristics of the network delay because an error in the adaptive algorithm can result in additional degradation of the voice quality. Analyzing the characteristics of a network delay often requires a lot of complex computations which makes it very expansive in terms of computer power for multi-line systems. In addition, fixed point processors are commonly used in association with voice processing applications, and the computations associated with analyzing the delay characteristics of a network involve floating point calculations. Floating point calculations are difficult to implement in a fixed point processor [7].

ISSN 2229-5518

Apart from the hardware based jitter buffer another jitter is used which is a dynamic jitter buffer or Adaptive jitter buffer which is software based monitoring of the delay in the incoming packets and delaying their play out keeping in mind variable delay [5]. The slower packets are played out in time. This may increase end-to-end delay which will increase the play out time. This increase in playout time can be irritating at time since the buffer is too big or the frames can be discarded for its smaller size. The two contradictory objectives of minimizing buffering time and minimizing late packet loss has given rise to various play out algorithms. Whereas in case of high delay between both ends and in case of unknown delay there is a need of selecting appropriate playout time [4]. One of the scheme can be dynamic playout technique which balances the buffer size.

IV. Methodology and Experimentations

A. Packet Loss Emulation:

In this paper we present a new playout buffer algorithm that significantly improves this trade-off. The simulation has been done in NISTNET 2.1.0 [4]. An illustration of how this software emulates the samples of two human audio voices is observed. The packets are generated over 30 ms. The simulation time is around 60 minutes. The jitter variation in Figure. 5(a) is constant where as in (b) is varying between 0 to 50 ms.

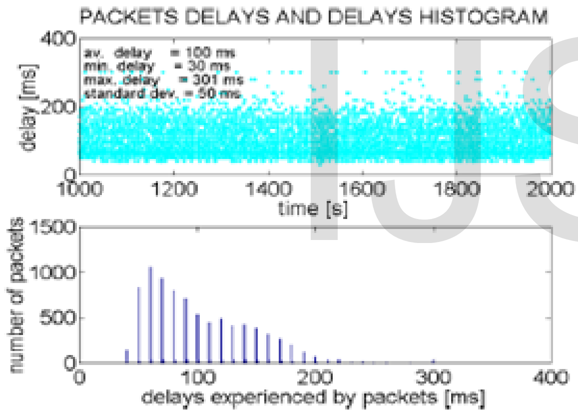

Figure. 5(a) pattern -delay and jitter are constant (delay=100ms,jitter=50ms)

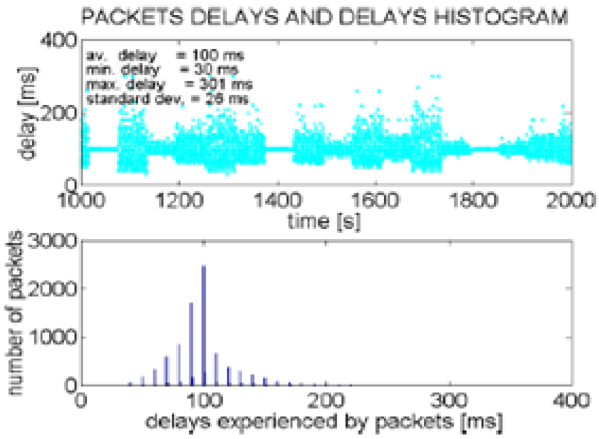

Figure. 5(b) delay=100ms, jitter jumps between 0,10,20,30,40,50 ms every minutte

The two schemes are analyzed after processing the data for the network delays of received and recorded at destination, its ratio is monitored by various parameters.

The data that captured after 60 mins has information about the time stamps, seq. #, and time of arrival, market bits at the receiver end. The delay or loss during one hour is shown in the Figure 6(a), fig. (b) shows the delay/loss trade off.

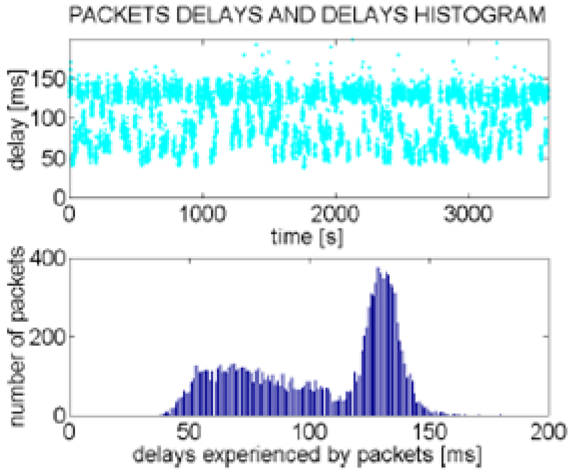

Figure. 6(a) Delays experienced by audio packets

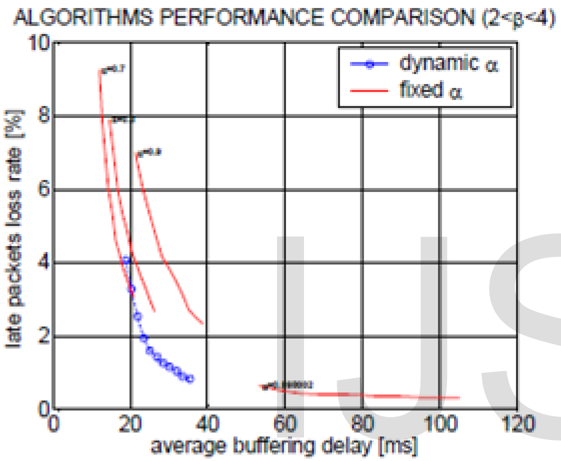

Figure. 6(b) Algorithm performance comparison

In the figure (c) & (d) the playout times are captured and compared for the system for the time period of 500 secs of sending.

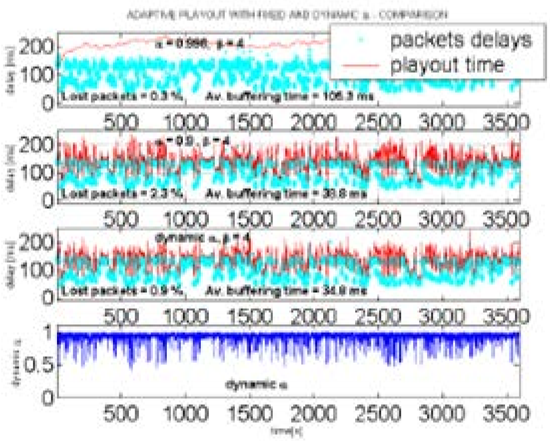

Figure. 6(c) Delay calculation

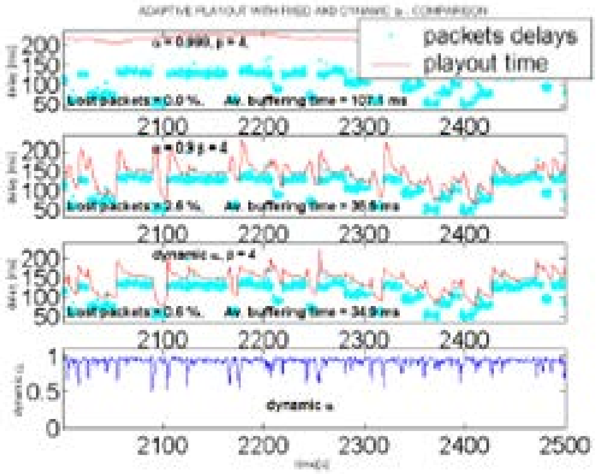

Figure. 6(d) Delay calculation

There are various traffic monitoring i.e. CISCO uses IP SLA for monitoring active traffic in the system. This is a reliable for calculation of network overhead performance.

V. Conclusion & Future work

In this paper, we focus on comparative analysis of RTP and RTCP. Initially VOIP and the threats which limit the performance of VOIP are observed. Experiments were performed to eliminate delay/jitter by comparing two voice sources; audio speech and silence period. Audio packets were generated after every 10ms whereas silence period means no packets. Network delay can be predicted effectively with adaptive playout buffering algorithm. Results shows that our proposed dynamic algorithm can achieve improved quality of service in voice communication over IP.

References

[1] H. Schulrzinne, S. Casner, R. Fedrick: *'RTP A Real Time Transport Protocol'*, July 2003.

[2] Nevil Brownlee, Chris Loosley: *'Fundamentals of Internet Measurement'*, March 2001.

[3] S. Loreto, S. Pietro, and F. Ii, "Real-Time Communications in the Web," 2012.

[4] Miroslaw Narbutt and Liam Murphy: *"VOIP Playout buffer Adjustment using Adaptive Estimation of Network Delays"* 2003.

[5] Fiexi Wang, Wenyu Lei etal. "Adaptive Media Playout Buffer Management For Latency Optimization Of Mobile Live Streaming" July 2017.

[6] S. Loreto, S. Pietro, and F. Ii, "Real-Time Communications in the Web," 2012.

[7] D. Capriglione and L. Ferrigno, "Measurement of the IP Packet Delay Variation for a reliable estimation of the Mean Opinion Score in VoIP services." 2016.