ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 11, November 2017

Design and Implementation of Play-out Buffer Adaptive Windowing algorithm for real-time Voice over IP Communication Systems

Subhajit Mukherjee [1], Shibdas Bhattacharya [2], Abhishek Dey [3], Prasun Kumar Mitra [4]

Lecturer, Department of Electronics & Telecommunication Engineering, Technique Polytechnic Institute, Hooghly, West Bengal, India[1]

Lecturer, Department of Computer Science & Technology, Technique Polytechnic Institute, Hooghly, West Bengal, India[2]

Lecturer, Department of Electronics & Telecommunication Engineering, Technique Polytechnic Institute, Hooghly, West Bengal, India[3]

Lecturer, Department of Computer Science & Technology, Technique Polytechnic Institute, Hooghly, West Bengal, India[4]

ABSTRACT: Maintaining the Quality of Service for real-time communications (VoIP, Video streaming) is a major challenge in IP-based networks which offer best-effort services to its underlying applications. However, irregularities in the packet switched network coupled with different levels of congestion result in jitter which is caused by variation in delay. Recent advancement of technology has introduced the Play-out buffer in such applications to compensate this jitter. The challenging aspect is to optimally configure the size of the play-out buffer, as this is crucial in order to control both queuing delay and packet loss due to the buffer overflow. Considering practical design and implementation of the play-out buffer as a critical problem for deploying VoIP in emerging networks, this paper has developed a test-bed prototype of voice communication system over IP-based networks.

KEYWORDS: play-out buffer; Java Media Framework; M/M/1/B Queuing Model

I. INTRODUCTION

The popularity of internet is increasing as millions of new users sign on day by day. The tremendous use of internet's limited bandwidth often results in congestion which can cause latency in packet transmission. Such latency can result in the increase in packet loss in real-time applications such as Internet telephony (VoIP) or Video Streaming since there is no meaning of late arrival packets for those kind of applications. A major problem suffered by real-time applications over the internet such as Internet telephony (VoIP) is the Quality of voice traffic as compared to the traditional telephone network since data network (or, IP based network) provides best effort services for delivering of data. Maintaining end-to-end delay within a tolerable limit over the best effort packet networks is a great challenge for the real-time VoIP communication system because the best effort packet network doesn't provide any guarantee of timely delivery of packets to its destination. Even these networks are fails to give any assurance about the safely delivery of packets to the destination host that may incur a significant amount of packet loss which is also unacceptable for IP telephony applications in order to provide a certain quality of service. So, to handle the real-time traffic over data networks additional techniques are required to maintain the Quality of Service (QoS) for such applications since real-time applications are delay sensitive, so timely delivery of packets to the destination is a great challenge.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 11, November 2017

II. RELATED WORK

Major QoS metrics that can affect the quality of real-time VoIP application are delay, jitter, and packet loss. Play-out buffer at the receiving end of such a real-time application has a major role in maintaining the QoS by scheduling the play-out time of individual packets to compensate the effect of jitter at the cost of an increase in end-to-end delay. Proper play-out scheduling algorithms are required for this purpose. Several algorithms related to play-out buffer time scheduling for individual voice packets are proposed in this context. The authors in [1] explore the use of aging techniques based on concord to improve the effectiveness of the historical information and hence, the delay predictions. Another work [2] clarified the relationship between mean opinion score (MOS) of played audio and network parameters, such as packet loss, packet transmission delay, and transmission rate, and presented a play-out buffer algorithm considering user's perceived quality of real-time applications. In [3] a dynamic play-out delay adjustment mechanism for video streaming is proposed to compensate for variable network jitters. The tradeoff between end to end delay and conventional speech quality is considered in [4] and according to that, a new jitter buffer algorithm is proposed to provide a better end-to-end user-perceived speech quality. In [5], neural networks and fuzzy systems were utilized to estimate the network delay characteristics for play-out delay adaptation in VoIP networks. An adaptive playback buffer (APB) [6] based on the probing scheme was designed and employed the estimated network situations and the delay and delay jitter margins to calculate the step length which was used to adjust the playback buffer. A new play-out buffering algorithm (eEM Algorithm) proposed in [7] based on ITU-T E-Model to incorporate the effects of loss burstiness on the perceived quality. To evaluate the expected quality, the algorithm considers packet loss correlations and takes into account important effects such as the recency effect, the smoothing of the user perception of sudden variations of the packet loss, and the temporal position of the losses in the speech stream.

The size of the play-out buffer has an important aspect in order to compensate both end-to-end delay and packet discard rate since large buffer size results an increase in play-out delay which will not satisfy the requirements for real-time communications whereas small buffer size results dropping of late arrival packets which will also degrades QoS in real-time applications [16]. A very few amounts of papers are available that are related to the estimation of optimum buffer size and the variation of this optimum buffer size based on the change in network condition. In [8], the authors have designed a jitter buffer algorithm to reduce jitter and compensate its effects and observed the effect of various buffer sizes on delay and jitter. According to the simulation result, the authors conclude that the optimum buffer size for their designed network (with a data transmission rate of 2 Mbps) is 10. But, when network condition will vary abruptly how the optimum value of the play-out buffer size will vary that is yet to be considered in this paper. An adaptive windowing algorithm is introduced in [9] to dynamically adjust the window size when a voice packet is received then; the histogram of delay is established according to the window packets. The work in [10] has addressed the problem of jitter in VoIP which inflicts a heavy loss on the quality of service. In this regard, they proposed an adaptive windowing algorithm which predicts the delay and updates the window in the network. But, in both of the papers [9, 10], simultaneous updating of window size results an increase in complicacy and computational complexity in adaptive windowing algorithm that cause the degradation in system performance. In [11], the authors have proposed a new jitter buffer algorithm to reduce the play out delay and packet discard ratio and subsequently conducted experiments using simulation to study the effect of window size for different network capacities. But they haven't find out the exact values of optimum buffer size for diverse network scenarios.

In order to retain a higher QoS in real-time VoIP application, the proper configuration of the play-out buffer is required since it is responsible for compensating the effect of jitter and at the same time it is also caused by end-to-end delay and packet loss. In this regard, an optimum buffer size is required in order to establish a finest trade-off between queuing delay and packet discard rate. Again, the optimum buffer size is varied according to the variation in network congestion. But, if congestions in network are changing so rapidly then dynamically adjusting of play-out buffer size based on the change in network congestion is not a good solution for resolving the QoS issues in real-time communications since network congestion is estimated in terms of packet loss rate which is a statistical parameter that varies randomly with time and frequent estimation of packet loss rate will increases the complicacy and computational complexity in adaptive windowing algorithm of play-out buffer that leads to the degradation of overall system performance. So, a careful adjustment of optimal play-out buffer size is required for the VoIP application over best effort a packet network which is yet to be considered in the aforementioned adaptive windowing algorithms.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 11, November 2017

In this paper, we have designed a VoIP communication system using Java Media Framework (JMF) that is used for the analysis of queuing delay and packet discard rate over a wide range of buffer size and network loss rate. The analysis of these QoS metrics is contributed for the estimation of optimum buffer size over diverse network congestion scenarios. Further, based on these optimum buffer sizes we have developed our adaptive windowing algorithm that adjust the optimum buffer size based on the estimation of average network loss rate and we have implemented it within our designed test-bed VoIP communication system. Moreover, the performance of the proposed adaptive windowing algorithm has also been compared with the conventional play-out buffer algorithm with fixed window.

The rest of the paper is organized as follows. Section II described about the design of test-bed VoIP communication system using Java Media Framework which allows RTP-RTCP architecture [15] as the transport protocol for data transmission and reception. Section III focuses on a thorough analysis of queuing delay and packet discard probability of the play-out buffer and M/M/1/B queuing model [18] is used in this regard. Section IV contains the evaluation of optimum buffer size over the diverse network congestion scenarios from the analysis results in Section III. Section V contains the details of the proposed adaptive windowing algorithm. Section VI illustrates the proposed algorithm's performance in comparison to the traditional fixed windowing algorithm. The work is concluded in section VII after the extensive performance analysis in section VI.

III. TEST BED MODEL

In order to analyze the QoS metrics over a wide range of buffer size and network loss rate and for implementing of our adaptive windowing algorithm to analyze its performance under diverse network congestion scenarios we have developed a real test-bed prototype of VoIP application and Java Media Framework (JMF) [23] serves as the underlying architecture of this VoIP application. JMF processor model and player model are used to develop the caller and callee application for VoIP communication. The network used for our test-bed setup is shown in Fig.1.

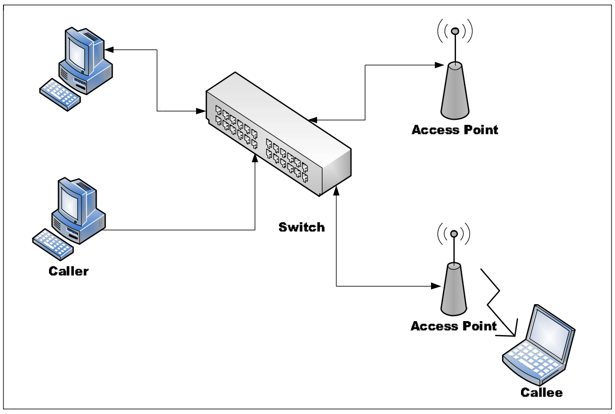

Fig.1. Network Model

As shown in Fig.1, caller and callee application is installed in the sender and receiver host within our test-bed network model. Since today's VoIP communication is extended into mobile phones in order to provide flexibility in terms of mobility and portability support to the end users, therefore we have chosen the wireless link to receive real-time traffic for our designed callee application within the test-bed model. Also, link instability and mobility issues create a great challenge for providing end-to-end QoS support in implementing VoIP application on the wireless platform. The network is shown in Fig.1comprises of both wired LAN and wireless LAN (WLAN). In wired LAN, all packets are routed by the switch to the appropriate destination and in WLAN switch and access points are participated in the routing of real-time VoIP packets to their desired destinations. Details of the caller and callee applications are described in the following section.

The caller application uses the Processor Model of JMF [22, 23] to process the audio data from a microphone using a standard codec. Here we used G.711 codec [25] for our experiments. The caller application captured live audio data

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 11, November 2017

with a specific audio format supported by JMF from a capture device and stored it into an input DataSource which is typically a data handler. The processor will start processing of audio data from the input DataSource as soon as the input DataSource filled with audio data from captured device. The processor outputs audio data into a format specified by the codec which is suitable for transmitting the audio data over the underlying network. Audio data from the output of the processor is stored into output DataSource. The RTP SessionManager, which is responsible for maintaining an RTP Session, is used to form individual RTP packets with a payload as audio data and transmit it to the destination host over the network by creating a SendStream. As soon as a SendStream is created, it begins to send out RTCP sender reports and behaves as a sender host as long as one or more send streams exist. The Processor model and design of caller application are shown in Fig.2 and Fig.3 respectively.

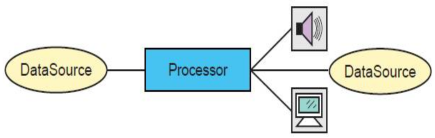

Fig.2. Processor Model of JMF [23]

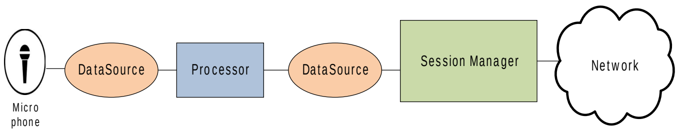

Fig.3. Design of caller application

At the receiving side, the RTP SessionManager of callee application creates a ReceiveStream for each new stream of RTP data to receive RTP packets from the network and store it into a DataSource. The receiver application uses Player Model of JMF [23] to decode the audio data from the encoded codec format and send it to an audio output device such as a speaker to play the audio data. The Player Model and design of callee application are shown in Fig.4 and Fig.5 respectively.

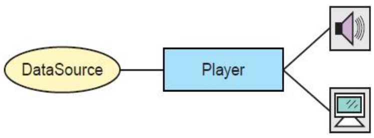

Fig.4. Player Model of JMF [23]

Fig.5. Design of callee application

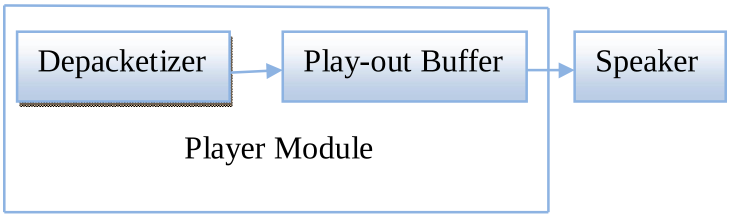

Fig.6. Architecture of player module in callee application

The player module shown in Fig.5 can also be divided into two sub-modules, depacketizer and play-out buffer as depicted in Fig.6. The RTP packets coming from DataSource module are placed into depacketizer sub-module where the voice payloads are extracted from RTP packets and send them to the play-out buffer. Inside the play-out buffer sub-

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 11, November 2017

module, the packets are buffered until their play-out time has reached. The sole responsibility of play-out buffer is to the scheduling of play-out time for individual voice packets so as to minimize the delay variation or commonly termed as jitter between voice packets. When it's time to play out a packet, the packet is placed into the speaker from the play-out buffer, and the callee is able to listen what the caller utters.

IV. ANALYSIS OF QUEUING DELAY AND PACKET DISCARD PROBABILITY

One of the major components of end-to-end delay suffers by a real-time communication system is queuing delay [12]. The size of the play-out buffer has a major role [13] in the context of such a system in order to optimize the trading off between queuing delay and packet discard rate since this QoS metrics that are related to play-out buffer size can affect the quality of a VoIP call. So, analysis of this QoS metrics has a chief importance in order to find out the optimum buffer size for each set of network loss rate and in this regard, our test-bed prototype model of VoIP application is used that is discussed in the preceding section.

For the estimation of queuing delay and packet discard probability of play-out buffer we need a generic queuing model and we have used M/M/1/B queuing model for this purpose in which arrival rate follows Poisson distribution and service rate follows an exponential distribution with a finite buffer size [18]. The reason behind choosing this model for our test-bed analysis is that the M/M/1/B is the simplest Markovian queuing model with only a single server & finite buffer size and the packet arrival rate in IP networks can be modeled as a Poisson process as suggested in [21]. The parameters for M/M/1/B queuing model are as follows.

λ= Mean arrival rate in packets per second (pps).
μ= Mean service rate in packets per second (pps).
$\rho=\lambda/\mu$= Packet utilization factor.
B=buffer size in packets.
n=Number of jobs in the queue.
W=WQ+S= the total time in the queue.
WQ= the time waiting in the queue.
S= the time in the service.
E[n] = E [W]*λ = expectation values of the number of packets in the system (Little's Law [18]).
E [W] = E [WQ] + E [Ws] = expectation value of the total time in the queue.
E [WQ] = average waiting time in the queue or queuing delay.
E [Ws] = average time in the service.
PB = Blocking probability of play-out buffer
The expressions for queuing delay and packet loss probability are,

$$\mathrm{E}\left[W_Q\right]=\frac{\rho*\left(B*\rho^{B+1}-(B+1)*\rho^{B}+1\right)}{\lambda*(\rho-1)*\left(\rho^{B}-1\right)}-\frac{1}{\mu} \qquad \text{eq. (1)}$$

$$P_B=\frac{\rho^{B}*(1-\rho)}{\left(1-\rho^{B+1}\right)} \qquad \text{eq. (2)}$$

We have used Network Emulator Windows Toolkit (NEWT) [24] to emulate the underlying network which is runs at both sender and receiver host. NEWT creates a virtual channel between sender and receiver host and the channel characteristics can be easily emulated from the NEWT application window. An extensive no of experiments are carried out using our test-bed model to collects the values of QoS metrics (queuing delay and packet discard probability of play-out buffer) and the nature of variation of these metrics under a wide range of buffer sizes and network loss rate are shown in Fig.7 and Fig.8 respectively.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 11, November 2017

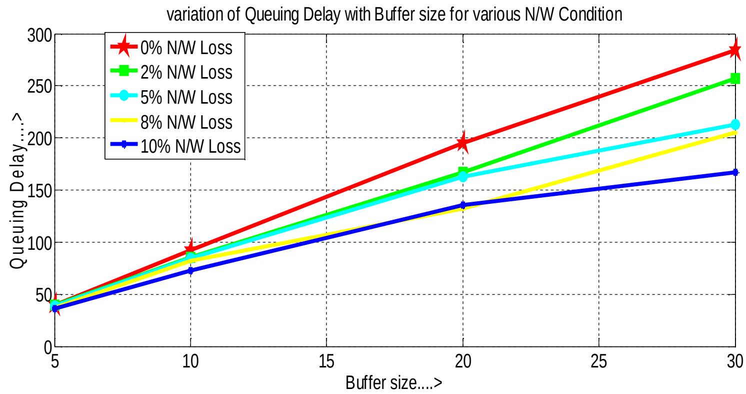

Fig.7. Variation of queuing delay with buffer size for various network condition

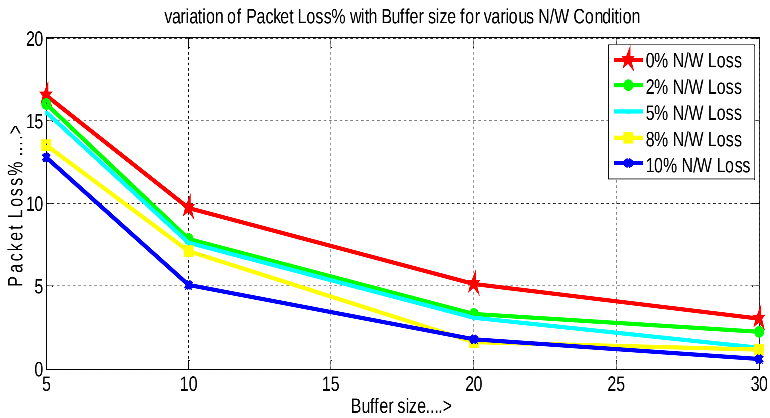

Fig.8. Variation of packet loss probability with buffer size for various network condition

From Fig.7 and Fig.8 we observe the fact that the queuing delay increases almost linearly with buffer size when the network loss is fixed and the rate of increase of queuing delay with buffer size is diminishes with the enhancement of network packet loss rate since increase in network loss rate will decreases the packet arrival rate, so that the service rate gets higher than the arrival rate which incurs a reduction in packet utilization factor that consequence to a decrease in queuing delay; whereas it has been observed from the test-bed readings that there is an exponential reduction of the probability of discarding of late arrival packets due to buffer overflow with respect to queue size. Also, the rate of decrease of packet discard probability of play-out buffer with buffer size takes place with the enhancement of network congestion owing to the reduction of packet arrival rate. So, from the above discussion we can conclude that the size of play-out buffer is most crucial for preserving of both queuing delay and a number of late arrival packets lost due to queue runoff within a tolerable limit in the context of real-time communications.

V. Evaluation of Optimum Buffer Size

The analysis of QoS metrics that are discussed in the preceding section is used to estimate the optimum play-out buffer size. In this section, we have developed an equation of queuing delay and packet discard rate of play-out buffer as a function of buffer size and network loss rate using the concept linear interpolation method in which the trend lines of standard mathematical functions are used to mapped with the curve of queuing delay and packet discard probability that are shown in Fig.7 & Fig.8 respectively. Two equations of queuing delay, DQ, and packet discard probability, PLQ are derived using this process. Thereafter, a cost function is developed as a linear combination of DQ and PLQ whose normalized value is used to generate our objective function which is responsible for the estimation of the optimum play-out buffer size. The equation for queuing delay is expressed by,

$$D_Q = a * B^2 + b * B + c \quad \text{eq. (3)}$$

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 11, November 2017

Where, B= buffer size in packets and a, b, c are constants for a fixed network loss rate but their values are changes according to the variation in network packet loss rate (x in %) as follows.

$$a = 0.004 * x^2 - 0.032 * x + 0.035$$

$$b = 0.007 * x^3 - 0.106 * x^2 + 0.327 * x + 8.774$$

$$c = -0.09 * x^3 + 1.222 * x^2 - 3.74 * x - 1.752$$

Thereafter, the expression for packet loss probability is given by,

$$PL_Q = \alpha * B^{-\beta} \qquad \text{eq. (4)}$$

Where, α, and β are constants similar to a, b, and, c in Eq. (3) whose values are changes according to the variation in network packet loss% (x) as follows.

$$\alpha = 0.021 * x^4 - 0.235 * x^3 + 1.703 * x^2 + 12.42 * x + 71.48$$

$$\beta = 0.0009 * x^3 - 0.013 * x^2 + 0.14 * x + 0.891$$

Now, a cost function, E is derived with a linear combination of DQ and PLQ and is expressed as,

$$E = f\left(\frac{D_Q}{D_{Max}}, \frac{PL_Q}{PL_{Max}}\right) \qquad \text{eq. (5)}$$

Where, DMax and PLMax are the maximum variation of queuing delay and packet loss probability of play-out buffer over the entire range of buffer size and network loss rate. i.e.

$$D_{Max} = \underset{B,x}{Max}(D_Q) \qquad \text{eq. (6)}$$

And,

$$PL_{Max} = \underset{B,x}{Max}(PL_Q) \qquad \text{eq. (7)}$$

Where, $B \in [5,30]$ and $x \in [0,10]$

Our objective function can be evaluated as a linear function of normalized cost function and it's a function of buffer size only. The expression for objective function, fobj is looked like,

$$f_{obj}(B) = \underset{B[5,30]}{f}\left(\frac{E}{E_{Max}}\right) \qquad \text{eq. (8)}$$

Where, EMax is the maximum variation of the cost function over an entire range of buffer size in a certain network condition. The value of the optimum buffer size, Bopt, for each set of network loss rate can be easily estimated from the objective function as follows,

$$B_{opt} = \underset{\substack{B[5,30] \\ x=const}}{Min}(f_{obj}) \qquad \text{eq. (9)}$$

The variation of the objective function, fobj with buffer size for diverse network congestion scenarios is shown in Fig.9 and the list of optimum buffer size for each set of network packet loss rate are Table 1 respectively.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 11, November 2017

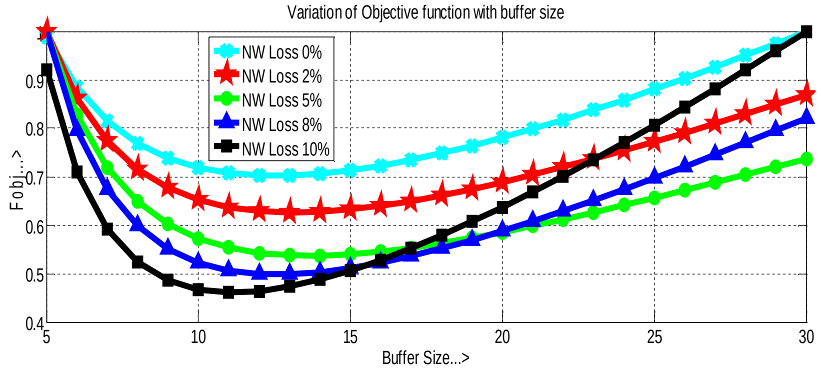

Fig.9. Variation of fobj with buffer size for various network conditions.

Table 1. Optimum buffer size (Bopt) for different network loss%

x (%)	EMax	fobj(B)	Bopt	DQ (ms)	PLQ (%)
0	1.0747	0.6923	12	118.2	7.3
1	1.0413	0.6478	13	113.8	6.2
2	1.0389	0.6071	13	119.2	5.2
3	1.0510	0.5712	13	116.55	4.8
4	1.0688	0.5413	14	115.07	4.4
5	1.0869	0.5174	14	114.8	4.21
6	1.1006	0.4994	14	115.8	3.93
7	1.0045	0.4895	14	118.2	3.6
8	1.0966	0.4797	13	112.8	3.6
9	1.0678	0.4780	12	106.9	3.6
10	1.0302	0.4745	11	99.84	3.6

VI. ADAPTIVE WINDOWING ALGORITHM

Our adaptive windowing algorithm is primarily focused on the adjustment of optimum buffer size based on the average estimated network loss rate instead of the estimation of current network loss rate since network loss rate is solely caused by network congestion which is a statistical phenomenon that varies randomly with time. So, frequent adjustment of optimum buffer size owing to the current network loss rate will degrade the overall system performance due to the increase in computational complexity of the algorithm.

We have designed our adaptive windowing algorithm of the play-out buffer at the receiving side of the test-bed voice communication system. The values of optimum buffer sizes are entered into a look up table whose index no's are represents the network loss rate. When an RTCP Sender Report packet is received by the application, the application extracts the sender's RTP packet count from Sender Report and checks the Reception Statistics to find out the number of packets validated till the NTP time stamp in Sender Report. The difference between these two is the network packet loss. After receiving 10 consecutive Sender Reports, the application calculated the average network loss rate as the mean of the individual packet loss that is estimated from each of sender report and based on that loss rate; it will vary the window size accordingly. We have chosen average network loss rate as the mean of 10 consecutive packet losses from 10 consecutive sender reports, because, for the estimation of average network loss rate with less than 10 consecutive packet losses, the variation of window size will occur more frequently that leads to a degradation of system performance as we observed from test-bed experiments.

The arrival rate is calculated using the inspection about Reception Statistics at one-second interval gap by differentiating the initial no's of packet validated and no's of packet validated after one second when a Sender Report packet is received. After receiving of every 10th Sender Report packet, the mean arrival rate is calculated based on the individual arrival rate calculated from individual sender report. We have chosen mean service rate as 50 packets per

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 11, November 2017

second (pps) since that is the mean processing rate of the codec (G.711) [25] we have used in our application. The Queuing delay and Packet loss probability are evaluated from Eq. (1) and Eq. (2) using the values of mean arrival rate, mean service rate and optimum buffer size for the intended average network loss rate. The algorithm for adaptive window selection of play-out buffer is summarized below which is to be followed as long as the voice communication continues.

Step: 1.
[Initialize] Set I=0, Ar_sum =0, L =0, and Lavg_prev =0
Step: 2.
[Insert the optimum buffer size value into buffer array]
Repeat for i =0 to 10:
Read: Optimum buffer size value, Optbuf
Set Bopt [i] =Optbuf
[End of Loop]
Step: 3.
Wait for RTCP Sender Report Packet to receive
Step: 4.
If RTCP Sender report packet receives, then:
Set I=I+1.
Get sender's packet count, pktsend from Sender Report.
Get No of packets validate, pktreceive from Reception Statistics.
Calculate packet loss% as

$$L = L + \left(\frac{pkt_{send} - pkt_{receive}}{pkt_{send}} \times 100 \right)$$

Wait for 1 second.
Get No of packets validated, pktreceive_new from Reception Statistics.
Calculate arrival rate, Ar = pktreceive_new − pktreceive
Calculate Ar_sum = Ar_sum + Ar
If I=10, then:
Calculate Mean arrival rate,
mAr = Ar_sum ÷ 10
Set Mean Service rate, μ =50
Calculate packet utilization factor,
ρ = mAr / μ
Calculate Queuing Delay using ρ as

$$E[w_Q] = \frac{\rho\left[B \cdot \rho^{B+1} - (B+1) \cdot \rho^{B} + 1\right]}{\lambda \cdot (\rho - 1) \cdot (\rho^{B} - 1)} - \frac{1}{\mu}$$

Calculate Packet Loss% using ρ as

$$L = \frac{\rho^{B}(1-\rho)}{1-\rho^{B+1}} \times 100$$

Calculate average network loss%,
Lavg = L ÷10
If, Lavg ≠ Lavg_prev then:
Set new buffer size,
Bufsize = Bopt [Lavg]
Else:
Lavg_prev = Lavg
[End of If Structure]
Else:
Go to Step: 3.
[End of If Structure]

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 11, November 2017

Else:
Go to Step: 3.
[End of If Structure]

VII. PERFORMANCE ANALYSIS

The performance of our proposed adaptive windowing algorithm is analyzed over diverse network congestion scenarios in comparison to the play-out buffer algorithm with fixed window that is used in the conventional system. The performance metrics used for comparison are the queuing delay, Packet discard probability of play-out buffer, and Mean Opinion Score (MOS) [17]. The queuing delay and Packet discard probability are calculated using Eq. (1) and Eq. (2). MOS gives a subjective measurement of a voice call quality in a scale from 1 to 5 with 5 gives the best performance. The numerical value of MOS can be calculated using E-Model [26]. The basic result in E-Model is the calculation of R-factor. The R-factor is defined by several terms associated with a voice channel across a mixed Switched Circuit Network (SCN) and a Packet Switched Network (PSN). The simplified version of R-factor is obtained from [14] by choosing the default values and is written as,

$$R = 94.2 - I_d - I_{ef} \qquad \text{eq. (10)}$$

Where, Id is the impairment associated with the mouth to ear delay of the path and Ief is an equipment impairment factor associated with the losses in gateway codec's. The expression for Id [14] is given by,

$$I_d = 0.024 \times d + 0.11 \times (d - 177.3) \times H(d - 177.3) \qquad \text{eq. (11)}$$

Where, d =one way delay,

$H(x)$ =Heavyside (or step) Function
= 0, if x<0
= 1, if x>=0

The expression of Ief [14] for G.711 codec is illustrated in the following section.

$$I_{ef} = 30 \times \ln(1 + 0.15 \times loss\%), \text{ if loss \%< 4} \qquad \text{eq. (12)}$$

$$I_{ef} = 19 \times \ln(1 + 0.7 \times loss\%), \text{ if loss \%> =4} \qquad \text{eq. (13)}$$

The R-factor is related to the MOS by the following set of expressions.
For R<0: MOS=1
For R>100: MOS=4.5
For 0<R<100:

$$MOS = 1 + 0.035 \times R + 7 \times 10^{-6} \times R \times (R - 60) \times (100 - R) \qquad \text{eq. (14)}$$

The performance of the adaptive windowing algorithm is measured with respect to fixed window in terms of Mean Opinion Score (MOS) over a wide range of network loss rate is shown in Fig.10 and Fig.11 shows the variation in packet discard probability due to buffer overflow with network loss rate for fixed and adaptive windowing algorithm.

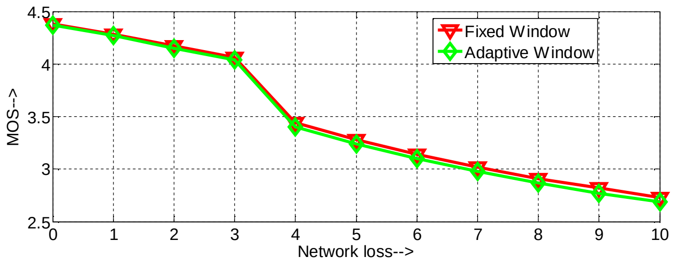

Fig.10. Comparison between variation in MOS with network loss rate for fixed and adaptive window

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 11, November 2017

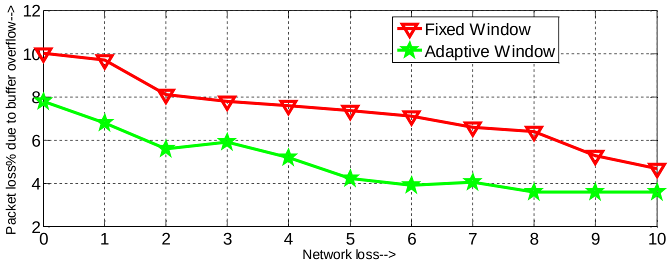

Fig.11. Comparison between variation in packet loss probability of play-out buffer with network loss rate for fixed and adaptive window

From Fig.11, we perceived that the proposed algorithm results in an improved performance in terms of reduction in packet discard probability of play-out buffer in comparison to conventional play-out buffer algorithm with fixed window. So, fewer amounts of packets are discarded due to the buffer overflow in adaptive windowing algorithm than fixed windowing algorithm of the play-out buffer that will cause an enhancement of instantaneous QoS perceived by the end users without incurs any significant enhancement of computational complexity that may caused by the degradation of overall system performance. However, this result in a minor rise in delay for VoIP application but that cannot be accounted to any major degradation of the overall MOS value as we observed from Fig.10. Hence, though the adaptive windowing algorithm caused a minor enhancement of delay in VoIP application, the overall user perception is kept constant as in the case of the fixed windowing algorithm of play-out buffer in the context of real-time VoIP communication.

Therefore, it is inferred from this section that our adaptive windowing algorithm provides better instantaneous QoS in terms of improving the probability of packet loss by keeping the overall acceptability of a VoIP application as perceived subjectively by the end-user i.e. quality of experience (QoE) [19] within a reasonable limit. While QoS is a technology-centered approach, QoE is a user-centered approach to measure the quality of a real-time communication system [20]. So, it is the sole responsibility of a service provider to decide that which one or more technical QoS parameter (delay, loss etc) could be upgraded in order to provide a higher QoS without any hampering of QoE to the end users of a real-time communication system.

VIII. CONCLUSION AND FUTURE WORK

This paper deals with the design and test-bed implementation of a play-out buffer adaptive windowing algorithm with the overall objective of sustaining good quality real-time communication even under congested network conditions. The primary contribution of this paper is to develop an adaptive windowing algorithm for play-out buffer based on varying network dynamics. In the next phase, a generic test-bed model is designed for voice communications over IP-based network. Finally, the credibility of our algorithm is validated through practical application in the test-bed model. Extensive performance analysis is conducted under diverse network congestion scenarios. The tradeoff between queuing delay and packet discard probability is optimized in our adaptive windowing algorithm. Backed by an underlying analytical framework, our algorithm selects the optimum buffer size based on the average network loss rate and accordingly controls both Queuing delay and Packet discard probability of play-out buffer. Test-bed results clearly point to the fact that the adaptive windowing algorithm performs better than play-out buffer algorithm with fixed window of conventional systems with reduced Packet loss, while sustaining the MOS value at acceptable levels.

REFERENCES

1. Cormac J. Sreenan, Jyh-Cheng Chen, Prathima Agrawal, B. Narendran. “Delay Reduction Techniques for Playout Buffering”. IEEE Transactions on Multimedia, Vol. 2, No. 2, June 2000.
2. Kohei Fujimoto, Shingo Ata, Masayuki Murata. “Adaptive Playout Buffer Algorithm for Enhancing Perceived Quality of Streaming Applications”. Telecommunication Systems 25:3,4, 259–271, 2004.
3. Mingfu Li, Tsung-Wei Lin, Shih-Han Cheng. “Arrival process-controlled adaptive media playout with multiple thresholds for video streaming”. Multimedia Systems 18:391–407 (2012).
4. Zizhi Qiao, Ramesh K. Venkatasubramanian, Lingfen Sun, Emmanuel C. Ifeachor. “A New Buffer Algorithm for Speech Quality Improvement in VoIP Systems”. Wireless Pers Commun 45:189–207 (2008).

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 5, Issue 11, November 2017

5. Mohan Krishna Ranganathan, Liam Kilmartin."Neural and Fuzzy Computation Techniques for". IEEE Transactions on Neural Networks, Vol. 16, No. 5, September 2005.
6. Wanqing TU, Weijia JIA. "APB: An Adaptive Playback Buffer Scheme for Wireless Streaming Media". IEICE Trans. Commun. E88-B(10), 4030–4039 (2005).
7. Luigi Atzori, Mirko L. Lobina, Marco Corona. "Playout Buffering of Speech Packets Based on a Quality Maximization Approach". IEEE Transactions on Multimedia, Vol. 8, No. 2, April 2006.
8. Negar Chehregosha, Mohammad Ali Pourmina, "Designing a jitter buffer for QoS improvement in VoIP networks", (IJCSIS) International Journal of Computer Science and Information Security,Vol. 13, No. 4, April 2015.
9. Rongwei Yu, Jie Yuan, Yue Xiao, and Gang Du, "Quality-based Jitter Buffer Algorithm Using Adaptive Variable-Size Window", IEEE-2012 2nd International Conference on Computer Science and Network Technology, Changchun, China.
10. M. V. Sreeraj, Dr. T. Satya Savitri, "Improving the Quality of VoIP Based on Loss Rate and Buffer Delay", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 11, November 2015, ISSN: 2277 128X.
11. Atri Mukhopadhyay, Tamal Chakraborty, Iti Saha Misra, Salil Kumar Sanyal, "Enhancing the QoS of a VoIP call using an adaptive jitter buffer playout algorithm with variable window size", International Journal of Mobile Network Communications & Telematics (IJMNCT) Vol.2, No.3, June 2012.
12. Vandana Gupta·S Dharmaraja, Viswanathan Arunachalam, "Stochastic modeling for delay analysis of a VoIP network", Ann Oper Res (2015) 233, pp:171–180, October 2013.
13. Ren Ping Liu, Gordon J. Sutton, and Iain B. Collings, "A New Queueing Model for QoS Analysis of IEEE 802.11 DCF with Finite Buffer and Load", IEEE Transactions on Wireless Communications, Volo 9, No. 8, August 2010.
14. Cole, R.G. and Rosenbluth, J.H., "Voice over IP Performance Monitoring", ACM SIGCOMM Computer Communication Review, Volume 31 Issue 2, April 2001.
15. H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC: 3550 (July 2003).
16. A. Clark, V. Singh, Q. Wu, "RTP Control Protocol (RTCP) Extended Report (XR) Block for De-Jitter Buffer Metric Reporting", RFC:7005 (September 2013).
17. Ted Wallingford: Switching to VoIP, O'REILLY (June, 2005).
18. Donald Gross, John F. Shortle, James M. Thompson, and Carl M. Harris: Fundamentals of Queuing Theory, Solutions Manual, John Wiley & Sons, Inc (2011).
19. Weiwei Zhang, Yongyu Chang, Yitong Liu, Atif Hameed, and Yuan Tian, "Study on the QoE for VoIP networks", Journal of Networks, Vol. 9, No. 2, February 2014
20. Technical Report on "Human Factors (HF); Quality of Experience requirements for real-time communication services", by ETSI TR 102 643, version 1.0.2, 2000.
21. George Asante, J.B. Hayfron-Acquah, K. Riverson, "Leveraging VoIP on Local Area Network using Java Media Framework", International Journal of Computer Applications (0975 – 8887) Volume 113 – No. 19, March 2015.
22. Java™ Media Framework API Guide, November 19, 1999, JMF 2.0 FCS.
23. NEWT, http://research.microsoft.com.
24. "Pulse Code Modulation (PCM) of Voice Frequencies", ITU-T Recommendation G.711, Fascile III.4, Blue Book, ITU 1988, 1993.
25. The E-model: a computational model for use in transmission planning, ITU-T Recommendation G.107, ITU 03/2005.
26. "One-way transmission time", ITU-T Recommendation G.114, ITU 05/2003.