An Adaptive Jitter Buffer Playout Algorithm for Enhanced VoIP Performance

Atri Mukhopadhyay[1], Tamal Chakraborty[1], Suman Bhunia[2], Iti Saha Misra[2], and Salil Kumar Sanyal[2]

[1] School of Mobile Computing and Communication,
Jadavpur University Salt Lake Campus, Kolkata-700098, India
[2] Dept. of Electronics & Telecommunication Engineering
Jadavpur University, Kolkata-700032, India
{atri.mukherji11,tamalchakraborty29,sumanbhunia}@gmail.com,
iti@etce.jdvu.ac.in, s_sanyal@ieee.org

Abstract. The QoS standard of a VoIP session degrades if its stringent time requirements are not met. Low end-to-end delay of the voice packets and low packet loss must be maintained. Jitter between voice packets must also be within tolerable limits. Jitter hampers voice quality and makes the VoIP call uncomfortable to the user. Very often, buffers are used to store the received packets for a short time before playing them at equal spaced intervals to minimize jitter. However, this introduces the problem of added end-to-end delay and discarded packets. In this paper, some established adaptive jitter buffer playout algorithms have been studied and a new algorithm has been proposed. The network used for the analysis of the algorithms has been simulated using OPNET modeler 14.5.A. The proposed algorithm kept jitter within a tolerable limit along with drastic reduction of delay and loss compared to other algorithms analyzed in this paper.

Keywords: VoIP, QoS parameters, Adaptive Algorithm, Jitter Buffer playout, Congested network.

1 Introduction

The characteristics of the Internet backbone are time-variant in nature. Its properties are so random and unpredictable that it is not an easy task to statistically determine the way the backbone is going to behave in a future point of time. The reason behind the lack of proper prediction of its characteristics is its dependency on the behavior of the other connections throughout the network [1]. The connectivity may be hampered due to several reasons rendering networking applications ineffectual. The networks suffer from congestion when traffics exceeding the capacity of the network are routed through it. As a result, the data packets suffer high delay and loss while passing through the network. Such delay and loss are unacceptable in case of applications having stringent time requirements. This set of applications is called Real-time applications and they include facilities like Internet Protocol (IP) telephony,

D.C. Wyld et al. (Eds.): ACITY 2011, CCIS 198, pp. 219–230, 2011.
© Springer-Verlag Berlin Heidelberg 2011

teleconference, etc. Of the various real-time applications, we have concentrated on Voice over IP (VoIP) since it has gained importance over the past few years owing to its low cost and ease of interfacing between data and voice traffic [2].

VoIP applications are not only sensitive to the extent of the delay and loss suffered by the voice packets but also to the inter-arrival jitter, i.e. the variation in delay suffered by consecutive voice packets [2], [3]. Jitter is one of the main factors that degrade the Quality of Service (QoS) in IP networks [4]. This variation in delay is often the result of network congestion. Generally, the VoIP applications send data at a constant rate. So, any alteration in the end-to-end delay suffered by two voice packets means that the time between two events occurring at the source and the time between the events perceived at the receiver are not equal. Such an event is not desired in a VoIP system as it degrades speech quality. Hence, we need some mechanisms to undo this variation in delay that is being incorporated into the voice packets by the network. One of the feasible solutions is the use of some mechanism which aims to reduce the network congestion, e.g. increasing the packet payload size [5]. Congestion can also occur in the intermediate access points. So VoIP performance can also be enhanced by optimizing the access point parameters [6]. But the most effective solution is to store the voice packets for a short time in the receiver buffer before playing it out, thus reducing the jitter [7]. However, using a fixed playout time for every packet is rendered useless if the network characteristics are variable and the voice packets suffer different extent of delay while passing through it [8]. Several algorithms have already been proposed so that the playout time for a voice packet is delayed in accordance with the variation in the network and thus provide better QoS for the VoIP applications. However, if the playout delay is too large, the end-to-end delay suffered by the voice packets is increased beyond acceptable limits and the VoIP performance may become irritating to the user because significant awkwardness occurs between speakers when delay exceeds 200 ms [8]. On the other hand, if the playout delay is too small, voice packets may be discarded due to late arrival [2]. This hampers the voice quality. It is not desirable that the voice packets suffer from either high delay or high loss. So, it is mandatory to obtain an optimum playout time to get the best performance out of a VoIP system.

In this paper, we have analyzed some of the already established adaptive jitter buffer playout algorithms and have tested for their efficiency in several network scenarios. Further, we have also taken a note of their shortcomings and have proposed a new adaptive jitter buffer playout algorithm that provides the optimum QoS to the VoIP application in terms of delay, loss and jitter. The performance of the new algorithm has been tested in varying network scenarios using OPNET simulator. Moreover, its performance has also been compared with the analyzed algorithms.

2 Related Work

A significant research has already been conducted in the quest of finding a suitable adaptive jitter buffer playout algorithm. As already mentioned, finding the proper playout delay is of utmost importance. In order to find out the efficiency of some of the already existing adaptive jitter buffer algorithms, we have studied five algorithms mentioned in [9] and [10].

2.1 Exponential Average Algorithm (EXP-AVG) [9]

In this algorithm, the delay estimate for the i^{th} packet is computed based on RFC 793 algorithm and the variation in the delays is calculated as suggested by Van Jacobson in the calculation of round-trip-time estimates for the TCP retransmit timer [11]. In this algorithm the estimate of the playout delay for packet i is evaluated by the equation (1).

$$P_i = d_i + 4v_i \tag{1}$$

where,

$$d_i = \alpha d_{i-1} + (1-\alpha)n_i \tag{2}$$

$$v_i = \alpha v_{i-1} + (1-\alpha)\left|d_i - n_i\right| \tag{3}$$

where, n_i denotes the one-way delay of the i^{th} packet and the value of α is 0.998002 [9].

2.2 Fast Exponential Average Algorithm (F-EXP-AVG) [9]

This algorithm is similar to the previous one. The only difference being that if the current packet's network delay 'n_i' is greater than d_{i-1}, then d_i is given by equation (4).

$$d_i = \beta d_{i-1} + (1-\beta)n_i \tag{4}$$

where, the value of β is 0.75 [9].

2.3 Minimum Delay Algorithm (Min-D) [9]

The primary objective of this algorithm is to minimize the delay. So it uses the minimum value of the network delay suffered by the packets in the current talkspurt to estimate the playout delay of the next talkspurt. Let S_i be the set of all packets received in the talkspurt prior to the one initiated by i. So, the delay estimate for packet i is calculated by (5). Apart from this modification, this algorithm is similar to the EXP-AVG algorithm.

$$d_i = \min_{j \in S_i}\{n_i\} \tag{5}$$

2.4 Spike Detection Algorithm (Spike-Det) [9]

One of the most common phenomena observed in a VoIP system is that some of the packets suddenly suffer from high end-to-end delay. As a result no voice packet reaches the receiver for some time followed by the arrival of a large number of voice packet reaching almost simultaneously. We describe this phenomenon as the 'spike'. The above stated algorithms do not take care of this problem. However, this algorithm seeks to overcome the problem with the incorporation of a spike detection mechanism. When, a spike is detected, the algorithm switches to 'SPIKE' mode and later reverts back to 'NORMAL' mode when the network condition becomes normal.

2.6 Window Algorithm [10]

This algorithm collects the network delays of last few received packets and the delay distribution is updated with every incoming talkspurt. The playout delay of the incoming packet is chosen by obtaining a delay that represents a given percentile among the last few received packets. This algorithm also detects spikes. On detection of a spike, the algorithm stops collecting packet delays. If a talkspurt starts during a spike, then the delay of the first packet of the talkspurt is used as the playout delay for that talkspurt. The efficiency of determination of playout delay for this algorithm depends on the window size, i.e. the number packets considered for recording their delay. If the window is too small, then the estimation of playout delay is likely to be poor. On the other hand, if the window size is too large, large memory is wasted for keeping tracks of long and unnecessary history.

3 The Simulation Setup

We have created the congested network scenario used for the analysis of the above mentioned adaptive jitter buffer playout algorithms and to assess our new algorithm with the help of OPNET 14.5.A. The set up consists of four nodes. Two ethernet4_slip8_gtwy_adv gateways are used to interface an IP cloud to the communicating nodes. The IP cloud simulates the presence of an IP backbone in the communication path of the nodes. The gateways and the IP cloud are connected with PPP_adv link whose data rate can be altered.

It is seen from Fig. 1, that one of the nodes acts as the Voice caller whereas another node acts as the Voice callee. These two nodes exchange voice packets between each other. Both these nodes are configured to use G.726 ADPCM coder with 32 kbps and it produces traffic at a constant rate. The other two nodes, i.e. node 1 and node 2 interchange packets unrelated to the VoIP communication, i.e. the cross traffic. Their communication bit rate varies randomly every second between the lower and upper extremes of 0 kbps and 1000 kbps respectively. The basic purpose of these nodes is to congest the links between the gateways and the IP cloud. It is worth mentioning that in order to simulate various network behavior, we have simulated the network several times with the capacity of the PPP_adv link having the values of 600 kbps, 800 kbps, 1000 kbps, 1200 kbps and 1400 kbps. Thus we have created a varying network, so as to induce variable end-to-end delay to the voice packets exchanged between the pair of voice nodes. The IP cloud serves to simulate the routing functionalities and can also increase the delay and packet loss rate. For simplicity, only the results with network capacities 600 kbps, 1000 kbps and 1400 kbps are shown as they cover the three types of jitter conditions, i.e. a network with high jitter (600 kbps network), a network with moderate jitter (1000 kbps network) and a network with low jitter (1400 kbps network).

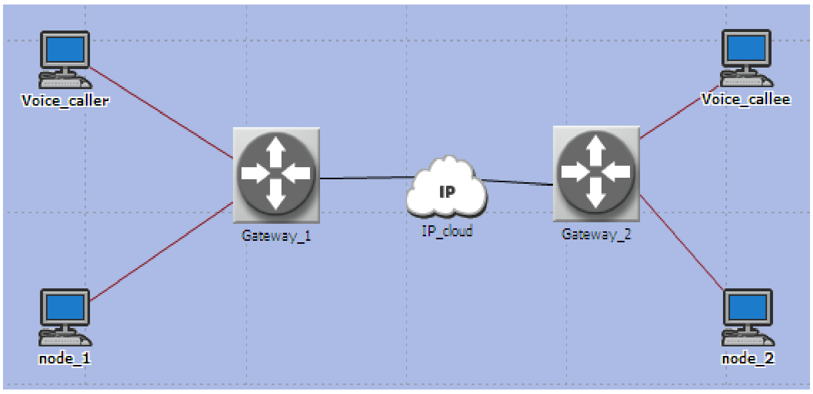


Fig. 1. The OPNET Simulation Setup

4 Analysis of the Existing Adaptive Jitter Buffer Playout Algorithms

The link capacity has been set in accordance with each of the above mentioned values and VoIP call simulations have been carried out between the two voice nodes to study their behavior. The end-to-end delay values for each of the voice packets were noted. Later, these set of readings have been used to implement the various algorithms and then we have compared the results to find out the improvement in VoIP performance, i.e. reduction in jitter.

It is observed from Table 1 that for a network which induces high jitter to the voice packets passing through it (network capacity of 600kbps), the F-EXP-AVG algorithm discards least number of packets, and hence the lowest discard ratio. However, it increases the playout delay to such an extent that the average delay increases beyond a tolerable value and hence the voice quality degrades. The other algorithms induce lower average delay, but discard a large number of voice packets since the packets arrive after the estimated playout time. As a result, the voice call standards go below tolerable limits. The average Mean Opinion Score (MOS) reflects the voice quality offered by each of the algorithms. It is evident that none of the algorithms perform satisfactorily under high jitter conditions.

In a network with moderate congestion (network capacity of 1000 kbps) and consequently moderate jitter, the average delay induced by the algorithms decreases considerably. However, the F-EXP-AVG still imparts higher delay to the voice packets whereas, the Min-D and Spike-Det discards a large number of packets thus suffering from large losses. Further, the performances of the algorithms in a network with low congestion (network capacity 1400 kbps) are also tabulated. Here, we can say that since the inter-arrival jitter for the packets is low, the playout algorithms do not incorporate a significant playout delay to the voice packets. Hence, the end-to-end delay does not increase much. However, we can see that the Spike-Det and Min-D algorithms discard a high percentage of packets and as a result the call quality provided by them gets degraded.

Table 1. Results for the algorithms for network capacity of 600 kbps

Network Capacity	Algorithm	Avg. delay (ms)	Discard ratio (%)	Avg. MOS
600 kbps	EXP-AVG	493.94	6.289	1.0232
	F-EXP-AVG	1446.76	0.482	1.0041
	Min-D	417.80	8.867	1.0191
	Spike-Det	338.17	6.318	1.0461
	Window	341.49	5.524	1.0320
1000 kbps	EXP-AVG	119.83	3.928	1.9114
	F-EXP-AVG	276.46	0.453	2.1900
	Min-D	112.12	6.547	1.6462
	Spike-Det	115.12	6.867	1.6143
	Window	103.63	3.865	1.9386
1400 kbps	EXP-AVG	89.52	0.756	2.8233
	F-EXP-AVG	102.93	0.049	3.4264
	Min-D	88.53	2.681	2.1664
	Spike-Det	87.69	4.728	1.8486
	Window	86.10	0.516	2.9840

5 Proposed Adaptive Jitter Buffer Playout Algorithm

After extensive analysis of some of the existing jitter buffer algorithms, we have come to the conclusion that, when the jitter imparted by the network to the voice packets is very high, the playout delay increases considerably. Moreover, the packet discard ratio also increases beyond tolerable limits. The net result of the above two factors is degradation in the quality of voice in the VoIP session. Our algorithm seeks to reduce the playout delay and packet discard ratio. Our algorithm can be summarized in the following steps which are to be followed as long as the VoIP call continues. We estimate the network characteristics by keeping track of the last '*w*' received packet. We use this accordingly to vary the value of α, where α is a parameter that determines how much a newly received packet depends on the previously received packets. The algorithm is pictorially represented by a flowchart in Fig. 2.

The Proposed Algorithm:

```
1. ni = ith packet network delay, α =0.875;
2. DD = abs(ni - ni-1);  /* DD indicates the absolute
                         value of the difference in
                         network delay of 2 consecutive
                         packets*/
3. IF (i < w)            /* w indicates the number of
                         packets to be considered or the
                         window size */
         find out the inter-quartile range of 'i-1'
   packets;
     ELSE
         find out the inter-quartile range of the last
   'w' packets;
4. IF (inter-quartile range < 5)
           IF (α +0.01< 0.998002)
                α = α +0.01;
           ELSE
                α = 0.998002;
     ELSE
         IF (α - 0.05 > 0.75)
                α = α - 0.05;
         ELSE
                α = 0.75;
5. IF(mode == NORMAL)
         IF (DD > (1 - α) x 100)
                mode = SPIKE;
         ELSE
                goto step 6;
     ELSE IF (ni < α x  ni-1)       /* that is, mode =
   SPIKE */
         mode = NORMAL;
     Else
         goto step 6;
6. IF(mode == SPIKE)
           di = 0.75 x  di-1 + (1-0.75) x  ni;
     ELSE                    /* that is, mode = NORMAL
   */
         IF(new talkspurt)
                di = ni;
         ELSE
                di = α x di-1 + (1 - α) x ni;
7. Y = abs(di - ni) ;
8. vi= 0.998002 x vi-1 + (1-0.998002) x Y;
9. Playout delay = di + 4 vi;
```

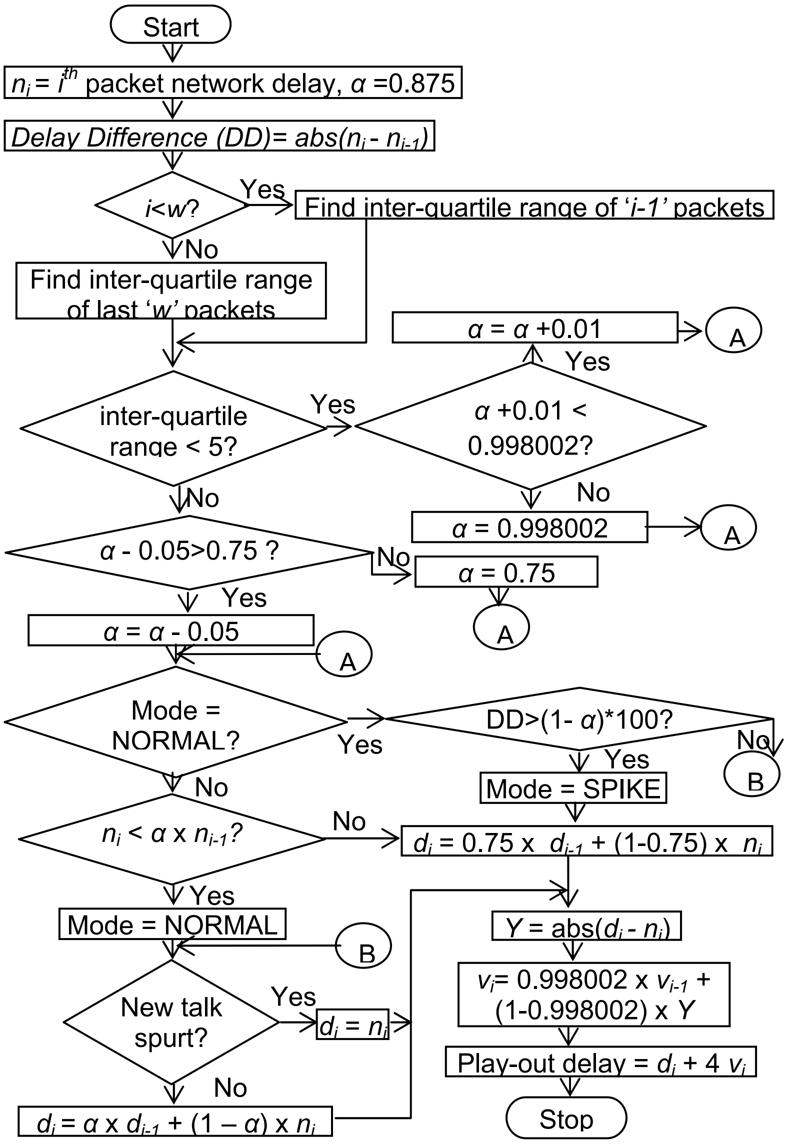


Fig. 2. Flowchart of the Proposed Adaptive Jitter Buffer Playout Algorithm

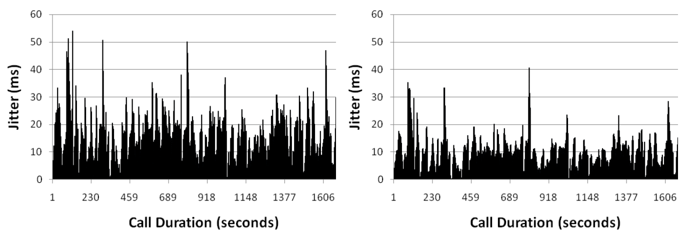
6 Results

The QoS of a VoIP call can be best described by the MOS value as both the end-to-end delays of the packets and the packet loss are considered for the calculation of the MOS [12]. The proposed algorithm is applied to find out its effectiveness and it gives a better MOS than the other discussed algorithms. The results also show that it reduces jitter considerably.

Table 2. Results for the proposed algorithm for different network capacity

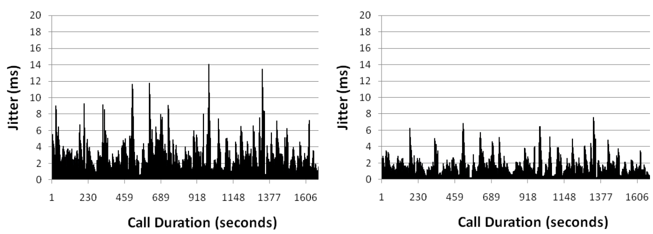
Bandwidth (kbps)	Avg. delay (ms)	Packet Discard Ratio (%)	Average MOS	Improvement in jitter (%)
600	276.93	2.635	1.4508	35.34
800	161.42	2.846	2.0657	38.96
1000	107.80	2.806	2.1059	44.86
1200	92.50	0.664	2.8709	50.19
1400	87.68	0.171	3.2932	54.49

It is seen from Table 2 that the proposed algorithm performs satisfactorily for all of the above network scenarios. In Fig. 3, it is evident that the jitter reduces significantly on applying the proposed adaptive jitter buffer playout algorithm. The average jitter throughout the duration of the call falls from 18.38 ms to 11.88 ms. The results reflect the effectiveness of the algorithm under congested network that impart high jitter to the voice packets passing through it. Fig. 4 and Fig. 5, illustrates the behavior of the algorithm with moderate and low jitter, where the network capacity is 1000kbps and 1400kbps respectively. It is seen that the proposed algorithm reduces jitter considerably and performs well, in all three network scenarios.



(a) (b)

Fig. 3. The Inter-arrival jitter for network capacity of 600 kbps (a) Without playout buffer (b) With proposed algorithm



(a) (b)

Fig. 4. The Inter-arrival jitter for network capacity of 1000 kbps (a) Without playout buffer (b) With proposed algorithm

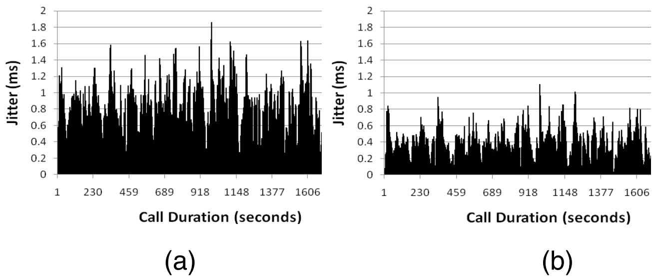


Fig. 5. The Inter-arrival jitter for network capacity of 1400 kbps (a) Without playout buffer (b) With proposed algorithm

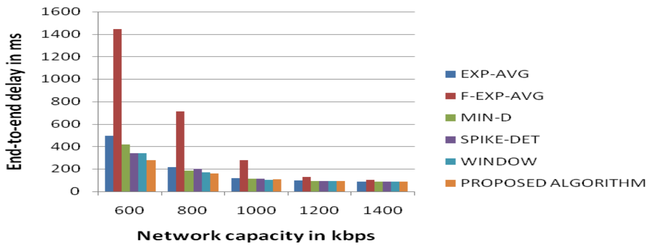


Fig. 6. Comparison of the end-to-end delays of the different algorithms

Further endeavors have been taken in order to find out where the proposed adaptive jitter algorithm stands, when compared with the performances of the other discussed algorithms. Our algorithm has given the lowest end-to-end delay among all the algorithms especially when the network is more congested and the extent of jitter in the voice packets is very high. The end-to-end delay results can be observed in Fig. 6. When examined for packet discard ratio it has been found that our algorithm performs quite well. However, the F-EXP-AVG algorithm has even lower packet discard ratio. The packet discard ratio comparison is illustrated in Fig. 7. Upon further examinations, it is observed that for a congested medium, our algorithm gives the best MOS values. However, for network with very low jitter, F-EXP-AVG gets the edge because of its lower packet discard ratio. The comparative results of MOS values have been included in Fig. 8.

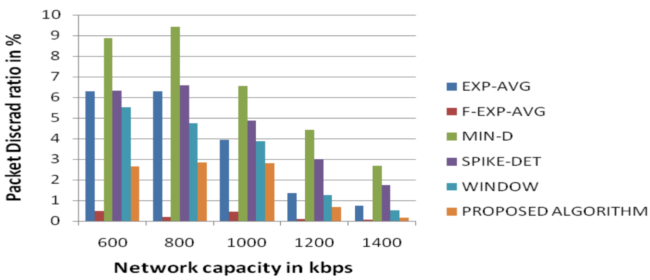


Fig. 7. Comparison of the packet discard ratio of the different algorithms

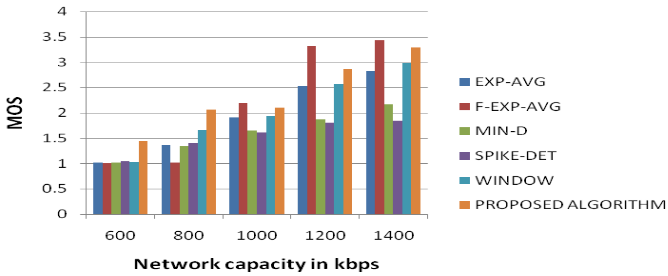


Fig. 8. Comparison of MOS of the different algorithms

7 Conclusion

A congested network transports voice packets with uneven delay. The result of this unevenness is incorporation of jitter in the consecutive voice packets. Jitter is not desirable during a voice call as it leaves the user dissatisfied. Several, algorithms have already been proposed to add a further playout delay to the voice packets in hope of minimizing the jitter. However, selecting the optimum playout delay is a tricky part. These algorithms often under-estimate or over-estimate the network delay of future incoming voice packets, resulting in discarding of the packets or long undesirable end-to-end delay respectively. We have proposed an algorithm that addresses to this problem and properly estimates the network delay of the future incoming voice packets. Our algorithm aims to enhance the QoS of the VoIP session. It seeks to decrease the end-to-end delay and packet discard ratio while allowing a tolerable amount of jitter to be present in the voice packets. The primary aim of our algorithm is to enhance user experience by improving the MOS of the call. We are conducting further studies in order to get even better QoS for the voice calls in a congested network scenario.

Acknowledgements. The authors acknowledge the support from DST, Govt. of India for this work in the form of FIST 2007 Project on "Broadband Wireless Communications" in the Department of ETCE, Jadavpur University.

References

1. Bolot, J.C., Vega-Garcia, A.: Control mechanisms for packet audio in the Internet. In: IEEE Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation, San Francisco, USA, March 24-28, pp. 232–239 (1996)
2. Davidson, J., Peters, J.: A Systematic Approach to understanding the basics of VoIP, Voice over IP Fundamentals. CISCO press (2000)
3. Khasnabish, B.: Implementing Voice over IP, June 12. John Wiley & Sons, Inc., Chichester (2003)
4. Chi, S., Womack, B.F.: QoS-based adaptive playout scheduling based on the packet arrival statistics: Capturing local channel characteristics. In: 2010 IEEE International Workshop Technical Committee on Communications Quality and Reliability (June 2010), doi: 10.1109/CQR.2010.5619942

5. Mukhopadhyay, A., Chakraborty, T., Bhunia, S., Saha Misra, I., Sanyal, S.K.: Study of enhanced VoIP performance under congested wireless network scenarios. In: Third International Conference on Communication Systems and Networks, 2011, pp. 1–7 (January 4-8, 2011)
6. Chakraborty, T., Mukhopadhyay, A., Saha Misra, I., Sanyal, S.K.: Optimization Technique for Configuring IEEE 802.11b Access Point Parameters to improve VoIP Performance. In: Thirteenth International Conference of Computer and Information Technology, Dhaka, Bangladesh, December 23-25 (2010)
7. Gournay, P., Anderson, K.D.: Performance Analysis of a Decoder-Based Time Scaling Algorithm for Variable Jitter Buffering of Speech Over Packet Networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing (May 2006), doi: 10.1109/ICASSP.2006.1659946
8. McNeill, K.M., Liu, M., Rodriguez, J.J.: An Adaptive Jitter Buffer Play-Out Scheme to Improve VoIP Quality in Wireless Networks. In: IEEE Military Conference (October 2006), doi: 10.1109/MILCOM.2006.302119
9. Ramjee, R., Kurose, J., Towsley, D., Schulzrinne, H.: Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks. In: Networking for Global Communications, INFOCOM 1994, June 12-16, vol. 2, pp. 680–688 (1994), doi:10.1109/INFCOM.1994.337672
10. Moon, S.B., Kurose, J., Towsley, D.: Packet audio playout delay adjustment: performance bounds and algorithms. Multimedia Systems 6(1), 17–28 (1998), doi:10.1007/s005300050073
11. Jacobson, V.: Congestion avoidance and control. In: 1988 ACM SIGCOMM Conf., Stanford, pp. 314–329 (August 1988)
12. Cole, R.G., Rosenbluth, J.H.: Voice Over IP Performance Monitoring. ACM SIGCOMM Computer Communication Review 31(2) (April 2001)