Volume 5, Issue 11, November 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Improving the Quality of VoIP Based on Loss Rate and Buffer Delay

[1]M. V. Sreeraj, [2]Dr. T. Satya Savitri
[1]Research Scholar, Department of ECE, JNT University, Anantapur, Andhra Pradesh, India
[2]Associate Professor, Department of ECE, JNT University, Hyderabad, Andhra Pradesh, India

Abstract— In Voice over IP (VoIP) network the playout buffering algorithms which are based on tradeoff between delay and loss can be used to improve the effect of jitter. Since the jitter is one of the main damages which inflict a heavy loss on the quality of service provided in wireless VoIP. To overcome this issue, in this paper improving the Quality of VoIP based on Loss rate and Buffer Delay approach has been proposed. In this approach an adaptive windowing algorithm been implemented where it predicts the delay and update the window in the network. Then a New version E-Model based playout buffering algorithm been proposed where playout buffer size can be adjusted for maximized quality.

Keywords— Buffer Delay, E-Model, Loss rate, Quality of VoIP, Adaptive windowing, Jitter.

I. INTRODUCTION

1.1 VoIP

VoIP (Voice over Internet Protocol) is an increasingly popular service for voice calls over IP networks. This technology enables voice communication through the Internet. It compresses the audio data into data packets and sent it efficiently over the networks this in turn is converted back into the audio data at the receiving end. VoIP sends this audio information in digital form in discrete packets rather than by using the habitual circuit-committed protocols of the Public Switched Telephone Network (PSTN). This technology uses the real-time protocol (RTP) to help ensure that packets get delivered in a timely way.

VoIP includes signaling for establishing and completing each call, as well as digitalizing, coding and packing the voice signal so that it can be transmitted by the data network. While VoIP services are commercially attractive due to their low cost, their success will be influenced by consumer satisfaction, in relation to the quality of the calls, and how closely this quality compares to that of conventional fixed or cellular telephone services [1].

In Voice over IP (VoIP) applications, delay, jitter and packet loss are the main network impairments that affect perceived voice quality. The generally accepted solution to smooth out the effects of jitter is to buffer the received packets before playing them out in their temporal sequence of generation. The playout delay of packets should be adapted in order to maintain the desired balance between delay and late packet loss [2] [3].

Issues of VoIP

- Delay in packet transmission from sender to receiver.
- The variations in packet inter arrival time create difference between when the packet is expected and when it is actually received is jitter.
- The loss of voice packets from sender to receiver [3] [4].

1.2 Need of Quality improvement in VoIP

VoIP is primarily impaired by packet loss, total end-to-end delay and the delay jitter. Packet loss generates gaps in the continuous voice stream, resulting in degraded voice quality. The total packet loss includes network transmission loss which occurs due to congestion and jitter buffer loss by late packet arrival. The total packet end-to-end delay does not cause a reduction in voice quality but it affects the interactive nature of conversations.

In addition, variation in delay (i.e., delay jitter) is extremely difficult to accommodate in conversation. The VoIP jitter buffer is designed to smooth out jitter. However, it affects both the total end-to-end delay time and the total packet loss rate. A small jitter buffer reduces end-to-end delay but is more likely to result in dropped packets that arrive late (i.e., after their schedule playout time). This will, of course, degrade voice quality. Increasing the jitter buffer size reduces the loss rate but increases the overall delay. Finding the tradeoff between these two factors is a key issue for designing the VoIP playout jitter buffer scheduler [5] [9].

1.3 Techniques for quality improvement

The following are some of the techniques which are used to improve the quality in VoIP based on low rate and buffer delay.

- Adaptive Jitter Buffer Play-Out scheme
- MOS-Based Rate Adaption
- Speech Playout Buffering Algorithm
- Adaptive Jitter Buffer
- Adaptive Variable-Size Window

1.4 Problem identification

In our previous paper "SCTP and FEC based Loss Recovery Technique for VoIP" is proposed. In this the data packets which were not affected is chosen by the FEC, and they are transmitted through SCTP. While transmission FEC is executed at every intermediate node to check packet lose. The packet loss estimator gets executed after the data packets reach the receiver endpoint and the retransmission of the lost packets request is given to the sender.

II. LITERATURE REVIEW

Ling fen Sun and Emmanuel C. Ifeachor [3] have presented a new methodology for developing models for nonintrusive prediction of voice quality. They have developed nonlinear regression models to predict perceived voice quality nonintrusive for four modern codec's (i.e., G.729, G.723.1, AMR, and iLBC). The method exploits the intrusive algorithm, PESQ, and a combined PESQ/E-model structure to provide a perceptually accurate prediction of voice quality nonintrusive, which avoids time-consuming subjective tests. Also they further applied the regression models to two main applications: voice quality prediction for real Internet VoIP traces and perceived quality-driven playout buffer optimization. For playout buffer optimization, the proposed perceptual optimized playout buffer algorithm also achieved optimum voice quality when compared to five other buffer algorithms for all the traces considered. However, significant numbers of spikes are accompanied by a gradual increase which cannot be detected by the above algorithm.

Marian VRABEL and Martin KLIMO [4] have presented the method for improving VoIP QoS. Here the algorithm for jitter buffer control is presented. This algorithm inserts and deletes segments of silence and periods of vowels. After the tests, the results are analyzed and compared to the reference jitter buffer. However for higher jitters, the quality of non-adaptive implementations is no longer satisfying.

T. H. Szymanski [5] has proposed a Guaranteed-Rate scheduling algorithm for packet-switched IP routers with rate, delay and jitter guarantees. The algorithm can be used to schedule traffic with 100% throughput in Input-Queued IP routers with unity speedup. The traffic is scheduled according to transmission frames of duration F time-slots. The recursive fair stochastic matrix decomposition is based upon the routing of a permutation through a binary rearrange capable network. The delay and delay jitter experienced along an end-to-end path in a packet-switched IP/MPLS network are therefore small and bounded by an integer number of IIDTs, and the buffer sizes within the IP routers are small and bounded. The proposed algorithm can be used to schedule Guaranteed-Rate traffic and to provide near-optimal queuing delays and essentially-zero delay jitter along end-to-end paths when playback buffers are employed. However a small buffer may cause significant losses, instability or performance degradation at the application layer.

Hyewon Lee et al [6] have designed two novel features for the rate adaptation of VoWLANs. Fast decrease (FD) guides the PHY rates of retransmission frames to low PHY rates so that those frames can be successfully delivered before the retry counter expires. Retry scheduling (RS) manages the timing for the retransmissions in order to avoid both deep channel fading and hidden terminal interference. The proposed novel features improve the QoS of VoWLAN. However there occurs packet loss in the system.

Manjari Chhawchharia and Atanu Guchhait [7] have presented a mechanism for improving VoIP capacity in 802.11b/g networks. Using Virtual Contention-Free Channel Access (VCFCA) the underlying drawbacks of standard-based DCF scheme is overcome. The proposed technique uses traffic aggregation in the downlink with a contention-free channel access mechanism in the uplink transmission. This methodology minimizes the effect of the identified drawbacks of the existing baseline model, thereby resulting in more efficient bandwidth utilization. However in the proposed mechanism the station remains idle for a long period during VCFCA.

Rongwei Yu et al [8] have proposed a new quality-based jitter buffer algorithm. An adaptive windowing algorithm is introduced to dynamically adjust the window size which indicates the numbers of packets used to estimate the future network delay and loss rate. In conclusion, E-Model is applied to evaluate the speech quality based on delay histogram. By searching for the maximum speech quality, the optimal buffer delay is obtained. The whole VoIP communication under our proposed algorithm not only suffers the smallest average delay and lowest packet loss, but also achieves the highest speech quality. However a time-scale modification technique is required.

III. PROBLEM IDENTIFICATION AND SOLUTION

3.1 Overview

In the extension work after the transmission of packets and estimating the packet loss rate, the VoIP quality will be evaluated based on the estimated buffer delay and loss rate.

For predicting the buffer delay adaptive windowing algorithm [8] can be used for updating, which in turn estimates the buffer delay. Here while transmitting the data packets the receiver end uses this variable window size algorithm for updating the window size. By this the future network delay is estimated. The accuracy of the network is improved with this by estimating the buffer delay.

And for estimating the loss rate, NEM (New version E-Model based playout buffering algorithm) [2] can be used. In this algorithm, based on the estimated loss and delay of the previous packets, the playout buffer size can be adjusted for maximizing the estimated conversational quality during the future conversional unit. By estimating loss and delay, conversational quality is determined. This can achieve a finest perceived speech quality, and reduced bursty loss.

3.2 Adaptive Windowing Algorithm

An adaptive windowing algorithm with the help of sliding window whose size can be immediately recomputed online based on rate of change examined from data in the window. This process helps the window to detect and adapt certainly to the distribution change and concept drift. Once the algorithm is finished, it come with reliable conclusion that there is no change inside the updated window. The ADWIN then provides with exact assurance of false positives and false negatives based on only one parameter that is a confidence bound δ, which confidence level of algorithm's output.

The main idea behind this algorithm is very simple and easy. The data which are produced based on distribution function which are divided into two sub windows will have a same observed average. If there is enough difference between two sub windows then it indicates that there is a change in these windows. In other way it can be said that an older fragment of other window will drop only in case if there is enough proof that its average value is distinct from the rest of the window. It simply checks the condition that whether the observed average in both sub windows are varies more than the threshold ε_{cut} which considers the value δ, length of the considered sub windows and so on.

1. Initialize Window U.
2. $U = U \cup \{x_t\};$
3. While ($|v_{uo}-v_{u1}|>= \varepsilon_{cut}$ }
4. Drop elements from tail of U
5. Split of U in to $U = U_0.U_1$;
6. Recount $v_{uo,}\ v_{u1}, \varepsilon_{cut};$
7. End while
8. Output updated window U.

To make mechanism more valuable, a time efficient and memory-efficient version ADWIN 2 is proposed. This version keeps a window of length λ with the help of O (log λ) instead of using O (λ) and checks only O (log λ) cut points. This advancement on time and memory influences its application to the proposed jitter buffer algorithm.

ADWIN act as a change detector to detect the status of current network change. The window size λ which represents number of historical statistical is recomputed online based on rate of delay change detected from the window itself. The window will then grow certainly if the network is stationary and will fall automatically and rapidly to discard old packets thwart is taking place. For each and every packets perform the following steps:

1. If packet p is first packet of a talk spurt, then play out delay of packet p depends on network delay and algorithm goes to step 8.
2. Call Adaptive windowing Algorithm in order to update window size λ
3. Make use of voice packets in window to get the estimated delay distribution based upon histogram method.
4. Calculate the packet loss rate based on delay distribution.
5. Then, calculate speech quality based on E-Model. After performing this, MOS value contains an independent variable i.e. network delay.
6. After that investigate about the optimal playout time with maximum MOS.
7. Set playout delay of this packet to the optimal time found in step 6.
8. End the algorithm

3.3 New version E-Model

This section describes about the proposed playout buffering algorithm called as NEM (New version E-Model based playout buffering algorithm). The main idea behind this proposed algorithm depends on the statistics related to loss and delay of the previous packets, even on the playout buffer size which is set to maximize the expected conversational quality at the time of future conversional unit.

For the proposed algorithm, the basic delay and any other transmission damages are considered with the help of E-Model. This is a type of computational model which make use of various parameters in order to calculate the quality of packet. The result which is obtained by the E-model provides with transmission rating factor R. This transmission rating factor helps to combine all different kind of transmission which are related to the considered connection, for e.g. codec, echo etc. This rating factor consists of following parameter:

$$R = 93.2 - I_d - P_{fr} \qquad (1)$$

Where,

R is the all kinds of transmission impairments, I_d represents the impairments only due to network delay, and P_{fr} is the probability of packet loss (**considered from the previous paper sec 3.4**)

$$I_d = 0.024(d) + 0.11(d - 177.3)H(d - 177.3) \quad (2)$$

Where the H(x) is step function,

$$H(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \qquad (3)$$

P_{fr} is the probability of packet loss which has been estimated in the previous. Through this P_{fr} we find the R in the above equation (1)

$$p_{fr} = 1 - F'_d(\mathrm{d}_i^n + \mathrm{D}'_i) \qquad (4)$$

Where

F'_d is the cumulative function of the retransmission packet delay, which also follows a pareto distribution. D'_i is the estimated local optimal buffer delay which can tolerate retransmission delay and d_i^n is the mean delay.

As it is already defined that the conversational quality is established based on loss and delay, and loss is decided based on delay, the conversational quality represents the function of delay. Hence MOS value is used in terms of end to end delay in order to estimate the conversational quality in the playout buffering algorithm. The advantage of buffer adjustment between talk spurts is that it produces a smoother playout with respect to constantly revising the method. This technique is adopted in the proposed algorithm where the buffer is modified based on maximizing the expected future transmission quality during conversational pauses.

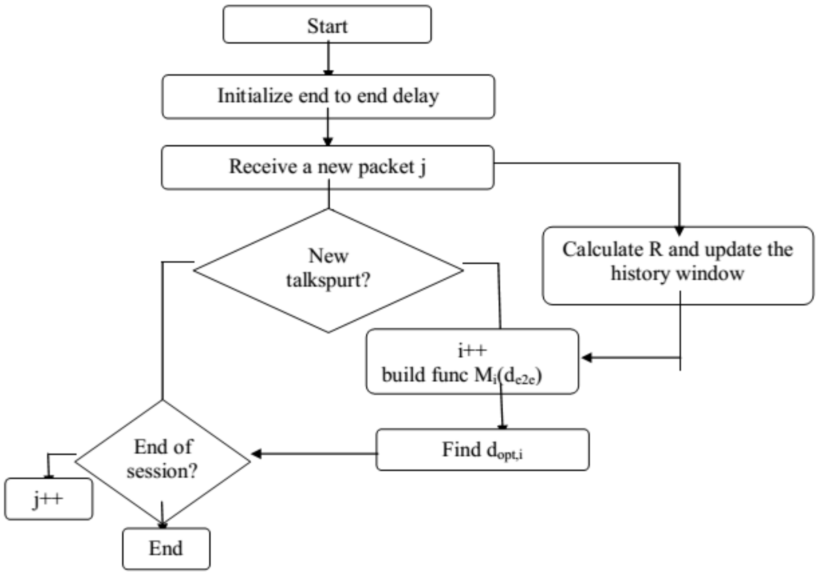

Fig 1: NEM algorithm

In the flowchart consider i = 1 ...) index the talk spurts and (j= 1 ...) index the packets at the time of the VolP session. Let Mi(de2e) denote the E-Model quality function which has been converted to MOS-scale for talkspurt It reflects the observed speech quality in talk spurt i.I.eff .i(de2 and I_d,i(de2e) represents the loss impairment and delay impairment during talkspurt respectively which are mainly used to get the prediction of function Mi(de2e). The proposed NEM strategy is described as follows

(1) First set the end-to-end delay of the first talk spurt.
(2) After that revise the history window after receiving a new packet. However calculate the value of Burst R only at the beginning of a new talkspurt.
(3) At the beginning of a new talkspurt (i.e. talkspurt i), (Burst R) is used, I_e.eff$_i$, $I_{d,i}(d_{2e2})$ which is based on the history window to build the function M_i (d_{e2e}) using equation.
(4) Find the optimal setting of the end-to-end delay $d_{OPt,i}$ for talkspurt which is defined as

$$d_{\mathrm{opt,i}} : M_i(d_{\mathrm{opt,i}}) = \max_{x \in R} M_i(x) \qquad (5)$$

IV. SIMULATION RESULTS

4.1 Simulation Model and Parameters

This section deals with the experimental performance evaluation of our algorithm through simulations. In order to test our technique, NS-2 simulator [10] is used. NS2 is a general-purpose simulation tool that provides discrete event simulation of user defined networks.

We have used the BitTorrent packet-level simulator for P2P networks [ ]. A network topology is only used for the packet-level simulator. Based on the assumption that the bottleneck of the network is at the access links of the users and not at the routers, we use a simplified topology in our simulations. We model the network with the help of access and overlay links. Each peer is connected with an asymmetric link to its access router. All access routers are connected directly to each other modeling only an overlay link. This enables us to simulate different upload and download capacities as well as different end-to-end (e2e) delays between different peers.

In the simulation, 11 nodes are used for 30 seconds of simulation time. The simulated traffic is SCTP. The topology is shown in the following figure.

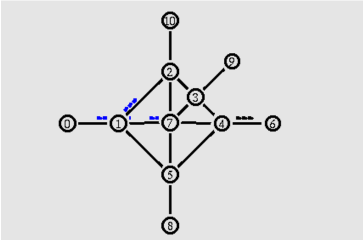

Fig 2 Simulation Topology

The simulation settings and parameters are summarized in table.

No. of Nodes	11
Simulation Time	30 sec
Traffic Type	SCTP
Packet Size	512
Chunk Size	250,500,750,1000Kb
Error Rate	0.1,0.2,0.3,0.4 and 0.5

4.2 Performance Metrics

The proposed SCTPFECLB technique is compared with the standard SCTP technique. The performance is evaluated mainly, according to the following metrics.

- **Packet Delivery Ratio:** It is the ratio between the number of packets received and the number of packets sent.
- **Packet Drop**: It refers the average number of packets dropped during the transmission
- **Throughput**: It is the total number of packets received by the receiver.

A. Based on Chunk Size

In our first experiment we vary the chunk size as 250,500,750 and 1000Kb.

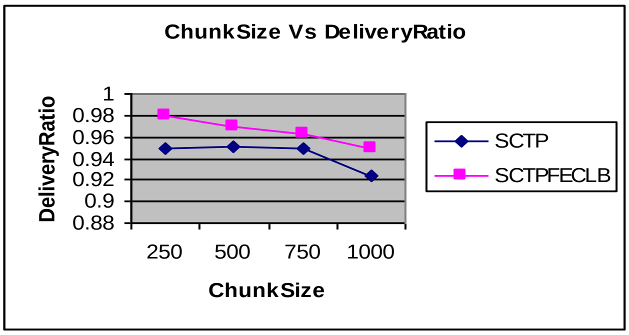

Fig 3: Chunk Size Vs Delivery Ratio

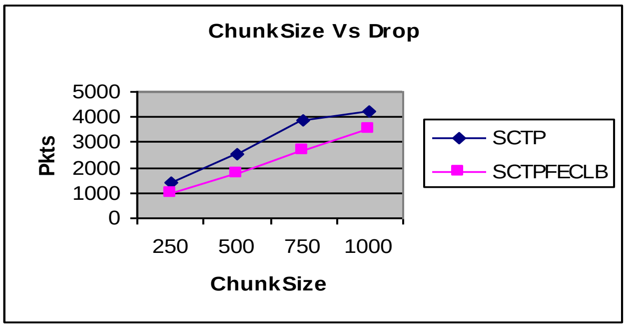

Fig 4: Chunk Size Vs Drop

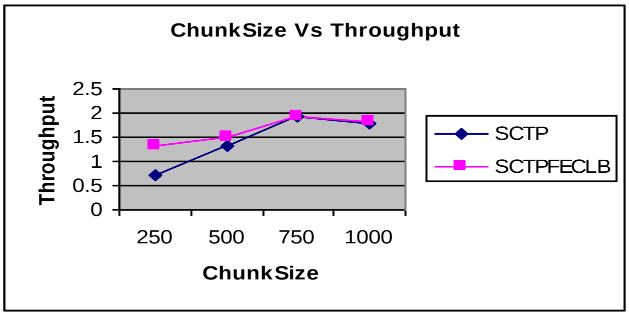

Fig 5: Chunk Size Vs Throughput

Figure 3 shows the delivery ratio of SCTPFECLB and SCTP techniques for different Chunk Size scenario. We can conclude that the delivery ratio of our proposed SCTPFEC approach has 2% of higher than SCTP approach.

Figure 4 shows the drop of SCTPFECLB and SCTP techniques for different Chunk Size scenario. We can conclude that the drop of our proposed SCTPFEC approach has 26% of less than SCTP approach.

Figure 5 shows the throughput of SCTPFECLB and SCTP techniques for different Chunk Size scenario. We can conclude that the throughput of our proposed SCTPFEC approach has 16% of higher than SCTP approach.

B. Based on Error Rate

In our second experiment we vary the error rate as 0.1, 0.2, 0.3, 0.4 and 0.5.

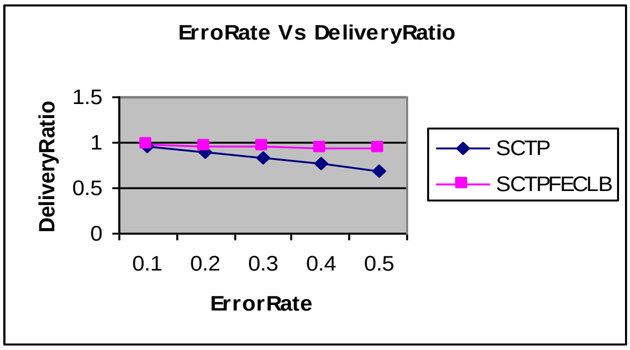

Fig 6: Error Rate Vs Delivery Ratio

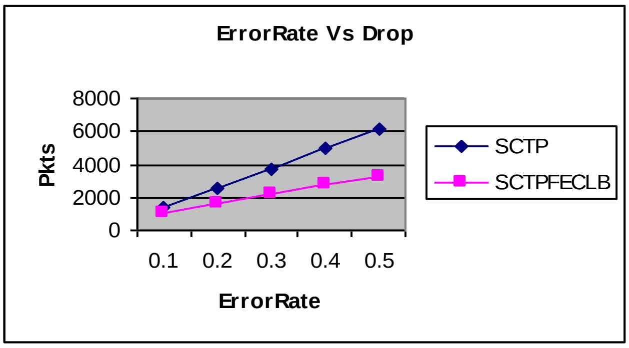

Fig 7: Error Rate Vs Drop

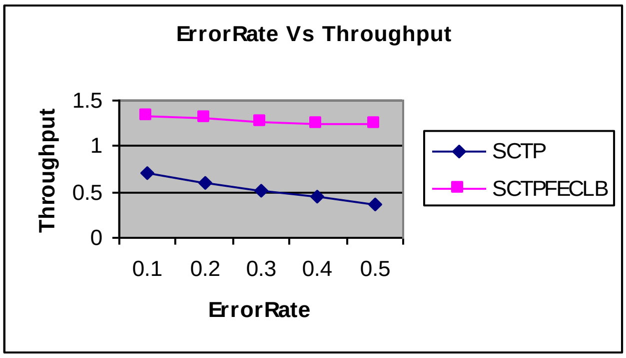

Fig 8: Error Rate Vs Throughput

Figure 6 shows the delivery ratio of SCTPFECLB and SCTP techniques for different Error Rate scenario. We can conclude that the delivery ratio of our proposed SCTPFEC approach has 13% of higher than SCTP approach.

Figure 7 shows the drop of SCTPFECLB and SCTP techniques for different Error Rate scenario. We can conclude that the drop of our proposed SCTPFEC approach has 40% of less than SCTP approach.

Figure 8 shows the throughput of SCTPFECLB and SCTP techniques for different Error Rate scenario. We can conclude that the throughput of our proposed SCTPFEC approach has 59% of higher than SCTP approach.

V. CONCLUSION

In this paper, we have proposed improving the Quality of VoIP based on Loss rate and Buffer Delay. In this approach an adaptive windowing algorithm been implemented where it predicts the delay and update the window in the network. Then a New version E-Model based playout buffering algorithm been proposed where playout buffer size can be adjusted to provide high quality. By calculating the loss and delay, also the conversational quality is determined. This helps to attain a finest identified speech quality, reduce bursty loss.

REFERENCES

[1] N. T. Moura, B. A. Vianna, C. V. N. Albuquerque, V. E. F. Rebello and C. Boeres, "MOS-Based Rate Adaption for VoIP Sources", Communications, 2007 ICC'07, IEEE International Conference on, IEEE 2007.

[2] Zhongbo Li, Shenghui Zhao, Xiang Xie and Jingming Kuang, "An Improved Speech Playout Buffering Algorithm Based on a New Version of E-Model in VoIP", Communications and Networking in China, 2008. China Com 2008, Third International Conference on, IEEE, 2008.

[3] Lingfen Sun and Emmanuel C. Ifeachor, "Voice Quality Prediction Models and Their Application in VoIP Networks", IEEE Transactions on Multimedia, Vol. 8, No. 4, August 2006.

[4] Marian VRABEL and Martin KLIMO, "Adaptive Jitter Buffer Implementation and VoIP QoS", Journal of Information, Control and Management Systems, Vol. 5, (2007).

[5] T. H. Szymanski, "A Low-Jitter Guaranteed-Rate Scheduling Algorithm for Packet-Switched IP Routers", IEEE Transactions on Communications, Vol. 57, No. 11, November 2009.

[6] Hyewon Lee, Seongho Byeon, Byoungjin Kim, Kwang Bok Lee and Sunghyun Choi, "Enhancing Voice over WLAN via Rate Adaptation and Retry Scheduling", Mobile Computing, IEEE Transactions on 2013.

[7] Manjari Chhawchharia and Atanu Guchhait, "High Capacity VoIP Services in 802.11 Networks", Wireless Pervasive Computing, 2007, ISWPC'07. 2nd International Symposium on. IEEE, 2007.

[8] Rongwei Yu, Jie Yuan, Yue Xiao and Gang Du, "Quality-based Jitter Buffer Algorithm Using Adaptive Variable-Size Window", Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference on 2012.

[9] Kevin M. McNeill, Mingkuan Liu and Jeffrey J. Rodriguez, "An Adaptive Jitter Buffer Play-Out Scheme To Improve Voip Quality In Wireless Networks", Military Communications Conference, 2006. MILCOM 2006, IEEE, 2006.

[10] Neork Simulator: http:///www.isi.edu/nsnam/ns

