Fourth International Conference on Networking and Services

Jitter-Buffer Management for VoIP over Wireless LAN in a Limited Resource Device

Mohsen Baratvand, Mahmoud Tabandeh, Arash Behboodi, Ali Fotowat Ahmadi
Department of Electrical Engineering, Sharif University of Technology
Tehran, Iran

***Abstract*—VoIP over WLAN is a promising technology as a powerful replacement for current local wireless telephony systems. Packet timing Jitter is a constant issue in QoS of IEEE802.11 networks and exploiting an optimum jitter handling algorithm is an essential part of any VoIP over WLAN (VoWiFi) devices especially for the low cost devices with limited resources. In this paper two common algorithms using buffer as a method for Jitter handling are analyzed with relation to different traffic patterns. The effect of different buffer sizes on the quality of voice will be assessed for these patterns. Various traffic patterns were generated using OPNET and Quality of output voice was evaluated based on ITU PESQ method. It was shown that an optimum voice quality can be attained using a circular buffer with a size of around twice that of a voice packet.**

***Index Terms*—Circular Buffer, Multi Buffer, Jitter Management, OPNET simulation, PESQ method, Voice over IP, Voice Quality, Quality of Service, Wireless LAN.**

I. Introduction

VOICE over Internet Protocol (VoIP) with rapid growth during recent years has turned into a powerful alternative for classical telephony systems. With an efficient resource management, scalability and reduced cost, VoIP seems to be an inseparable component of next generation networks [1]. Moreover Wireless Local Area Networks (WLANs) are increasingly making their way into residential, commercial, industrial and public areas such as hotels, airports and restaurants. Universities and research centers also benefit from this technology [2]. Developed under the regulation of IEEE802.11 workgroup, WLAN with simple and fast establishment and higher data rates than the traditional wireless local loop systems can achieve a distinguished position which makes them the appropriate choice for local wireless network in medium size companies. However, the security flaws along with handoff problems are still important issues in these networks. Because of the packet based nature of Wireless LANs, VoIP applications are being widely deployed in WLAN devices. A number of greatest companies have done great investments for developing VoIP-WiFi handsets. Development of low cost VoIP-WiFi handsets can become an economically beneficial product which can easily compete with other Wireless Local loop products.

Nevertheless, there is some important constraints in WLAN's which can affect the development of VoWiFi systems. It was justified in [3] that WLAN's can only support between three to twelve users. Moreover, the IP Networks do not support the real time transmission of voice packets due to their variable characteristics [4]. Therefore, it deeply affects the QoS of Voice over IP systems via jitter, delay and packet loss. Based on various analyses, the Voice over IP communications are particularly sensitive to packet loss and delay; 1% packet loss and 100-200 ms delay are considered the maximum acceptable value [3]. Packet loss, Delay and Jitter and their impact on QoS of VoIP have been the subject of several papers [5] [6].

The amplified randomness of Wireless networks which is inherent in its nature asks for elaboration of an efficient algorithm to compensate the flaws of the network. In an internal Wireless LAN, VoIP communications delays and packet loss cannot be a major problem and the concept of jitter is rather an important issue. Besides, low cost VoIP-WiFi handsets normally have limited resources and thus an optimum algorithm for jitter handling should be extracted for them.

In this paper, we suggest using a buffer as an equalizer for jitter and analyze the effects of different buffer types as well as sizes on the quality of voice with various background utilization. End to end delay and jitter are simulated with OPNET software [7]. PESQ scheme is used for QoS measurement [8]. Two major buffer types are assessed: Circular buffer and Multi buffer. The quality of voice will be analyzed with different transmission intervals: $t_{tx} = 10, 20, 30 and 40 ms$. It is shown that, using a circular buffer with size of about twice that of a voice packet, an optimum voice quality will be achieved. Different buffers are simulated with the acquired data from OPNET. The accuracy of the claims will be proved both with output figures and with intuitional illustration. The figures present the Quality of output voice based on PESQ measurement at different buffer sizes for different traffics, Transmission intervals and various background utilizations. Some mathematical analysis will support the intuitional illustration.

In section II, a system model for Voice over IP in Wireless LAN is defined. The method of voice quality measurement is also described. Section III will introduce different buffer types and suggest related algorithms for use. Final conclusion based on various simulations will be explained in section IV.

II. SYSTEM MODEL

A Voice over IP call in WLAN is implemented using OPNET software [7]. The End-to-End delay, delay variation, packet loss and throughput are obtained using a simulation in OPNET environment using WLAN model library. The OPNET simulation contains Two WiFi mobile workstation using IEEE802.11b technology with built-in VoIP ability. Wireless LAN structure is assumed to be

0-7695-3094-X/08 $25.00 © 2008 IEEE
DOI 10.1109/ICNS.2008.17

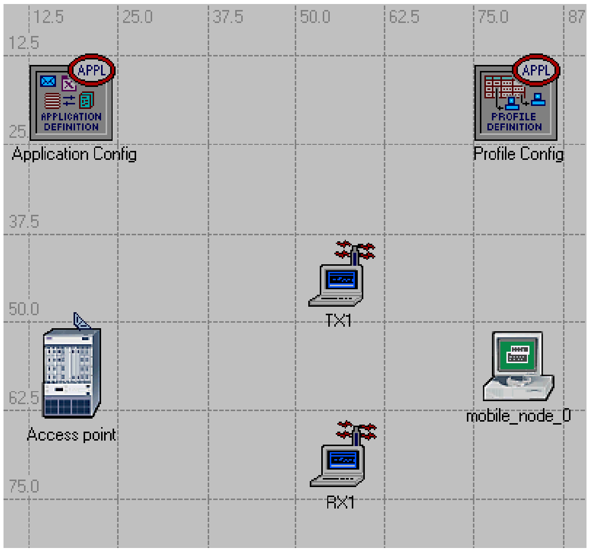

Fig. 1. OPNET simulation model

Infrastructure BSS which means that an access point is needed for the network. A VoIP call is defined between two nodes. The structure of VoIP-WLAN simulation in OPNET can be seen in Figure 1. The voice packets use the RTP protocol over UDP layer as the transmission protocol [9]. The ITU-T G.711 recommendation is assumed as the default voice codec in our VoIP communications so that the transmission rate will be 64 kbps [10]. However there are other types of Voice codecs like G.729 and G.723 which are also suitable for voice conversations specifically for Voice over IP due to lower bandwidth requirement. A lower bit rate can be achieved in these codecs using compression and for the cost of degradation in voice quality. Nevertheless, G.711 is chosen in order to consider the worst case for bandwidth and the best in quality. IEEE802.11b network is adjusted at 11 Mbps with DSSS technology. Background utilization is provided via another wireless workstation which generates background traffic over UDP layer by random wireless transmission with a constant bit rate. Utilization varies from low traffic around 0 Kbps up to 500 Kbps. Simulation is done with various transmission times $t_{tx} = 10, 20, 30 and 40 ms$ along with various background utilization. Each simulation is done for a number of times in order to mitigate the randomness of system. Then, the End-to-End delay and delay variations are calculated using OPNET software for each one. Figure 2 presents four different delay distributions corresponding to different background utilization form a low to high traffic. The adjustment for background utilization is done in a way that traffic is transmitted after the first second of VoIP call. Therefore, the behavior of network is similar in all of the models as it is obvious in Figure 2. It can be seen that delay variation augments together with the background traffic.

PESQ (Perceptual Evaluation of Speech Quality) is a method for determining objective voice quality instead of subjective methods in the telecommunications networks. The method uses a logistic function to map a score output from an objective voice quality method (PESQ algorithm) into a Mean Opinion Score (MOS) which is an estimation of the quality of the voice signal that was transmitted through the wireless network. The logistic function has the form:

$$y = 1 + \frac{4}{1+exp(-1.7244x+5.0187)}$$

Where x is the score from PESQ algorithm. It is usually in the range of -0.5 to 4.5 and y is the mapped MOS score which is in the range of 1 to 5 wherein if y=5, then the quality of voice is considered excellent and if y=1 then the quality of voice is considered unacceptable. We use PESQ algorithm for measurement of voice quality. PESQ combines the best features of two other techniques: the time-alignment technique from PAMS (Perceptual Analysis Measurement System) and the accurate perceptual modeling from PSQM (Perceptual Speech Quality Measurement). PESQ algorithm is applicable not only to voice codecs but also to end-to-end measurement. Defined by ITU-T recommendation P.862 in February 2001, PESQ has become the most widely accepted standard for measuring voice quality over VoIP networks. However, the use of PESQ is not limited to VoIP.

The whole process of simulation consists of three main parts. First, the arrival times of voice packets are extracted from OPNET simulation. Final output voice is provided regarding the results of each simulation and the appropriate algorithm. Finally quality of output voice is assessed using PESQ method.

One can offer a big buffer for mitigation of jitter effect but the basic assumption for our VoIP-WiFi station is the requirement of having limited resources and low cost. This is because of being able to be commercialized in a large scale and having competitive value against the similar solution of Wireless Local Loop. It is assumed that the receiving part consists of a low cost main processor such as a simple microcontroller which would logically be limited in resources. The microcontroller allocates its internal RAM to both voice buffer and program stack to avoid using an external RAM and additional cost. However it should be mentioned that along with the rapid development of technology, low cost processors with high processing power and high resources are being developed but our assumption can provide an optimum resource management algorithm even for the powerful low cost processors. Nonetheless, these assumptions are only for justifying the use of low size buffer and have no effect on our simulation process.

III. JITTER MANAGEMENT

In wireless networks, packet loss and jitter are the most important parameters. In other words, the behavior of packet loss and the variation of network delay, rather than the maximum delay allowed in the system, determine the quality of reconstructed voice at the receiver. By improving buffer management strategies we could have much better voice quality especially in limited resource devices. Jitter is generally defined as the variation in End-to-End de-

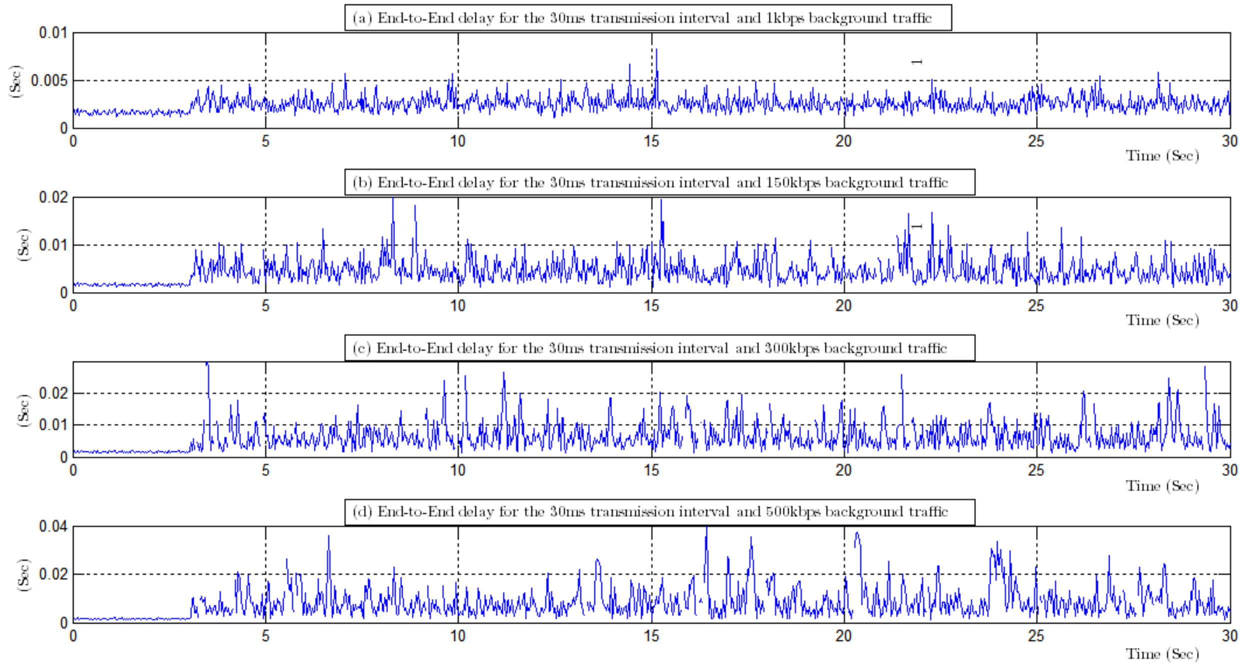

Fig. 2. End-to-End delay for the 30 ms transmission interval in two VoIP-WiFi handsets call and different background traffics: (a)1 kbps; (b)150 kbps; (c)300 kbps; (d)500 kbps

lays.

There are two main points which should be considered in the implementation of jitter-buffer algorithm and can affect the voice quality. First; once the voice buffer is full, it remains full for a certain period of time during which many packets may be lost and lead to consecutive clippings in voice. The second point is opposite of the first one. When buffer is empty, which means the current packet has been possibly lost or it has a larger delay rather than usual, the receiver does not play anything and it can cause disorder for the listener. Instead we can use some corrective methods such as making silence, using voice prediction or some other techniques in order to achieve a better QoS. The packet loss rate during this period changes slowly and has large fluctuations.

Two principle algorithms are used for jitter management both of which exploit a buffer to mitigate the jitter effect. Circular buffer and Multi-buffer are two main buffer types that will be assessed in this paper.

A. *Multi buffer*

Multi buffer uses K independent buffers with size S_D byte so the total size of the buffer is $K * S_D$. While the receiver receives a packet, it stores the packet on the empty buffer and starts to send the frames to the G.711 codec. As soon as the current buffer is full the next buffer will be used for accumulation. It is obvious that because of discrete nature of Multi buffer, the buffer cannot be read smoothly and causes the diminution of voice quality.

It can be seen that the optimum S_D is equal to size of packet payload because each buffer is assigned to one packet or to integer multiples of packet. However the later only adds some additional complexity which is unnecessary due to usage of multiple independent buffers. Therefore it can be assumed that a buffer with size of integer product of packet payload can be treated just like a multi-buffer with S_D equal to size of packet payload.

Figure 3 shows a flowchart of both of algorithms. For the Multi buffer management, after allocating buffer, we should initialize pointers. We use a pointer called $Buff\_index$ in order to switch between K buffers. A read pointer is used to read from buffer. When the first packet arrived, we must enable the output voice. Then one byte should be sent to codec every $125\mu s$ to have 64 kbps transmission rate. Whenever a new packet is arrived, we should check if the adequate space is available or not for the incoming packet. This can be done by this equation:

$$Buff\_index * P\_Len < read\_ptr < (Buff\_index + 1) * P\_Len$$

This equation assures us whether the incoming packet can be stored in the buffer or not. If we are allowed to store the incoming packet, therefore, the buffer index increases by one and if buffer index reaches to the end of buffer, it should be replaced by one.

In order to read from buffer, at first we must check if the buffer is empty. This is accomplished by checking this equation:

$$read\_ptr \neq Buff\_index * P\_Len \qquad (1)$$

Therefore, pointers could not pass over each other to make noise or to degrade voice quality in our system.

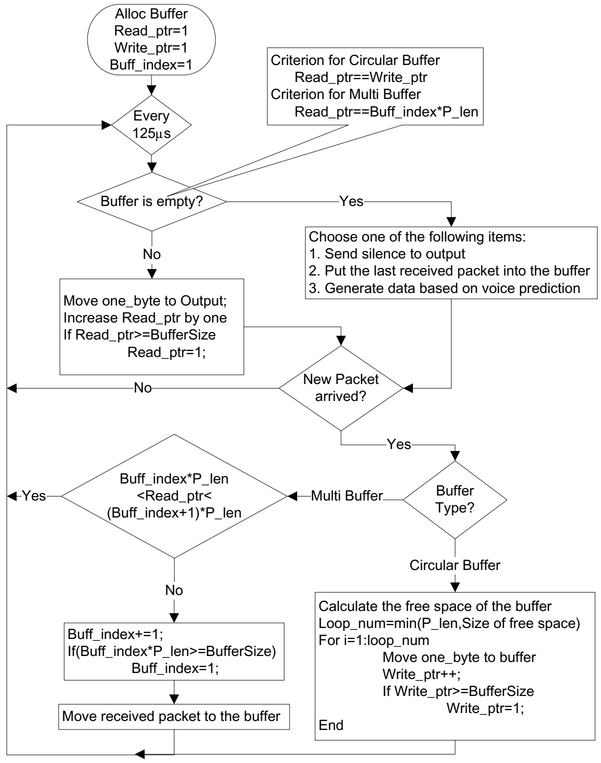

Fig. 3. Flowchart of Multi Buffer Management

B. Circular buffer

There is another popular algorithm which uses Circular buffer, a single continuous buffer with size S_C. It uses two pointers for management of reading from buffer and writing to it and thereby prevents unwanted rewriting on previously saved packets. Each time a packet is received it will be stored on the buffer and write pointer will be increased with the size of received packet. Read pointer is also increased anytime a frame is sent to G.711 codec. This method obviously is more continuous than multi-buffer so the voice can be smoothly played. Moreover it uses the resources more efficiently and can handle different payload sizes without any modification of S_C. Because the buffer should have adequate spaces for receiving a new packet while playing the previous one, the optimum S_C should be the two times of packet payload size.

The Complete flowchart of the algorithm for circular buffer is presented in figure 3. The flowchart can be directly implemented in any embedded system. At first circular buffer should be initialized. Each time a packet receives it will be stored in the buffer and write pointer will be increased with the minimum of received packet size and the remaining free space length. This is done via a variable called Loop_num which is equal to the minimum of the size of free space and packet length. Read pointer is also increased anytime a frame is sent to G.711 codec.

The major difference between the Circular buffer and Multi-buffer is that in the former the space of RAM is used more efficiently. If the free space in buffer is smaller than the incoming packet length, only a small part of the whole packet will be lost while in multi buffer we lose the whole packet entirely even if the free buffer size would be $P\_Len - 1$.

The final point to be considered is that the read and write pointer should not pass over each other. Hence, while the buffer is empty, we should generate silence or exploiting prediction to mitigate the degradation in quality of voice.

IV. Conclusion

Various simulations have been done based on previous algorithms and following results have been obtained. In each simulation, quality of voice has been assessed based on different background utilization and for specific transmission interval. The length of buffer in each simulation varies over the multiples of packet payload. The relation between packet payload and transmission interval for G.711 is as follows:

$$P\_Len = 8 * \Delta t \qquad (2)$$

And we will have the following table:

10ms	80 Bytes
20ms	160 Bytes
30ms	240 Bytes
40ms	320 Bytes

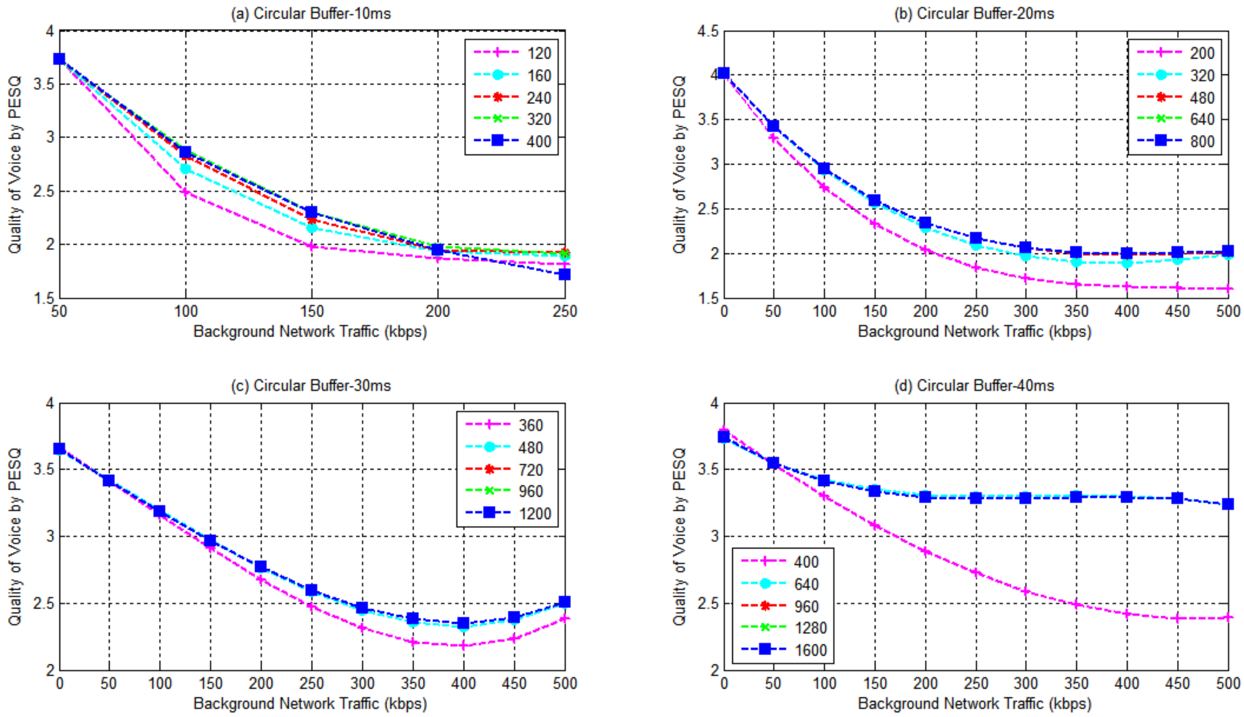

Fig. 4. Voice Quality vs. Traffic for different buffer sizes and different transmission intervals: (a)10ms, (b)20ms, (c)30ms, (d)40ms - Circular Buffer Management

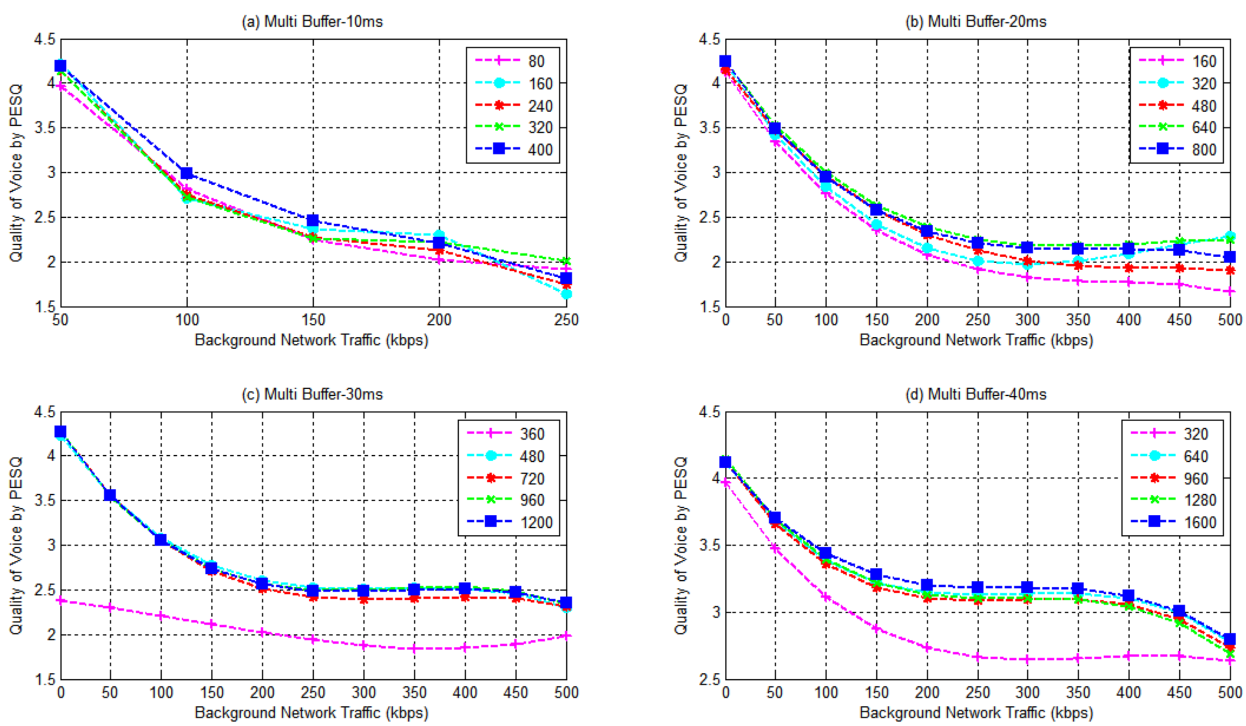

Fig. 5. Voice Quality vs. Traffic for different buffer sizes and different transmission intervals: (a)10ms, (b)20ms, (c)30ms, (d)40ms - Multi Buffer Management

The simulation result is presented in figures 5 and 6. The figures from 5 shows the simulation result for Circular buffer at $t_{tx} = 10, 20, 30 and 40 ms$. In circular buffer there is no mandatory constraint for size of buffer unlike Multi-buffer that buffer size should be integer multiple of packet payload. We start the simulation with an S_C which is a little bigger than packet payload. It is obvious in all of the related figures that a circular buffer with a size around twice that of packet payload is an optimum choice. Increasing the buffer size does not provide a much better voice quality than the optimum choice. In all figures the purple line indicates the optimum solution.
Figure 6 shows the graphs for Multi-Buffer simulation. If one compares the graphs between two buffer management policies, it may primarily seem that the Multi-buffer graphs are rather haphazard. Because sometimes in bigger buffer sizes, the voice quality degraded more than the smaller buffer sizes. But indeed this is the intrinsic property of the multi buffer management. Because the variation of voice quality in multi buffer management is not completely predictable. It can be seen from this criterion: if buffer is full and has not a sufficient empty space for the incoming packet, no matter the empty space size is one or $P\_Len - 1$. The point is that we will miss the whole packet. Due to this discrete behavior the buffer size especially in high jitter networks could not give us a good criterion to predict voice quality precisely.
It can be inferred from the result of our simulation that Circular buffer with a size twice of packet payload is an optimum solution for jitter management. Due to its continuous nature and its predictability, Circular buffer can mitigate the jitter effect along with efficient resource consumption.

References

[1] M. Hillenbrand, J. Gotze, P. Muller, *Voice over IP - considerations for a next generation architecture*, Software Engineering and Advanced Applications, 2005. 31st EUROMICRO Conference, pp:386 - 393

[2] F. Anjum, M. Elaoud, D. Famolari, A. Ghosh, R. Vaidyanathan, A. Dutta, P. Agrawal, T. Kodama, Y. Katsube, *Voice performance in WLAN networks - an experimental study*, Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE, 1-5 Dec. 2003Volume: 6, pp: 3504- 3508 vol.6

[3] J. Feigin, K. Pahlavan, M. Ylianttila, *Hardware-fitted modeling and simulation of VoIP over a wireless LAN*, Vehicular Technology Conference, 2000. IEEE VTS-Fall VTC 2000. 52nd Volume 3, Issue , 2000 pp:1431 - 1438 vol.3

[4] Lingfen Sun, E.C. Ifeachor, *Voice quality pre.diction models and their application in VoIP networks*, IEEE Transactions on Multimedia, Aug. 2006, Volume: 8, Issue: 4, pp: 809- 820

[5] Li Zheng, Liren Zhang, Dong Xu, *Characteristics of network delay and delay jitter and its effect on voice over IP (VoIP)*, IEEE International Conference on Communications, 2001. ICC 2001. 11-14 June 2001, Volume: 1, pp: 122 - 126 vol.1

[6] S.-Q.Li, *Study of information loss in packet voice systems*, IEEE Transactions on Communications, Volume 37, Issue 11, Nov 1989 pp:1192 - 1202

[7] OPNET Modeler/Radio Simulation Tool: http://www.mil3.com

[8] *Perceptional Evaluation of Speech Quality (PESQ), An Objective Method for End-to-end Speech Quality assessment of Narrow-band Telephone Networks and Speech Codecs*, ITU-T Rec. P.862, Int. Telecommun. Union, Feb. 2001

[9] RFC 3550, Standard 64, *RTP : A Transport Protocol for Real-Time Applications*, http://tools.ietf.org/html/rfc3550

[10] *Recommandation G.711*, http://www.itu.int/rec/T-REC-G.711/e