Mitigating the Latency Induced Delay in IP Telephony through an Enhanced De-Jitter Buffer

Asif Karim[1], Eshtiak Ahmed[2], Sami Azam[3*], Bharanidharan Shanmugam[4], Pronab Ghosh[5]

1 College of Engineering and IT, Charles Darwin University, Northern Territory, Australia
2 Faculty of Information Technology and Communication Sciences, Tampere University, Tampere, Finland
3 College of Engineering and IT, Charles Darwin University, Darwin, Australia
4 College of Engineering and IT, Charles Darwin University, Darwin, Australia
5 Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh

{asif.karim[1], sami.azam[3], bharanidharan.shanmugam[4]}@cdu.edu.au, eshtiak.ahmed@tuni.fi[2], pronab1712@gmail.com[5]

Abstract. IP Telephony or Voice over IP (VoIP) at present, is promising a shining future for voice services. There are several technical aspects which make the technology attractive; on the other hand, few technical loopholes and shortcomings make user's experience less than optimal and also brings forth significant security issues. This paper offers a technical dissection of the Quality of Service (QoS) of VoIP. "Signaling" part of VoIP has been discussed based on the Session Initiation Protocol (SIP) along with propositions to tackle problem like Jitter that often causes latency in communication. To address the issue of Jittering, an alteration in the working mechanism of De-Jitter buffer has been put forward where it is shown that addition of few extra variables within the De-Jitter buffer to synchronize the packet arrival and release timing can certainly improve the user experience. Reducing the latency is of prime importance to voice data services as it directly affects the acceptance trend of VoIP among mass consumers. The scale of improvement has also been compared to that of a normal jitter buffer as well as a detailed illustration have been provided on Session Initiation Protocol (SIP), a key component of the overall system that makes thing happen. The proposed modification in the de-jitter buffer has been illustrated along with positive results. It shows a one third improvement in the average latency, resulting into twice as better performance and nearly halved latency.

Keywords: Signaling, VoIP, SIP, Jitter, De-Jitter Buffer, IP Telephony.

1 Introduction

Voice over Internet Protocol (VoIP), which in other words, can be called IP telephony, is a technology which serves the purpose of both voice and multimedia communications using the Internet Protocol (IP) networks. More specifically, it refers to the services such as voice communication, SMS, multimedia etc. via the public Internet, rather than using the public switched telephone network (PSTN). VoIP with a bit enhancements and modifications, is indisputably destined to challenge the prevalent PSTN based phone system. VoIP means the transmissions of voice data traffic in packets over a data network primarily using protocols such as Session Initiation Protocol (SIP), this data network can be Internet or an Intranet.

There are several synonyms in the market of this technology which is based on various Internet Protocols (IP), such as: IP telephone, packetized voice etc. Usually the user first initiates call on one side of Internet, the analog voice signal is then digitized and compressed. Now this compressed signal is packetized into SIP packets and just simply delivered to the transmission media. At the receiving end, this process is just reversed [1]. The adoption of this technology is on the rise with each year. Some of the advantages for its widespread adoption includes easy and inexpensive installation as well as maintenance. Plus, VoIP scales up or down really efficiently, which is a real convenience for the companies as they do not always know how many employees it will have next year, it may have more, or employee number can also be slashed. So, phone line restructuring becomes a painless issue which hardly costs anything significant [2]. A VoIP phoning system supports all the traditional features of PSTN based lines and novel features are constantly being provided by the vendors as value added service. Further, setting up conference calls, an extremely important need for any businesses, is really easy in IP based telephony and offers commendable quality of service. Besides, VoIP can be used for Fax services as the traditional fax machines cost a lot over long distances and quality attenuation over long distance is also an issue [3]. In VoIP based services, these bottlenecks are non-existent. Simultaneous functionality is also an important feature of modern IP telephones, such as working with emails from the IP phone set at the same time carrying on conference calls using that very phone set. Thus the usage of VoIP is only set to increase.

However, as with many technologies in their forming stages, VoIP also has few performance issues to be considered. This paper not only addresses some of these design issues, but also brings forth few propositions to improve the shortcomings. Although VoIP is likely to take over the traditional telephone system, this mechanism won't be of much convenience to business organizations as they will have to risk their business due to some exacting loopholes in the VoIP's working model. For several reasons, the experience of using it can be hampered such as, high amount of traffic passing through same bandwidth or delay in the process of compressing and then decompressing the information from sender to receiver [22]. Another major concern of this technology is the security.

There are number of loopholes in this aspect [5]. Among myriad of security concerns on a VoIP network, ARP (Address Resolution Protocol) Spoofing has been one of the trickiest to deal with. In ARP attacks, the attacker broadcasts spoofed announcement of the MAC address forcing subsequent IP packets to flow through the attacker's host. This consents the eavesdropping of communications between two users [6]. The attacker can easily exploit the clear and plain format of a SIP packet to his advantage and can incur damage. There are lot mechanisms for encrypting textual data in a network such as public-private key, ciphers, hash functions, deploying image-keys etc. [7], but using cryptosystems for voice data often compromises the quality and latency even further. Cryptographic algorithms itself may show performance issues while processing [26]. While the bandwidth and its sharing is a problem that is user and condition specific, the delay factor and security concerns can be considered as problems on a technical level.

A VoIP system works by converting analogue voice signals into digital, then transmitting those as data over the broadband\internet communication channel. It is a very efficient way of making calls - for a start, once it is set up, it is considerably cheaper than normal phone lines.

1.1 Problem Formulation

Delay is generated in a VoIP network due to multifarious reasons. One of the significant reason for this is Firewall inspection of SIP packets. Delay is the amount of time it takes to reach a packet to its destination. Traditionally in PSTN, a delay of 150ms is acceptable. Therefore, in a VoIP model, if it exceeds 150ms, the users at both ends can become perplexed to who should continue speaking and who should be listening, as they experience an incessant gap between words or fractions of a word [4]. The limitations caused by this problem has been illustrated in a later section of this paper.

Even though there are multiple concerns to be addressed in terms of improving the VoIP services, this paper will look more into the Quality of Services (QoS) issues rather than security threats. The objective of this paper is to pinpoint few of the snags and propose a model to address the delay issue.

2 Relevant Works

Jitter is the congestion that occurs when a large number of Internet connections attempt to compete with each other at the same time; resulting in many tiny packets of information vying to use the same IP network; this can germinate significant delays in voice communication. There have been a good number of studies where this problem has been taken into consideration. One way includes making two different virtual switches out of one switch. However it has the problem of non-uniformity among switches in that not all the switch supports this kind of functionality. Further there has been some development in side areas that aids in helping the router to prioritize VoIP packets over normal data packets to process accordingly. One such technology is DiffServ. It is a Quality of Service (QoS) protocol that prioritizes IP voice and data traffic to help preserve voice quality, even when network traffic is heavy. It uses the ToS (Type of Service) bits in the IP header to indicate traffic importance and drop preference [11]. One other technique that has been developed to fasten the speed at which the packets flow is Header Compression. Header compression tries to leverage the repetitive bytes found in the header of a VoIP packet [12]. This attempt has somewhat been successful due to the significant size of the packet header to the payload. This technology is already integrated into many routers. However, if this applied too rapidly, the offshoot will be a negative one as it will, in fact, compound the jitter and delay factor. So, the best practice is to balance with care.

A number of studies have been done where the researchers proposed solutions of delay buffers to be implemented for controlling delays of all kinds including jitter [10, 21]. The idea here is that the voice packets will be stored in a buffer temporarily and will be sent out in a consistent manner, thus the packets will reach to destination in a certain interval of time. The users will then have a fairly good expectant time of hearing the voice from the other end and can adjust accordingly. This is a popular implementation and thus found its way into many hardware vendors doorstep. Many routers include this facility also. Another study highlights the importance of De-Jitter buffer and analyses behavior of jitter buffers with and without packet reordering capability and measures the additional packet loss incurred by packets dropped in buffer on top of the measured network packet loss [9]. However, it does have problems such as a larger De-jitter buffer, which causes higher latency inflicted on the system, while a smaller jitter buffer causes higher packet loss. A de-jitter buffer is architected to eradicate the effects of jitter from the decoded voice stream.

De-jitter buffer, as shown in Fig. 1, buffers each arriving packet for a short span before sending it out. This surrogates additional delay and packet loss for jitter. There are two types of de-jitter buffers. A **fixed de-jitter buffer** maintains a constant size whereas an **adaptive jitter buffer** has the faculty of adjusting its size dynamically in order to optimize the delay/discard trade off [4].

Both fixed and adaptive de-jitter buffers are equipped with the facility of automatically adjusting to changes in delay. For example if a step change in delay of 15 milliseconds transpires then some short term packet discarding can occur there, resulting from the change however the de-jitter buffer would be instantly realigned.

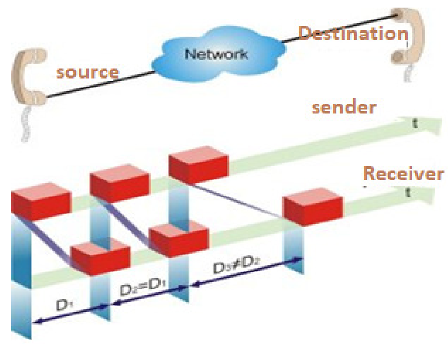

Fig. 1. De-Jitter Buffer.

The current implementation of the de-jitter buffer still has some limitations which affects its performance. As example, if there is too many packets in the buffer at a certain point of time, a lot of the packets have to be discarded as their TTL had already crossed the accepted margin, caused by the buffer overrun problem. Making the buffer larger to tackle the problem could result in much higher level of latency. As there is no specific way to say for sure when which packet will arrive, it could happen that packets will arrive in a different order than the actual or expected one. At this point, setting a fixed delay can create a latency or break in the conversation. Also, larger packets tend to incur network congestion, introducing more and more delay.

Number of studies and research have already been accomplished on this topic but the problems still remain intact. Many of the previous works actually cite a workaround instead of furnishing a complete solution. A recent work highlighted the issue of *Jitters* in VoIP, and proposed a solution that uses optimal packet Call Flow Routing (CFR) model in Real Voice Optimization to reduce Jitter in VoIP systems [8]. Another study focused on presenting how to achieve maximum VoIP connection quality with optimum de-jitter buffer delay [23]. The results show that, five-sixths of the connections were of either high or medium quality having a relatively small delay with the de-jitter buffer. In [18], the parameter selection of de-jitter buffer has been considered as the key to improve the service performance. The simulation results indicate that both the transmission delay and bit error rate can be reduced with packet size and buffer length being optimized. Table 1 summaries the above-discussed related works.

Table 1 Summary of the above discussed systems

Authors	*Proposition*	*Shortcomings*
[10] Salem et al.	Use large memory buffer to hold the packets to transmit later in a consistent manner	Does not address the associated higher Latency due to large buffers
[21] Voznak et al.	Also deploys similar buffer with extended memory usage	Suffers from the Latency issue that can be seen with large memory buffers
[9] Rodbro & Jensen	Uses Header Compression	Unless balanced properly, Header Compressions can increase Jitters.
[8] Adebusuyi et al.	Uses techniques such as Real Voice Optimization	Does not perform well in 4G based VoIP services
[23] Lebl et al.	Takes an approach to address a number of different parameters related to QoS	The outcome is not consistent and may vary according to the buffer size

All of these works have proposed good solutions using de-jitter buffers, however, some of the existing problems with de-jitter buffer such as the buffer overrun problem and latency induced by bigger buffer size have not been addressed. This research aims to propose a potential solution in this regard.

3 VOIP Technique and Network Structure

After the connection is established, the voice data is converted into digitized from. However, as digital data requires a lot of bits, it is compressed. Now the sample of voice is packetized to be sent over the Internet. These packets are wrapped with RTP (Real Time Protocol). The packets are synchronized by means of Identification Number, inserted in the RTP header for each packet. Each packet is then further wrapped around UDP (User Datagram Protocol). UDP carries RTP as payloads. As the packets reach the other end, the whole process is reversed, that is the packets are disassembled and put into the proper order; digital bits are extracted out and decompressed, converted to analog signal and finally delivered to the user's handset. The role of SIP (Session Initiation Protocol) here is that it encompasses all the aspects from call initiation to call termination; and to do that it includes number of related protocols such as: RSVP, RTP RTSP, SAP and SDP.

VoIP can be configured and settled into different topologies and configurations. But before discussion of how the VoIP is configured, some related terms such as Media Gateways and Call Managers or Soft Switch need to be understood.

3.1 Media Gateways and Call Managers

Media Gateways (Servers) act as a translation unit between disparate telecommunications networks [1]. VoIP Media Gateway (MG) performs conversion between VoIP Media Gateways perform the conversion between TDM voice to Voice over Internet Protocol. Media Gateways are controlled by a Media Gateway Controller known as Call Managers or Soft Switches, providing the call control and signaling functionality [1]. Communications via these two entities are done through protocols such as SIP (which will be discussed later), MGCP or H.248.

AMG has to perform several operations in a VoIP network. Primarily:

- Carries out A/D conversion of the analog voice channel.
- Transforms a DS0 or E0 to a binary signal compatible with IP or ATM.
- Multi-vendor interoperability.
- Transport of voice mainly using IP-based RTP/RTCP.

3.2 Topologies

There are number of ways in which a VoIP network can be structured [13]. One of which is the use of PC and the employment of a router. An easy and inexpensive way to use VoIP is the 1:1 VoIP Gateway architecture.

.

3.3 Session Initiation Protocol (SIP)

Session Initiation Protocol (Developed by IETF) is a protocol which is termed as an application layer signaling protocol. SIP primarily deals with interactive communication using multimedia elements. Signaling means to initiate, modify parameters and terminate sessions between end users. SIP does exactly this. SIP calls may be terminal-to-terminal, or they may require a server to intervene [15]. Note that SIP shares a close affinity with IP. A SIP message looks very much like a HTTP message and thus in plain text. SIP addresses are also very much like emails such as sip:name@a_domain.com.

The user can usually set up call in two modes, called redirect and proxy, and servers are designed to handle these modes.

Besides signaling, SIP can also be used to invite other people in a conversation or can simply be employed to start a new session. SIP is completely independent of the type of content and the mechanism used to convey the content [16]. SIP is there only for signaling purposes, not to worry about any other aspects of media flow or type. SIP simply works as glue inside the whole process among different layers and protocols.

To support session services, there are five facets of multimedia session management for SIP:

- User Locality: Identifying which end- system will be used for communication.
- User Potentiality: Determining the media and media parameters to be used for this communication.
- User Availability: Determining whether or not the called party is enthusiastic to engage in communications.
- Session Setup: Setting up the session parameters at both the called and calling parties.
- Session Management: Including the transfer and termination of sessions, the modifying of session parameters, and the invoking of session services.

3.4 SIP Elements

Before going into the discussion of how SIP works some terms of the world of IP telephone using SIP need to be understood.

1) User Agents (UA): An endpoint in a VoIP system, normally IP phones or media gateways. It usually has an interface towards the users. When user A wants to call user B, he fires up the appropriate program containing a SIP UA. The user interacts with user B through the interface. Naturally, user B at the other end also has a similar kind of program; and he can either accept or reject the invitation form user A. The UA also incorporates various media tools to actually handle the media content. Normally the UA establishes the session and the media tool it contains handles the content of the session. In many applications, these two comes under one program. For example a Video Conferencing Software [17].

2) Redirect Server: Redirect servers aid the SIP UAs by furnishing location information where the user can be sensed. From Fig. 5, it can be seen how Redirect Servers perform their action. For example, say user A wants to call user B (located at domain.com) and thus sends an invitation. But at domain.com, a redirect server is positioned to handle the incoming calls, so, the UA of user A, instead of contacting with the intended address, either SIP:A@192.168.1.1 or SIP:A@192.168.1.2. These addresses are different from the intended address. The redirect server may also propose which of the address is more likely. It then simply sends back messages that contain this information and the UA of A tries to establish session with these addresses. However, it can also return address of other server which may have more information, in other words, this process can be chained [17].

3) Proxy Server: The aim of the proxy is same but in this case, instead of returning the possible address information, the proxy tries those addresses by itself in order to establish the session. Note that in that case too, there may be intermediary proxies and location servers. Oftentimes, a single server can be configured as both redirect and proxy; and these two can also co-exist in a single system [17]. In practice, a SIP based proxy server facilitates the establishment and maintenance of communication channel between two SIP addresses. Any SIP device can communicate to another SIP device, but in order to achieve that, they deploy a go-between, called a SIP proxy, to begin the communication, which then drops out, allowing point-to-point communication.

4) SIP Register: An application typically running on a server that aids the UAs to register themselves so that they are able to receive calls. The registrar can or can't be co-located with the proxy or redirect servers. Generally the user provides information about his location [17].

4 Limitations

This section will have a detail discussion on the limitations of VoIP mentioned earlier.

Among various and diverse kinds of performance issues relating to VoIP, “Delay” is the most crippling one. The delay in a VoIP network can be for many reasons. A VoIP network must be able to deliver packets within a standard max of 150ms. This puts a major consideration on how much security overhead can be included. Also, it leaves very little margin for error recovery in case of an error in packet delivery. One type of delay is called **fixed delay** and are mostly occurred within the signal processing systems [19], such as the processing delays within the voice coders/decoders (codecs) that make the A/D signal conversions, and are also detected within the physical transmission systems, such as the copper pairs. The **variable delays** come from queuing times at packet processing points, such as routers and switches, plus transmission variables, such as the path that a particular packet, or series of packets takes within the network [19]. Firewall routers such as double socket routers re-establish an IP packet flow on the inner side of the router after it has been disconnected on the outer side. This aids in a regulating IP packets in a consistent diction but also introduces delays. Sometimes there is a delay when the firewalls update the iptables.

Also, larger packets tend to incur network congestion, introducing more and more delay along with the delay in each hop the packet travels through. This type of non-uniform, variable delay is known as **jitter**. Jitters transpire because not all the packet will take the same route, this actually is more of a problem as this can introduce the arrival and processing of packets out of order. RTP is based on UDP, so there are no reassembles at the protocol level. However, rearranging can be done in application layer but it tends to be slow, which needless to say, compounds the delay malefactor. The following figure (Fig. 2) details how this Jitter happens:

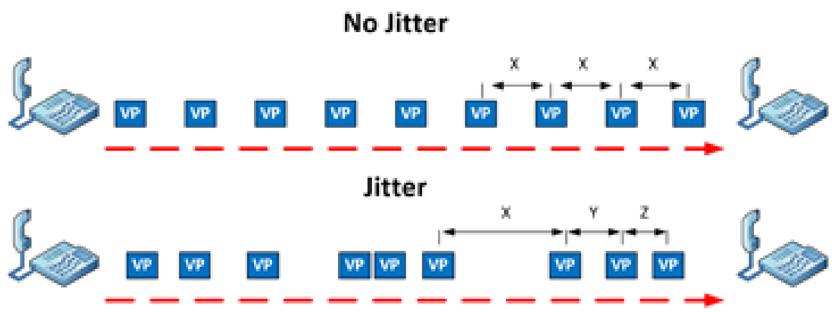

Fig. 2. VoIP Jitters [14].

In Fig. 5, the amount of time (say, z_p) it requires for packets A and B to send and receive is equal when there is no jitter. But when the packets confronts delay in the network, this uniformity of z_p is affected, and the expected packet may be received *after* it is expected. This is why a *jitter buffer*, which hides inter-arrival packet delay variation, is essential [19]. Voice packets in IP networks have highly variable packet-inter-arrival intervals. Common practice is to count the number of packets that arrive late and create a ratio of these packets to the number of packets that are successfully processed. Then the ratio can be utilized to adjust the jitter buffer to target a predetermined, allowable late-packet ratio; this in most cases, compensates for delays. Note that these buffers can be either dynamic or static. Quality of Service (QoS) is a make or break for the acceptance of VoIP. The ever exponential growth of internet user and the development of multitude of multimedia oriented services exerts increasing demands on web servers and network links, which can cause overloaded end-servers and congested network links [20]. These issues almost make it indispensable that the design of VoIP, or for that matter any internet based application, should strive to achieve every bit of possible design efficiency.

5 The Proposed Solution

Several solutions have been introduced to get an upper hand on obstacles like delay and jitter. In this section this study will present a proposed way around for the issue of Jitter.

5.1 Modification in De-Jitter Buffer

To tackle the problem of jitter and fixed delay, the implementation of de-Jitter buffer is common. However, as has been said before, if there are too many packets in the buffer, many of the packets having their TTL already crossed the accepted margin, must be discarded due to the buffer overrun problem. Also, the next problem is if the size of the buffer is large then it causes too much latency. Because there is no guarantee that after exactly what span of time, the packets will arrive. Thus, one packet that has been formed later may appear earlier in the buffer than a packet that has been formed and sent before the previous one.

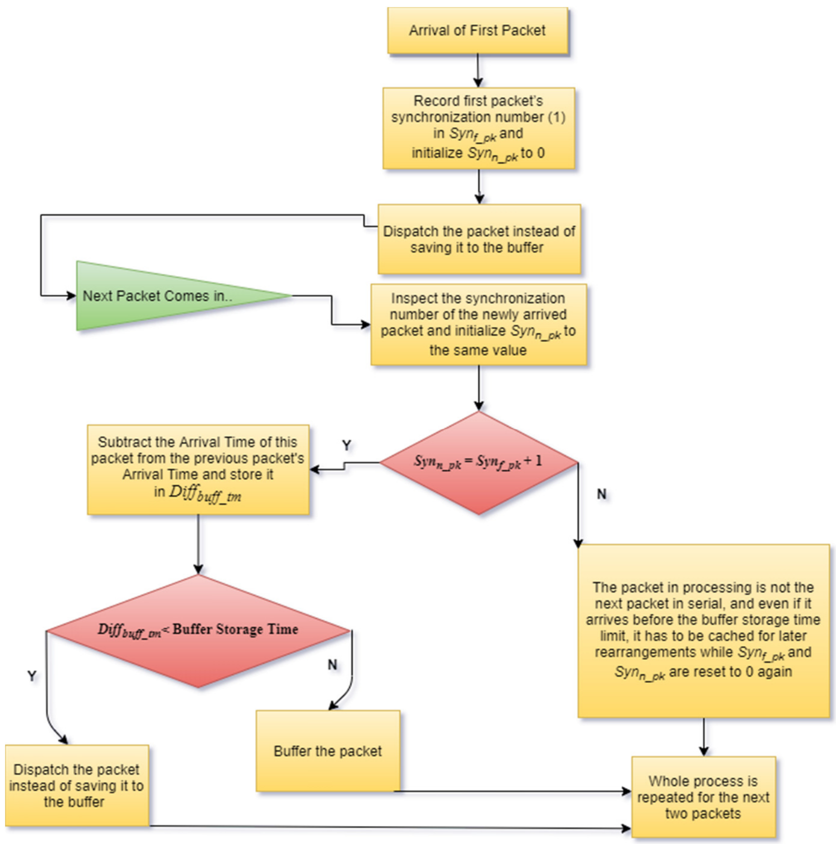

Fig. 3. Flow Diagram of the Proposed Modification.

Thus, if the delay length of the buffer is set to 200 ms, all the packets will be played out after 200 ms, thus it may create latency or break in the conversation for a packet that has been reached earlier. So, the proposed solution will bring some modification in the working procedure to the de-jitter buffer. The modified buffer, as shown in Fig.3, will have two extra variables, say $\mathrm{Syn}_{\mathrm{f\_pk}}$ and $\mathrm{Syn}_{\mathrm{n\_pk}}$; $\mathrm{Syn}_{\mathrm{f\_pk}}$ will record the first packet's synchronization number, (that is, 1), i.e. the packet serial number, inscribed in the header of the packet itself. It will then instead of saving the packet, will dispatch it. Now the next packet comes, at this moment $\mathrm{Syn}_{\mathrm{f\_pk}}$ equals to 1 and $\mathrm{Syn}_{\mathrm{n\_pk}}$ equals to 0.

The buffer will inspect the synchronization number of this packet and if this number is one greater than Syn_{f_pk}, i.e. $Syn_{n\_pk} = Syn_{f\_pk} + 1$, then it will subtract the arrival time from the previous packet's arrival time and store it in another variable, say $Diff_{buff\_tm}$. Now, if $Diff_{buff\_tm}$ is less than the buffer storage time, it will again instead of saving the packet, will forward it.

However, if $Syn_{n\_pk} <> Syn_{f\_pk} + 1$ (unequal), then the packet in processing is not the next packet in serial, and even if it arrives before the buffer storage time limit, it has to be cached for later rearrangements while Syn_{f_pk} and Syn_{n_pk} are reset to 0 again. For the next two packets, the same mechanism is repeated. This way, cached packets that arrive before the expected delay margin, will be propagated instead of probable drop-out after a certain period, and it will not create latency, in-effect improving the overall performance. Synchronization is also maintained using the same logic. However, there will still be some codec related delays that needs to be accounted for.

6 Results And Analysis

The impact of reducing the delays in jitter can have significantly strong effect in the reduction of latency. In this section this dramatic effect will be explained through facts and figures. The observable latency in a VoIP service are often the combination of parameters such as *Average Jitter* and *Average Latency* into one single metric known as "Effective Latency". It is calculated as:

$$\text{effective\_latency} = \text{avg.\_latency} + (2 * \text{avg.\_jitter}) + 10.0 \quad (1)$$

In equation 1, effect of jitter are usually being doubled as its impact is relatively high on the overall performance, and a general constant of 10 ms (milliseconds) is added to account for the delay caused by the respective codec. *Average Latency* can be measured using usual concepts [25], for instance, if the sum of measured latency is 960 ms and the number of latency sample is 30, then the average latency is 32 ms (960/30). *Average Jitter* is calculated by comparing the interval when RTP packets were sent to the interval at which they were received [24]. For instance, if the first and second packets leave 75 ms apart and arrive 95 ms apart, then the jitter is 20 ms (95-75). Now if we sample such differences, for example 5 times, and receive series of jitters such as 20, 45, 36, 15 and 14, then the average jitter is 26 ms; it basically follows the equation 2 (***P*** denotes the specific packets departure and arrival time in ms) .

$$\text{Avg.\_jitter} = \frac{\Sigma\left((P_2 - P_1) + (P_4 - P_3) + \ldots + (P_n - P_{n-1})\right)}{N} \quad (2)$$

Now, before implementing the improvements in a normal De-Jitter buffer, the average latency for 5 consecutive minutes had been recorded as 25, 32, 33, 28 and 50 milliseconds and the corresponding average jitters as 35, 48, 62, 42 and 65 milliseconds (Using equation 2). These two parameters have now been used to calculate the effective latency using equation 1 and the result was found to be a sharp increase in effective latency that the user will experience.

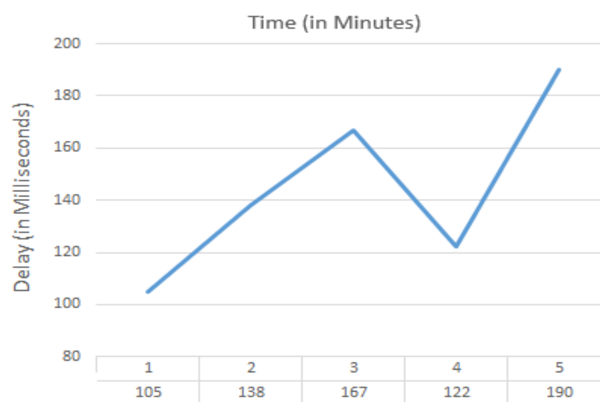

Fig. 4. Effective Latency in Normal Jitter Buffer.

The effective latency in milliseconds for five consecutive minutes have been calculated as 105, 138, 167, 122 and 190 respectively using equation 1. Fig. 4 illustrates the rise of latency, or in other words, declination of quality of service (QoS) in the conversation graphically.

It can clearly be seen from Fig. 8 that even though there are some improvements around fourth minute (T4), the general trend shows an increase in effective latency, which causes disruption in service to heighten sharply as well. Once there is delay associated with jitters starts to materialize, the quality oftentimes may deteriorate rapidly in subsequent seconds.

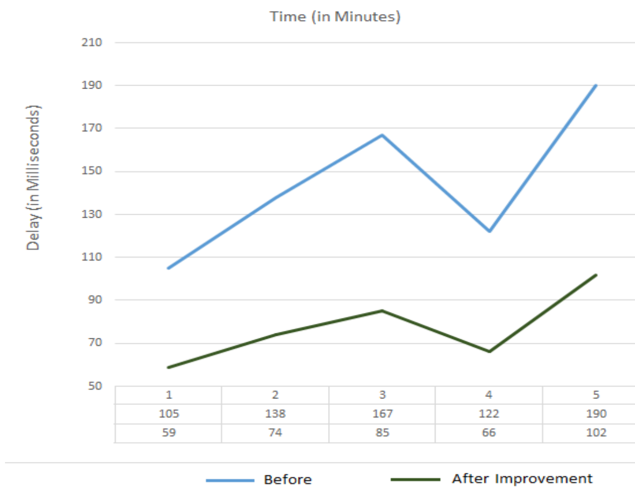

Fig. 5. Comparison of Effective Latency experienced before and after the implementation of the recommended improvements in Jitter Buffer.

Though sometimes the curve will have opposite direction, that is quality may get better with time, but that can be effect of other network factors, such as throughput, bandwidth or congestion getting improved.

As can be seen from Fig. 5, a one third improvement in the average latency due to the improved design in the Jitter buffer may double the performance in the quality of the conversation. It basically halves the effective latency. Such is the importance and effect of improvement in the jitter buffer, which can clearly affect considerable positive impression in the overall quality of service. Thus, an efficient design and construction of De-Jitter buffer indeed is a crux part of a VoIP system. Additionally, the proposed study presents better result in terms of comparison with earlier discussed systems that include application of large memory blocks.

7 Conclusion

As data traffic continues to increase and cross that of voice traffic, the convergence and integration of these technologies will not only continue to rise, but also will streamline the way for a truly unified and seamless avenue of communication. VoIP can provide significant benefits and cost savings. Since voice and data traffic can be integrated, the necessary infrastructure to provide both services are reduced. Also, bandwidth will be utilized properly as bandwidth on a network is rarely completely flustered with data traffic, and circuit switching calls usually waste a lot of bandwidth in general. VoIP is still in its embryonic condition in many areas of it development. It truly is a revolutionary technology that will pose many challenges to Circuit Switched infrastructure.

This paper discusses the issues of latency caused by Jitter and also proposes an improved De-Jitter buffer which can certainly have a positive effect on the technology. The implementation results indicate many positives. It shows a one- third improvement in the average latency due to the improved design in the Jitter buffer, resulting into potentially doubling the performance in the quality of the conversation. Additionally, it the effective latency factor has been observed to be nearly halved. The detailed working procedure of SIP and the importance of Quality of Service (QoS) in a VoIP network have also been discussed. Although the results are very promising and could potentially improve the quality QoS of the VoIP service, more improvements can be done. This solution needs to be tested on a larger scale to identify possible improvement suggestions. Traditional noise issues such as additive and subtractive noises may affect the performance, thus further work is needed to factor against such issues for a better Quality of Service.

References

1. Richard Kuhn, Thomas Walsh and Steffen Fries, "Security Considerations for Voice over IP Systems", National Institute of Standards and Technology, U.S. Department of Commerce, Jan. 2005.
2. Lydia Uys, "Voice over internet protocol (VoIP) as a communications tool in South African business", African Journal of Business Management, Vol.3 (3), pp. 089-094, 2009.
3. "Adding Reliable Fax Capability to VoIP Networks", Dialogic White Paper, 2010.
4. Carlos D. Nocito and Michael S. Scordilis, "Monitoring jitter and packet loss in VoIP networks using speech quality features", IEEE Consumer Communications and Networking Conference (CCNC), Jan. 2011.
5. Eduardo B. Fernandez, Juan Palaez and Maria Larrondo-Petrie, "Security Patterns for Voice over IP Networks", Journals of Software, 2(2), pp. 19-29, 2011.
6. Zouheir Trabelsi and Wassim El-Hajj, "ARP Spoofing", Information Security Curriculum Development Conference, Sep. 2009.
7. Asif Karim, "Multi-layer Masking of Character Data with a Visual Image Key", International Journal of Computer Network and Information Security, 2017. 9(10): p. 41-49, Oct. 2017.
8. Kehinde Adebusuyi, Ezea Hilary, and Gerald Kelechi Ijemaru, "A Research of VoIP Jitter on Packet Loss in GSM Voice over IP Systems", International Journal of Innovative Technology and Exploring Engineering (IJITEE). Vol-6, No-12, pp. 5-12. Dec. 2017.
9. Christoffer Rodbro and Søren Holdt Jensen, "Time-scaling of sinusoids for intelligent jitter buffer in packet based telephony". Speech Coding, IEEE Workshop Proceedings, 2002.

10. Ahmed M. Abdel Salam, Wail Elkilani and Khalid M. Amin, “An Automated approach for Preventing ARP Spoofing Attack using Static ARP Entries”, International Journal of Advanced Computer Science and Applications, 5(1), 2014.
11. Jie Li and Qi-fan Cui, “The QoS Research of VoIP over WLAN”, International Conference on Communications, Circuits and Systems. 2006.
12. Yichuan Wu, Jianping Zheng and Kui Huang, “The performance analysis and simulation of header compression in VoIP services over cellular links” 9th Asia-Pacific Conference on Communications, Sep. 2003.
13. Weyl Wang, Kevin Bourg, Brian Lane and James Farmer, “VoIP Architecture,” FTTx Networks, pp. 307-322, Dec. 2017.
14. Amany S. Amin Hanem M. El-Sheikh, “Scalable VoIP Gateway,” The 9th International Conference on Advanced Communication Technology, Feb. 2017.
15. Daniel Noworatzky. How to troubleshoot voice quality problems in VoIP phone systems. http://info.teledynamics.com/. Accessed: 8.9.20, 2018.
16. Carlos Davids, Vijay K. Gurbani and Scott Poretsky, “Terminology for Benchmarking Session Initiation Protocol (SIP) Devices: Basic Session Setup and Registration,” RFC 7501, Apr. 2015.
17. Xiaohui Yang, Ram Dantu, and Duminda Wijesekera, “Security Issues in VoIP Telecommunication Networks”, Handbook on Securing Cyber-Physical Critical Infrastructure, pp. 763-789, 2012.
18. Gao Hui-sheng and Hang Cheng, "Simulation of De-Jitter Buffer Parameter Optimization of in Power PTN", Study on Optical Communications 1, 2018.
19. Li Zheng, Liren Zhang, and Dong Xu, “Characteristics of network delay and delay jitter and its effect on voice over IP (VoIP)”, IEEE International Conference on Communications, Aug. 2002.
20. Jin K. Lee and Kang G. Shin, “NetDraino: Saving Network Resources via Selective Packet Drops”, Journal of Computing Science and Engineering, Vol. 1, No. 1, pp. 31-55, Sep. 2007.
21. Miroslav Voznak, Adrian Kovac and Michal Halas, “Effective Packet Loss Estimation on VOIP Jitter Buffer”, Springer, Lecture Notes in Computer Science, LNCS-7291, pp.157-162, 2017.
22. Ng Liang Shen, Noraniah Abdul Aziz, Tutut Herawan, “End-to-End Delay Performance for VoIP on LTE System in Access Network” In Lecture Notes in Computer Science, Vol 7804. Springer, Berlin, Heidelberg, 2013.
23. Aleksandar Lebl, Mladen Mileusnić, Dragan Mitić, Žarko Markov, Branislav Pavić, “De-Jitter Buffer Role in Improving VOIP Connection Quality – Examples from Practice”, International Scientific Conference On Information Technology and Data Related Research, 2017.
24. Muhamad Amin, “Voip Performance Measurement Using QoS Parameters”, The 2nd International Conference on Innovations in Information Technology, 2005.
25. P.C. Saxena, Jasola Sanjay, Ramesh C Sharma, “Impact of VoIP and QoS on Open and Distance Learning”, Turkish Online Journal of Distance Education 7(3), 2006.
26. Md Enamul Haque, S.M. Zobaed, Muhammad Usama Islam, Faaiza Mohammad Areef, “Performance analysis of cryptographic algorithms for selecting better utilization on resource constraint devices”, 1st International Conference of Computer and Information Technology (ICCIT), 2018.