Java Based VoIP Performance Monitoring Tool

Husna Zainol Abidin, Mohd Ameer Yuslan Razmi, Farah Yasmin Abdul Rahman, Ihsan Mohd Yassin

***Abstract*— This paper describes the development of a user friendly Voice over Internet Protocol (VoIP) monitoring tool. This monitoring tool was developed by using Java Programming and tested on a VoIP network testbed where it uses Session Initiation Protocol (SIP) and Real-time Transport Protocol (RTP) to transmit the voice signal with G.711 modern codec. Quality of Service (QoS) parameters that are being measured are the delay, throughput, jitter and packet loss of the VoIP traffic. This monitoring tool's results are considered acceptable since these results were compared with other monitoring tools and it is found that the results obtained from the developed tool are relatively similar to the other tools.**

***Index Terms*— Voice over Internet Protocol (VoIP), Quality of Services (QoS), Java, monitoring**

I. Introduction

Voice over Internet Protocol (VoIP) is an emerging technology based on open standards that allows voice data to travel across the Internet rather than using the Public Switched Telephone Network (PSTN). In VoIP, the voice signal is digitized, compressed and sliced into packets which will be sent with other packets across the packet switched network. However, there are features embedded in the network to ensure that the priority will be given to the voice packets to travel in order to reduce the latency and delay while transmitting the voice packets. At the receiving end, packets will be re-assembled as a normal sound voice call. There are two protocols that are normally used to transmit this voice signal which is Session Initiation Protocol (SIP) and Real-time Transport Protocol (RTP) [1]. SIP is an application-layer control protocol that can establish, modify and terminate multimedia sessions or calls [2] while RTP provides end-to-end network transport functions suitable for applications transmitting real-time data over multicast or unicast network services [3].

Fig. 1 shows a simplified block diagram of a VoIP operation from an analog signal which was derived from a standard telephone, digitized and transmitted over the IP via intermediary devices. After it reaches the destination, it will be converted back to analog telephony using a similar device which is an input to a standard telephone.

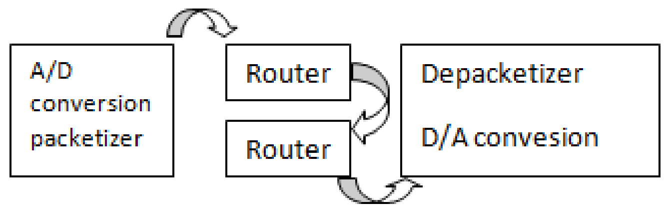

Fig. 1. Elements of basic operation of VoIP

Codec is used to convert an analog voice signal to digitally encoded version. Codec vary in different sound quality, bandwidth required, delay, the computational requirements and others. Thus, it must be chosen with regards to the network that will be used to transmit the voice signal. Table I shows the comparison of VoIP codec in terms of required sampling rate, bandwidth, nominal bandwidth and payload size.

TABLE I. Comparison Of VoIP Codec [4]

Codec	Measurement			
	Sampling Rate (kHz)	*Bandwidth (kbps)*	*Nominal Bandwidth (kbps)*	*Payload Size (ms)*
G.711	8	64	87.2	20
G.722	16	48	Unknown	30
	16	56	Unknown	30
	16	644	Unknown	30
G.723.1	8	5.3	20.8	30
	8	6.3	21.9	30
G.726	8	16	Unknown	20
	8	24	47.2	20
	8	32	55.2	20
	8	40	Unknown	20

This paper describes the development of a user friendly Java based monitoring tool which is dedicated to measure the VoIP performance. The available open source networking tools such as VQManager[5] and Wireshark[6] do not have capability to measure throughput, that is one of the essential network QoS parameters. Besides, results obtained from most of web based monitoring tools are not really accurate as they need to get back to their server which might affect the

Manuscript received May 20, 2010.
Husna Zainol Abidin is with the Faculty of Electrical Engineering, Universiti Teknologi MARA, 40450, Shah Alam, Selangor Malaysia. (phone: +603-55436124; fax: +603-55435077; e-mail: husnaza@salam.uitm.edu.my).
Mohd Ameer Yuslan Razmi is a student of the Faculty of Electrical Engineering, Universiti Teknologi MARA (e-mail: ameer131@yahoo.com).
Farah Yasmin Abdul Rahman is with the Faculty of Electrical Engineering, Universiti Teknologi MARA, 40450, Shah Alam, Selangor Malaysia, on PhD study leave. (e-mail: farahy@salam.uitm.edu.my).
Ihsan Mohd Yassin is with the Faculty of Electrical Engineering, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, Malaysia. (email: ihsan_yassin@salam.uitm.edu.my).

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

performance measurement particularly the delay. Inspired by these issues, this tool is developed to address those setbacks. The rest of the paper is organized as follows. Section II will discuss some related works that have been done by other researchers while Section III focuses on the development process. Results are presented in Section IV while Section V will finally conclude the whole paper.

II. Related Works

The Next Generation Network (NGN) which is catered for different type of traffics consists of real time and non real time traffics, needs an effective and practical network performance monitoring tool. Due to this scenario, researches had carried out some research works in developing monitoring techniques to observe the performance of real time traffic particularly on VoIP.

Reference [7] which was written by de Silva Jr. et al. had analyzed the QoS of VoIP on SIP by using Java Programming. The performance parameters that were measured are delay, jitter and packet loss. The tests have been carried out on a testbed in six different scenarios. However, this study did not come out with a user friendly interface. A new novel system that can monitor VoIP service and detect VoIP network threats practically was proposed in [8]. The proposed system collects attributes of VoIP traffic based on NetFlow, and executes monitoring based on statistic and behavior.

Kim et al. [9] have proposed a network monitoring method for multimedia services such as VoIP and Internet Protocol Television (IPTV) via delay measurement by using Real-time Transport Control Protocol (RTCP) while in reference [10], Kitatsuji et al. proposed a high-speed table lookup method to monitor the VoIP performance in a mobile network.

A Real-Time Communication Monitoring framework or RTC-Mon was presented by Fusco et al. in [11]. RTC-Mon provides a platform for the quick development of a high-speed and real time monitoring applications which focuses on VoIP traffic. Hershey et al. in [12] found out that available monitoring tools are lack of support in monitoring Quality of Experience (QoE) which represent the perception of quality experienced by the end users. Thus, they had come out with a new approach that addressed the limitation by aggregating the observation of real time application particularly VoIP.

III. Methodology

A. *Development of the Tool*

This monitoring tool was developed by using Java programming language. Java programming are normally used for distributed applications as it split up with the client-server model and provides real distributed processing which is also suitable for the development of Internet applications [7].

NetBeans Integrated Development Environment (IDE) version 6.8 was used to develop this monitoring tool that has a user friendly Graphical User Interface (GUI). Beside NetBeans, there are another three popular Java IDEs that also can be used to develop this monitoring tool which are JCreator, Eclipse and BlueJ. NetBeans was chosen because NetBeans has capability to do graphical programming while source codes are still needed in JCreater, Eclipse and BlueJ.

Software called Iperf [13] was used for network programming. Iperf was developed as a modern alternative for measuring maximum Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) bandwidth performance. It is commonly used as a network testing tool that can create TCP and UDP data streams which user can specify a specific datagram size and measure the throughput of a network that is carrying them. Hence, Iperf is suitable to use as the back-end of a monitoring tool. Furthermore, Iperf allows the tuning of various parameters and UDP characteristics, and it gives reports about throughput, bandwidth, jitter and also packet loss. The tool which was written in C++ can be run over any network and output standardized performance measurements. Therefore, it is suitable to be used for comparison of wired and wireless networking equipment and technologies in an unbiased way. The measurement methodology can be manipulated by the user since it is an open source software [13].

Iperf is capable to show which IP and port of the server that have been connected as well as the IP and port at our end. Next, it shows how many data and bandwidth that had been transferred during a specific period of time from the server side. After completing a specific period of time, the server will report the measurement to the client. Jitter and packet loss that occur the data transmission can be determined from this server report. However, Iperf did not give the value for delay parameter. Thus, an additional tool is used to perform it. As we know, voice packets are normally transmitted by using UDP but UDP is connection-less where we could not know whether the packets had successfully reached the destination or not. Due to that reason, the exact delay between the two hosts could not be measured. This problem can be solved by using a Ping command where the network condition of TCP link utilization is low and UDP packet has a priority to travel along the network.

B. *Testing the Tool*

Initially, a sample of voice was recorded and transmitted over a network testbed. Software called “G.711 Converter Tool” is used to perform this operation as shown in Fig. 2 [14]. After a sample voice is successfully transmitted over the network, an online voice conversation is tested on the network to measure the reliability of the network. Again, the monitoring parameters for throughput, bandwidth, jitter, packet loss and delay are measured. A soft phone called X-lite is used in this online voice conversation. Several configurations must be done before this soft phone can be used such as the display name, username, password, authorization user name, domain name and so on.

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

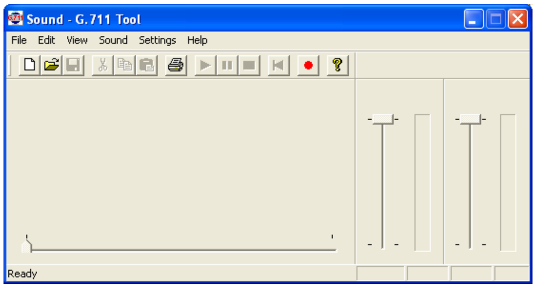

Fig. 2. G.711 Converter Tool

Fig. 3 shows the flowchart of the monitoring process. Results obtained from the GUI are compared with two open source monitoring tools that are widely used known as VQManager and WireShark. In order to increase the utilization of the link, call generator software named SIPp, is used [15].

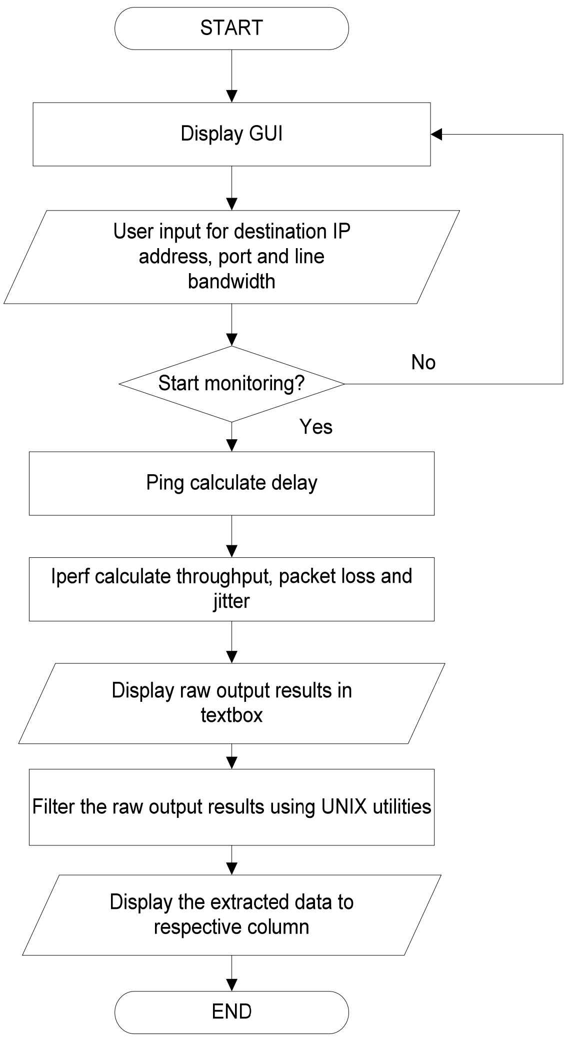

Fig. 3. Flowchart of the Monitoring Process

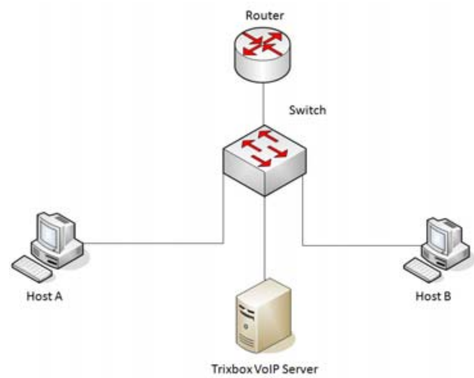

Fig. 4. Topology of LAN VoIP Network Testbed

The monitoring tool was tested on a testbed as shown in Fig. 4. The workplace uses a Network Card Interface (NIC) with speed of 10Mbps at the VoIP server and the codec used is aLaw G.711. G.711 is chosen because there is no compression being done on the transmitted data. Therefore, it offers the lowest latency which consequently will produce the similar voice as being produced by PSTN. However, G.711 codec takes more bandwidth compared to other codecs which is up to 84 Kbps including all TCP/IP overhead. This issue can be resolved by using network that consist of a larger bandwidth [16]. A router has been added between both host A and host B computer to let the traffic pass trough VoIP server before reaching to the end user.

IV. Results and Discussions

Fig. 5 illustrates the monitoring GUI that has been produced. The results are measured not simultaneously for each type of monitoring tool since the NIC can only be listened once at a time. Consequently, these results are having slightly differences between each other. Measurements are taken up to 400 calls per second which is the maximum load that the VoIP server is able to serve. Otherwise the server will be crashed.

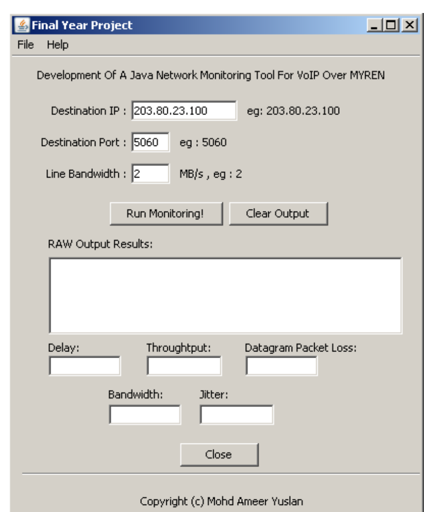

Fig. 5. Developed GUI for this project

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE II. COMPARISON OF THROUGHPUT

No of calls / second	Measurement		
	Developed Monitoring Tool (kbps)	*VQManager (kbps)*	*WireShark (kbps)*
0	59.80	n/a	n/a
50	59.53	n/a	n/a
100	58.37	n/a	n/a
150	57.57	n/a	n/a
200	57.80	n/a	n/a
250	55.50	n/a	n/a
300	54.97	n/a	n/a
350	54.17	n/a	n/a
400	53.63	n/a	n/a

Table II depicts the comparison of throughput readings between the developed monitoring tool with VQManager and Wireshark for VoIP. From the table we can see that the produced tool is able to measure throughput compared to other tools. This is because VQManager and WireShark do not offer any throughput measurement for UDP traffics.

TABLE III. COMPARISON OF PACKET LOSS

No of calls / second	Measurement		
	Developed Monitoring Tool (%)	*VQManager (%)*	*WireShark (%)*
0	0	0	0
50	0	0	0
100	0	0	0
150	0	0	0
200	0	0	0
250	0	0	0
300	0	0	0
350	0	0	0
400	0	0	0

Table III shows measurement of packet lost for all three networking tools. Based on the table, it can be seen that there is no packet drop while transmitting the voice traffic over the testbed as there is no background traffic has been injected into the testbed. However, it is envisaged that the packet drop will occur if this tool is deployed on the real network as there will be a lot of background traffics traverse in the network. As stated in reference [17], a packet loss of below 5% does not affect conversation. Quality of voice is affected by delay and jitter. Thus, it can be concluded that packet loss is not that crucial in provisioning QoS for VoIP.

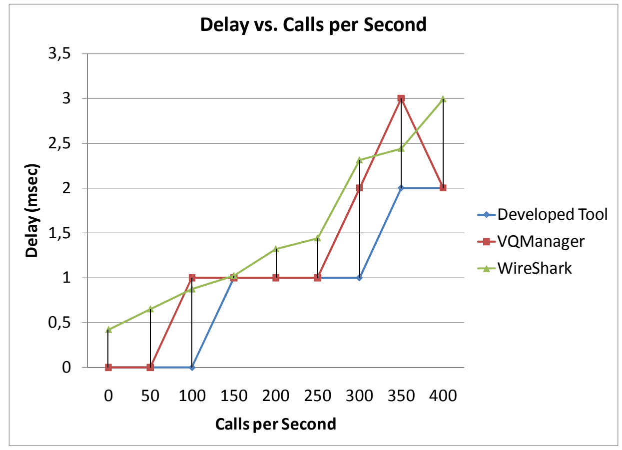

Fig. 6. Graph of delay comparison

The developed tool is seen to produce the lowest delay compared to VQManager and WireShark as depicted in Fig. 6. However, this result can be considered acceptable as the graph trend is relatively similar with another two tools. As illustrated in Fig. 7, it can be seen that the jitter obtained by the developed tool is approximately similar compared to other tools.

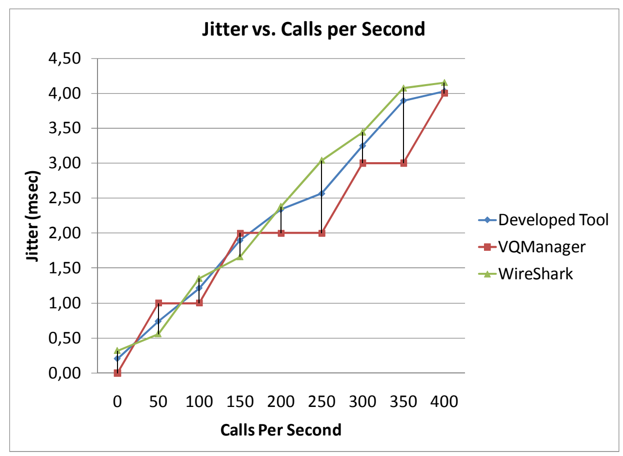

Fig. 7. Graph of jitter comparison

V. CONCLUSION AND FUTURE WORKS

A VoIP monitoring tool that is capable of measuring the delay, throughput, jitter and packet loss has been designed and implemented using Java and other related software. The proposed monitoring tool is tested on a testbed and the results were compared with VQManager and WireShark to verify that this proposed monitoring tool's results are acceptable. Based on the comparison, it is found that the results obtained by the developed tool are considered acceptable as the results seems to be comparatively similar compared to VQManager and WireShark. The difference might be due to the measurements which are not taken concurrently.

For future works, this developed monitoring tool can be improved by adding measurement for other traffics such as data and video. Moreover, a graph displaying the real-time measurement can be added to improve the visual appearance

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

of this GUI. A database can also be constructed using Structured Query Language (SQL) that will automatically save all the parameters measured and display it on a website.

REFERENCES

[1] CISCO's website, http://www.cisco.com/univercd/cc/td/doc/product/voice/sipsols/biggulp/bgsipov.htm Accessed on 3 October 2009 10:32PM (GMT +8)

[2] CISCO Support Forum's website, https://supportforums.cisco.com/docs/DOC-1422 : Accessed on 3 October 2009 11:00PM (GMT +8)

[3] Penn State University's website, http://citeseer.ist.psu.edu/183712.html : Accessed on 30 October 2009 4:32 PM (GMT +8)

[4] Filho, E.Z.; Fonseca, K.V.O., "Performance Monitoring of Individual VoIP flows on DiffServ Networks", Telecommunications Symposium, 2006, September 2006, pp. 370– 375.

[5] VQManager's website, http://www.manageengine.com/products/vqmanager/index.html : Accessed on 28 October 2009 1:17AM (GMT +8)

[6] WireShark's website, http://www.wireshark.org/ : Accessed on 29 October 2009 11:48PM (GMT +8)

[7] da Silva, J.M.; Lins, R.D., "Analyzing the QoS of VoIP on SIP in Java", Telecommunications Symposium, 2006, September 2006, pp. 576– 581.

[8] C. Lee, H. Kim, K. Ko, J. Kim and H. Jeong, "A Study On Structure for Monitoring and Detecting VoIP Abnormal Traffic," Future Generation Communication and Networking, 2008., vol. 2, December 2008, pp. 178–182.

[9] H.J. Kim and S.G. Choi, "Traffic Quality Monitoring System between Different Network Providers," The 12th International Conference on Advanced Communication Technology 2010 (ICACT2010), Vol. 2, 2010, pp. 1153 – 1158.

[10] Y. Kitatsuji. T. Hasegawa and H. Yokota, "Efficient Table Lookup Method for Performance Monitoring of VoIP Flows in Mobile Environment," IEEE Global Telecommunications Conference, 2008. (IEEE GLOBECOM 2008), 2008, pp.1 – 6.

[11] F. Fusco, F. Huici, L. Deri, S. Niccolini and T. Ewald, "Enabling High-Speed and Extensible Real-Time Communications Monitoring," IFIP/IEEE International Symposium on Integrated Network Management, 2009 (IM '09), 2009, pp. 343 – 350.

[12] P.C.Hershey, J.M. Pitts and R. Ogilvie, "Monitoring Real-Time Applications Events in Net-Sentric Enterprise Systems to Ensure High Quality of Experience," IEEE Military Communications Conference, 2009 (MILCOM 2009), 2009,pp. 1 – 7.

[13] Iperf's website, http://iperf.sourceforge.net : Accessed on 20 November 2009 10:21 AM (GMT +8)

[14] Znak's website, http://www.znak.fi/user/pdonner/g711tool : Accessed on 12 December 2009 8:32 AM (GMT +8)

[15] SIPp's website, http://sipp.sourceforge.net/ : Accessed on 28 October 2009 12:28AM (GMT +8)

[16] VoIP-Info's website, http://www.voip-info.org/wiki/view/ITU+G.711 : Accessed on 29 October 2009 2:31 PM (GMT +8)

[17] S. Agrawal, J. Ramamirtham and R. Rastogi, "Design of Active and Passive Probes for VoIP Service Quality Monitoring", Telecommunications Network Strategy and Planning Symposium, 2006, November 2006,pp. 1– 6.

ISBN: 978-988-17012-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)