Mathematical Approximation of Delay in Voice over IP

Mohammed M. Alani
Department of Computing,
Middle-East College
Muscat, Oman
Email: m {at} alani.me

Abstract— **This paper introduces a novel method of approximate calculation of delay in voice over IP systems. The proposed method relies on mathematical operations like interpolation and integration to produce a continuous function. This continuous function shows the delay in different times. The paper shows a detailed example of applying this method successfully to a sample call. The resulting delay function can help in better understanding of the variations of delay in different times and how it is affected by network load.**

***Keywords*- delay, ip, voice over ip, voip**

I. Introduction

Measuring delay in Voice over IP (VoIP) implementa-tions has been a key concern since the emergence of VoIP. Many components participate in this delay in such a way that makes it very difficult to have accurate measures.

The most basic method of measuring delay is to record the time of sending and time of receiving of voice pack-et. But, this method is not practical most of the time be-cause of the lack of proper clock synchronization mech-anisms.

Delay is identified in [1] as “the amount of time that a packet takes to travel from the sender's application to the receiver's destination application”. The components that compose this delay are:

1. compression and transmission delay at the sender.
2. propagation, processing and queuing delay in the network.
3. buffering and decompression delay at the receiver.

The first and third components are measurable to a great extent at the sender and receiver ends. However, in this paper, we are interested in the second component; the propagation, processing, and queuing delay in the network.

II. Literature Review

Both of the RTP standards; the old RFC1889 [2], and the new RFC3550 [3] did not include clear mechanisms of measuring delay.

In 2001, Cole and Rosenbluth introduced a comprehensive method of monitoring VoIP performance in [4]. This paper suggests introducing measurement points in the path of the packet to get accurate measurements and the paper recommends that more studies need to be carried out.

Also in 2001, Karam and Tobagi published a paper discussing delay and jitter in VoIP traffic. This paper focuses on suggesting proper scheduling scheme especially in low-bandwidth networks that carry VoIP data [5].

Most studies that came after 2003, relied heavily on the ITU-T recommendation G.107, [6], that introduced the E-Model. The E-Model provided a subjective quality score named Mean Opinion-Score (MOS) which can be identified as the metric of perceived voice quality.

ČÁKY et al. published a paper;[7] in 2006 discussing end-to-end VoIP quality measurement. This paper relied on voice recording of samples to measure VoIP delay. The paper focused more on measuring the effects of packet loss on the voice signal.

Ngamwongwattana and Thompson introduced in 2009 a method for measuring one-way delay of VoIP packets without clock synchronization in [8]. This method focused on using differences in packets arrival times in synchronize the timing of both ends.

In 2010, Makoto Aoki et al. published a paper that discusses measurement scheme for one-way delay variation in [9]. This paper works on detecting and removing clock skew and measuring all the variables that participate in the one-way delay.

Many other papers relied on the ITU-T recommendation in [5] in various implementations. Such papers are [10], [11], and [12].

III. Proposed Method

At the start, few quantities are to be identified;

time of receiving packet number i

time of sending packet number i

is the time delay of packet numbered i to arrive from source to destination

is the difference between delay of two consecutive packets; packet i+1 and packet i

Now, we will identify the value of

$d_i = r_i - s_i$

For the sake of simplicity, we will assume that all the time is measured according to the receiver time. This assumption is valid as we later will use differences between time values and will not use absolute values of time.

Also,

$d_{i+1} = r_{i+1} - s_{i+1}$

Now,

$\delta_{i+1} = d_{i+1} - d_i$

$\delta_{i+1} = (r_{i+1} - s_{i+1}) - (r_i - s_i)$

Extracting the brackets

$\delta_{i+1} = r_{i+1} - s_{i+1} - r_i + s_i$

$\delta_{i+1} = r_{i+1} - r_i - s_{i+1} + s_i$

Creating new brackets,

$$\delta_{i+1} = (r_{i+1} - r_i \qquad (1)$$

The two quantities; , and are measurable quantities. The difference in sending time can easily be measured at the sender end, and the difference in receiving time can be easily measured at the receiving end. It is also possible to measure the difference in sending time at the receiver end by measuring the differences between timestamps of RTP messages. According to the standard of RTP, [3], "The timestamp reflects the sampling instant of the first octet in the RTP data packet." Thus, the difference between timestamps can be multiplied by the sampling rate of the CODEC being used to get the actual time difference.

After collecting the values of adequate number of samples of the RTP stream for , and , these collected values can be used to construct a certain equation of δ as a function of time using a proper interpolation method.

$$\delta(t) = a_n t^n + a_{n-1} t^{n-1} + \cdots + \qquad (2)$$

The choice of *n*, the power of the polynomial is based on the required accuracy and the number of sample packets taken. Using a large number of sample packets can lead to a case of over-interpolation.

It is sound to say that can be thought of as the derivative of delay function. As the physical meaning of the difference between consecutive delays is the derivative of delay function. Hence, we will produce an equation to find delay as a function of time; though

(3)

This would produce an equation similar to this,

$$d(t) = b_m t^m + b_{m-1} t^{m-1} + \cdots + b \qquad (4)$$

The values of to can be found easily during the integration process. It is the value of that will require extra work.

To calculate accurately, we will need a known value of at a certain value of . It might seem inviting to assume and hence . But, this assumption will render the whole process unusable because it will cause packet groups of different delay but similar fluctuation in delay to give the same delay values calculated by Equation 3. Hence, this assumption is not valid.

One way of getting an acceptable approximation is to measure the Round-Trip Time (RTT) for a ping packet sent and received during the time between the first sample and the last sample.

IV. Example of the Proposed System

To check the applicability of the proposed method we have implemented it using measurements taken for a VoIP call made between two computers using VoIP software. The sample space taken was about 3,000 RTP packets.

A. *Implementation Environment*

Software used to make the test VoIP calls was Ekiga [13]; an open-source VoIP client software that supports H.323. To capture the RTP voice packets, WireShark [14] software was used. For the interpolation calculations MATLAB R2008a [15] was used. The CODEC used in the call was G.711 because it has a constant bit rate that is high enough to have a lot of samples in a short period of time [16].

B. *Chosen Samples*

The number of packets transmitted during the call from the sender to the receiver was about 3,000 RTP packets. To study the overall approximate behavior of delay, 10 samples were chosen from the sequence. These samples were separated by equal number of packets, i.e., we chose the 1st, 300th, 600th, 900th, …, 3000th packets.

C. *Calculations*

Table I shows the values of , , and for the 10 sample packets taken from the sample call packets.

TABLE I. Calculated Timings of the Chosen Samples

Sequence Number	**(s)**	**(s)**	**(ms)**	**Receiving Time (s)**
28050	0	0	0	14.192618
28350	8.780169658	8.789	-0.008830342	22.97278766
28650	8.790258890	8.789	0.001258890	31.76304655
28950	8.788271618	8.789	-0.000728382	40.55131817
29250	8.788883848	8.789	-0.000116152	49.34020201
29550	8.788685496	8.789	-0.000314504	58.12888751
29850	8.788427377	8.789	-0.000572623	66.91731489
30150	8.789088227	8.789	0.0000882269	75.70640311
30450	8.789275959	8.789	0.000275959	84.49567907
30750	8.789873258	8.789	0.000873258	93.28555233
31050	8.790229064	8.789	0.001229064	102.0757814

The receiving time in Table 1 is with the reference to the receiver time counting from the start of the call.

After getting 10 values of , now we will use these values to create a polynomial similar to Equation 2. For this purpose, a MATLAB function called '`polyfit`' was used with the receiving time as input and as output, to generate a polynomial of power 8. The result was the following polynomial coefficients:

$$-15.05061 \times 10^{-15} \quad 78.17648 \times 10^{-13} \quad -17.3677 \times 10^{-10} \quad 21.$$
$$\times 10^{-4} \quad -21.63609 \times 10^{-3} \quad 0.3401 \quad -2.2576$$

Thus, the resulting polynomial is:

$$\delta(t) = -15.05061 \times 10^{-15}t^8 + 78.17648 \times 10^{-13}t^7 - 17.3677 \times 10^{-10}t^6 + 21.51534 \times 10^{-8}t^5 - 16.2242 \times 10^{-6}t^4 + 7.61008 \times 10^{-4}t^3 - 21.63609 \times 10^{-3}t^2 + 0.3401t - 2.25764 \quad (5)$$

To check the accuracy of this interpolation, the values of was calculated from the resulting polynomial and compared to the measured values. Table II, and Figure I show the values of calculated and measured .

TABLE II. MEASURED AND CALCULATED VALUES

Receive Time (s)	**Measured $\delta(t)$ (s)**	**Calculated $\delta(t)$ (s)**	**Error (*10^{-12})**
22.97278766	-0.008830342	-0.008830342	0.18207
31.76304655	0.001258890	0.001258890	-0.06084
40.55131817	-0.000728382	-0.000728382	0.026645
49.34020201	-0.000116152	-0.000116152	0.003552
58.12888751	-0.000314504	-0.000314504	0.014654
66.91731489	-0.000572623	-0.000572622	-0.00088
75.70640311	0.0000882269	0.0000882268	-0.05329
84.49567907	0.000275959	0.0002759592	-0.06572
93.28555233	0.000873258	0.0008732584	-0.06740
102.0757814	0.001229064	0.0012290644	-0.25550

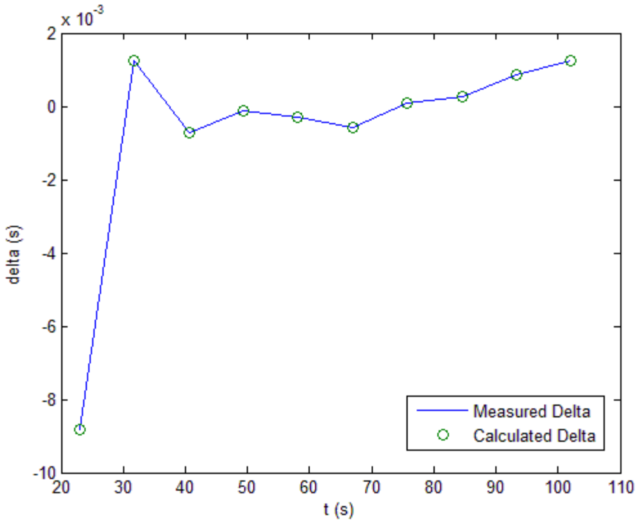

Figure 1. Calculated and Measured Values of δ(t)

Now that we have as a function of time, we will proceed with the procedure by finding the delay as a function of time using Equation 3.

$$d(t) = \int \delta(t).dt$$

$$d(t) = \int -15.05061 \times 10^{-15}t^8 + 78.17648 \times 10^{-13}t^7 - 17.36$$
$$+ 7.61008 \times 10^{-4}t^3 - 21.63609 \times 10^{-3}t^2 + 0.$$

$$d(t) = \frac{-15.05061 \times 10^{-15}t^9}{9} + \frac{78.17648 \times 10^{-13}t^8}{8} - \frac{17.3677 \times 1}{7}$$
$$+ \frac{7.61008 \times 10^{-4}t^4}{4} - \frac{21.63609 \times 10^{-3}t^3}{3} + \frac{0.340}{2}$$

$$d(t) = -1.67229 \times 10^{-15}t^9 + 9.77206 \times 10^{-13}t^8 - 2.48110 \times 10^{-10}t^7 + 3.58589 \times 10^{-8}t^6 - 3.24484 \times 10^{-6}t^5 + 1.90252 \times 10^{-4}t^4 - 7.21203 \times 10^{-3}t^3 + 0.17005t^2 - 2.25764t + b_0 \quad (6)$$

To calculate an ICMP echo request was sent from the sender of the RTP packets to the receiver to calculate the average time required for the packet to arrive. To provide better accuracy, the ICMP request and reply time was calculated using WireShark's packet capturing software. The round-trip time for the ICMP message was 415.196 ms. With 35.421s, and 35.626s in receiver time as the sending and receiving time. By dividing the round-trip time over two, we can get a good estimate of the time delay.

$$d(t_{ICMP}) = \frac{415.196}{2} = 207.59\ ms$$

$$t_{ICMP} = \frac{send\ time + receive\ time}{2} = \frac{35.421 + 35.626}{2} = 35.5235s$$

Apply these values to Equation 6;

$$0.20759 = -1.67229 \times 10^{-15}(35.5235)^9 + 9.77206 \times 10^{-13}(35.52$$
$$\times 10^{-8}(35.5235)^6 - 3.24484 \times 10^{-6}(35.5235)^5 +$$
$$\times 10^{-3}(35.5235)^3 + 0.17005(35.5235)^2 - 2.2576$$

$$0.20759 = b_0 - 12.82197$$

$$b_0 = 13.02956$$

Hence,

$$d(t) = -1.67229 \times 10^{-15}t^9 + 9.77206 \times 10^{-13}t^8 - 2.48110 \times 10^{-}$$
$$+ 1.90252 \times 10^{-4}t^4 - 7.21203 \times 10^{-3}t^3 + 0.1700$$

Figure 2 shows a sample from the calculated delay based on the proposed procedure. The figure shows the behavior of the delay in the period between 23s and 100s.

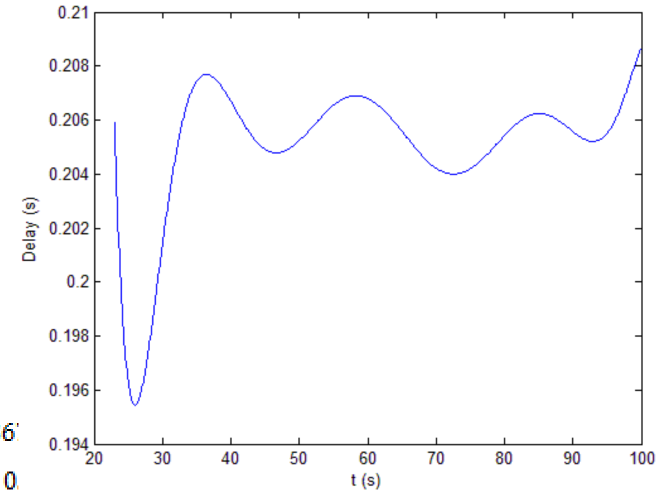

Figure 2. Sample Delay Values Calculated Using the Proposed Procedure

V. Discussion

The continuous function resulting from the proposed procedure gives a god approximation of the delay values and how it changes with time. During the process of producing this function, few comments need to be made about each stage of the process.

A. Samples Selection

The total number of samples used in the suggested procedure depends on the desired accuracy. However, choosing a lot of samples will result in either a polynomial with very high order that when integrated gives physically unrealizable quantities, or a low-order polynomial that does not reflect the required values of all sample points. The interval between those samples can be chosen based on the period that is under study.

B. Interpolation

It is possible to generate a polynomial that represents all the sample points taken through MATLAB easily. However, if the fluctuation in the values of is high, a higher-order polynomial will be needed. Higher-order polynomials can result in integrals that give unrealizable values (like negative delay values, for example).

In the solved example, the order was chosen such that the order of the resulting integral would equal the number of samples-1, i.e. the order used in the interpolation process is the number of samples -2.

In some cases, polynomials do not provide acceptable interpolation to the sample points. In these cases, the use of other methods, such as Spline, or rational interpolation is suggested.

C. Resulting Delay Function

The resulting polynomial, after integration, gives a good estimation of the delay the voice packets are spending to get from the source to the destination. This continuous function can be used in studying the impact of different factors on the quality of VoIP traffic.

D. Implementability

The suggested method can be applied to RTP traffic whether it was carrying audio or video. It is independent of the type of network, signaling protocol, and client software used. It is a passive measurement method that does not interfere with the call traffic.

VI. Conclusions

This paper presented a novel method to calculate approximate values of delay in VoIP traffic carried through RTP packets. The suggested procedure employs interpolation and integration to produce a continuous function that represents delay. This continuous function can give a very good approximation to the values of VoIP delay which can be helpful in understanding the network behavior when changes in traffic happen.

The suggested procedure can be applied to VoIP or video traffic, and is network-type-independent, signaling-protocol-independent, and client-software independent. This means that this research can be taken further through implementing it on video traffic and more research can be done to improve the accuracy of the proposed system.

The procedure was implemented successfully to a sample call and the result was a function that shows the values of delay in any time during the call.

References

[1] Calyam et al., Performance Measurement and Analysis of H.323 Traffic, Proceedings of the 5th International Workshop on Passive and Active Network Measurement, Paris, France, April, 2004.

[2] IETF RFC1889, http://www.ietf.org/rfc/rfc1889.txt .

[3] IETF RFC3350, http://tools.ietf.org/html/rfc3550 .

[4] R. G. Cole and J. H. Rosenbluth. 2001. Voice over IP performance monitoring. **SIGCOMM Comput. Commun. Rev.,** Vol. 31, No. 2, April 2001, pp. 9-24.

[5] Karam, M.J.; Tobagi, F.A., "Analysis of the delay and jitter of voice traffic over the Internet," *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* , vol.2, pp.824-833, 2001.

[6] ITU-T Recommendataion G.107. The E-Model, a computational model for use in transmission planning, Mar 2003.

[7] ČÁKY et al., End-To-End VOIP Quality Measurement, Acta Electrotechnica et Informatica, Vol. 6, No. 1, 2006, pp.1-5.

[8] Ngamwongwattana and Thompson, Measuring One-Way Delay of VoIP Packets Without

Clock Synchronization, Proceedings of International Instrumentation and Measurement Technology Conference, Singapore, 5-7 May 2009.

[9] Makoto Aoki et al., Measurement Scheme for One-Way Delay Variation with Detection and Removal of Clock Skew, ETRI Journal, Volume 32, Number 6, 2010, pp. 854-862.

[10] H. Xie and Y. Yang, A Measurement based Study of the Skype Peer-to-Peer VoIP Performance, Prceedings of the 6th International Workshop on Peer-to-Peer Systems, USA, Feb. 2007.

[11] B. Sat and B. Wah. 2007, Playout scheduling and loss-concealments in voip for optimizing conversational voice communication quality. In **Proceedings of the 15th international conference on Multimedia** (MULTIMEDIA '07), ACM, NY, USA, pp. 137-146.

[12] Narasimha Raghavan Veeraragavan, Andrea Bondavalli, Leonardo Montecchi, Roman Vitenberg, Nicola Nostro, and Hein Meling. 2013. Understanding the quality of experience in modern distributed interactive multimedia applications in presence of failures: metrics and analysis. In **Proceedings of the 28th Annual ACM Symposium on Applied Computing** (SAC '13). ACM, NY, USA, pp. 439-446.

[13] http://www.ekiga.org

[14] http://www.wireshark.org/

[15] http://www.mathworks.com/products/matlab/

[16] IETF RFC5391 , G.711. http://tools.ietf.org/html/rfc5391