Voice Traffic Performance Measurement in Packet Networks

V. Matić, A. Bažant and M. Kos
University of Zagreb
Faculty of Electrical Engineering and Computing
Department of Telecommunications
email{viktor@tel.fer.hr}

Abstract. *Quality of voice transportation over packet networks depends on three critical parameters: packet loss, absolute packet delay and packet delay variation (jitter). Common practical problem is how to estimate voice quality for a different number of voice sources with different source characteristics in target networks. Important characteristics of a voice source are: rate (packets per second), packet length (depending on a type of compression), type of source (VBR or CBR) and statistical description. In this paper we propose methodology and practical approach for systematic measurement of voice traffic parameters in respect to varying voice source characteristics. Measurement results that are conducted with presented methodology on an experimental DiffServ network are given, too.*

Keywords. VoIP, packet loss, jitter, delay, DiffServ, voice quality.

1. Introduction

Recently, mechanisms that offer guaranteed QoS penetrate in existing packet networks. Network providers, no matter what kind of technology they use, MPLS, DiffServ, IntServ, ATM or other, have to measure traffic performance and to estimate service quality they can offer to customers [2]. Any customer who wants to verify quality of service agreed with network provider can conduct similar measurements. But, methodology and approach to measurements for these two cases are different. Customer is aware of services he is subscribed for. He also knows traffic characteristics that are necessary for the achievement of the agreed QoS. Hence, measurements can be done for a reduced set of traffic parameters. On the other hand, network service provider has to measure wider range of traffic parameters because of a variety of offered services.

It is important to correctly map traffic parameters with a desired service. When one defines service quality, he actually puts some constraints on packet rate, jitter, packet loss, packet delay, etc. Network is allowed to apply the constrained parameters to a specific traffic type. It is hard to determine voice quality on the basis of measured traffic parameters because of a subjective nature of speech quality estimation. Common practice allows us to use empirical values as shown in table 1.

Table 1: Voice quality estimation

Quality	Packet loss [%]	Maximum delay variation [ms]
excellent	0	0
good	3	75
average	10	125
bad	25	225

The next step is to describe services by means of corresponding traffic parameters and then, to design sources that can generate packet streams with characteristics as close as possible to real sources. After that, it is necessary to design voice traffic sinks. Sinks are not less important then sources because they actually measure traffice performance. What is left to be done is to compare output results of traffic characteristics measurements with service requirements. Based on the comparison it is important to decide which services could be supported by network.

2. Voice Source Emulation

Emulated sources and sinks are necessary for measurements of voice traffic parameters in a real network. Source emulators should be configurable through following parameters: number of sources, packet length, packet rate (number of packets per second), type of service (VBR or CBR). If a source is VBR, then its statistical parameters are needed, too.

Packet length is equivalent to corresponding voice compression algorithms used to code hypothetical voice stream. Packets sent into a network by emulator are not actually empty, even if they do not carry voice information. Real voice transport is based on real time protocol, such as RTP [4], that carries not only voice, but also the information like packet sequence number, time stamp (time of packet creation) and other information important for real time delivery. It is necessary to embed that kind of information in each packet because of the emulation purposes. Sequence number and time stamp are especially important because these parameters are used on receiver side as an entry to a packet loss and jitter calculation.

While it is easy to describe CBR source by means of packet length and fixed inter-packet time, VBR source is much more complicated. VBR traffic is generated during a voice communication due to the implementation of silence suppression algorithms in a terminal equipment. While the source is silent it does not generate packets. This is so called On/Off source, and it can be described with Markov chain model. The full Markov model of a voice conversation comprises six states [3]. This model is too complex for implementation and practical use. The two state model can be derived from the full model by a series of simplifications. The new model is good enough for measurement purposes.

3. Two States Markov Chain Model of a Voice Source

As it is shown in figure 1, this model consists of two stable states.

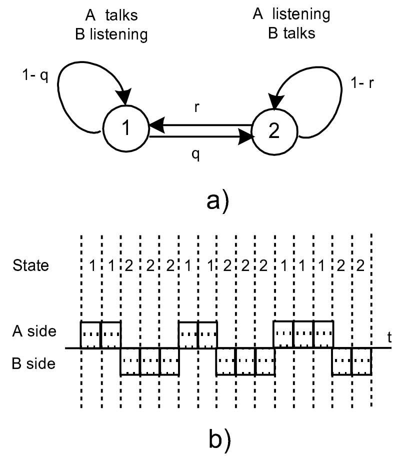

Figure 1: Markov chain with two states

This model is a simplification of the full conversation model. It can not describe states such as: both sides are silent or both sides are talking. After entering state 1 emulation application starts to send packets into a network, and, as long as it stays in that state, it continues to send equally spaced packets. Because of a discrete nature of this process, next state is calculated after a packet is sent.

This process can be described with a transition probability matrix,

$$\mathbf{P} = \begin{bmatrix} 1-q & q \\ r & 1-r \end{bmatrix}. \qquad (1)$$

If the process is stationary [6] it can be written:

$$\mathbf{pP}=\mathbf{p} \qquad (2)$$

where $\mathbf{p} = [\hat{P}_1 \ \hat{P}_2]$. From the matrix equation (2) we can derive stationary probabilities as it is shown in (3)

$$\hat{P}_1 = \frac{r}{r+q}, \ \hat{P}_2 = \frac{q}{r+q}. \qquad (3)$$

A problem is how to choose transitional probabilities to satisfy desired mean time spent in one of the states (T_1, T_2), while still retaining fixed stationary probabilities. In other words, there is an infinite number of r and q combinations satisfying the equation (3). It is not only important to identify stationary probabilities for a voice conversation. It is also important to determine how long process continually remains in each of the states. That is a reason way constraints are put on the mean times (talking or listening). Exact values of the time intervals are gained from statistical measurements on real conversations [5].

If the first packet of the talking interval has been generated, we can ask ourselves: "What is a probability that i packets will be generated in a sequence?". This could be written as

$$p(i) = (1-q)^{i-1}q. \tag{4}$$

Now we can evaluate mean value for this stochastic process as shown in the equation (5). Notice here that the number of packets received in a sequence conforms to the geometric distribution

$$E[X] = \sum_{i=0}^{\infty} ip(i) = q\sum_{i=0}^{\infty} i(1-q)^{i-1}. \tag{5}$$

The equation (5) can be written in other form:

$$E[X] = q\frac{\partial}{\partial(1-q)}\sum_{i=0}^{\infty}(1-q)^{i}. \tag{6}$$

It could be easily seen that expression (6) represents geometrical series. Hence, the result of summation is $1/\left[1-(1-q)\right]$. Finally, the result is

$$E[X] = q\frac{\partial}{\partial(1-q)}\left[\frac{1}{[1-(1-q)]}\right] = \frac{q}{[1-(1-q)]^2} = \frac{1}{q}. \tag{7}$$

Now when we know the expected number of generated packets in a talk interval, we can also evaluate mean time spent in state 1. Since packets in the talk interval are sent equally spaced by Δ ms, we can write

$$T_1 = \Delta E[X] = \Delta\frac{1}{q} \tag{8}$$

$$T_2 = \Delta E[Y] = \Delta\frac{1}{r} \tag{9}$$

where T_1 and T_2 are mean times spent in state 1 and state 2, respectively.

If we express transitional probabilities q and r through expected time T_1 and T_2, we can write,

$$q = \frac{\Delta}{T_1},\ \ r = \frac{\Delta}{T_2} \Rightarrow$$

$$\hat{P}_1 = \frac{\Delta}{\left(\frac{\Delta}{T_1}+\frac{\Delta}{T_2}\right)T_2},\ \ \hat{P}_2 = \frac{\Delta}{\left(\frac{\Delta}{T_1}+\frac{\Delta}{T_2}\right)T_1} \Rightarrow$$

$$\hat{P}_1 = \frac{T_1}{T_1+T_2},\ \ \hat{P}_2 = \frac{T_2}{T_1+T_2} \tag{10}$$

It can be seen that both assumptions are met (10): stationary probabilities are in the given ratio to each other (e.g. 40%-60%), and expected times T_1 and T_2 are equal to predefined values.

3.1. Model Generalization for N Sources

In the case of multiple voice sources it is necessary to introduce a general model derived from the basic two state Markov model. Thus, it is observed that the steady-state arrival process is a binomial process. This is clearly due to the fact that each speaker's behavior is an independent Bernoulli trial, with probability of supplying a packet equal to $r/(r+q)$ [3]. The question is: "How to express the probability that in k-th time interval (interval duration is fixed to Δ ms) n of m sources are active?". For steady-state, or in other words, when $k \to \infty$ one can write

$$P_n = \begin{pmatrix} m \\ n \end{pmatrix}\left(\frac{r}{r+q}\right)^{n}\left(\frac{q}{r+q}\right)^{(m-n)}. \tag{11}$$

In practical application of this generalization there is a problem how to calculate the equation

$$\begin{pmatrix} m \\ n \end{pmatrix} = \frac{m!}{n!(m-n)!}; \quad m\in N,\ n\in N, m\geq n \tag{12}$$

in standard C programming language implementation. Largest number which can be represented with standard C library is 170!.

Hence, that number of channels is insufficient for extensive voice source emulation. On the other side, network interface capacity and computer processing power should be the only practical limitations. For example, personal computer with Pentium processor and 10 Mbit/s Ethernet network card can generate up to 500 voice channels.

Solution of the problem is the approximation of binomial with normal distribution (formal proof that this could be done is given in [6]):

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-a)^2}{2\sigma^2}}. \tag{13}$$

The use of normal distribution is preceded with parameter mappings as shown in

$$\begin{aligned} a &= m\hat{P}_1 \\ \sigma^2 &= m\hat{P}_1\hat{P}_2 \end{aligned} \tag{14}$$

where a is mean and σ is standard deviation. Expectation of normal distribution is a product of number of users and stationary probability $\hat{P}_1$. This product represents expected number of active voice channels in k-th time interval. Let's say that silence to talk ratio is 60%-40%, and maximum number of users is 1000. Then the expected number of active users in k-th time interval is 600, what is a logical result.

We can write distribution function expressed through error function as shown in

$$\begin{aligned} F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{(t-a)^2}{2\sigma^2}} dt = \\ = \frac{1}{2}\left[1 - erf\left(\frac{a-x}{\sqrt{2}\sigma}\right)\right]. \end{aligned} \tag{15}$$

Using expression

$$erfc(z) + erf(z) = 1 \tag{16}$$

distribution function could be expressed through complementary error function

$$F(x) = \frac{1}{2} erfc\left(\frac{a-x}{\sqrt{2}\sigma}\right). \tag{17}$$

Finally, the inverse function of (17) is given by

$$x = a - \sqrt{2}\sigma[erfc^{-1}(2v)] \tag{18}$$

where $F(x) \equiv v$. If we use v as a random generated number conforming to uniform distribution, then variable x conforms to normal distribution. Almost all operating systems and programming languages support uniform distribution through `rand` function. Hence, practical implementation of expression (18) is trivial.

The exact measure of error introduced with this approximation doesn't exist [6]. But the approximate estimation can still be done by observing figure 2 that shows both distributions (binomial and normal) on the same graph. Each graph has two parameters: m as a maximum number of users, and p as a stationary probability of silence.

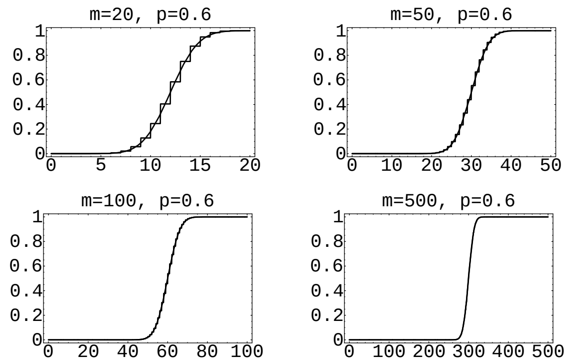

Figure 2: Approximation of binomial with normal distribution

As it can be seen, approximation works fine for the number of users greater then 50.

4. Implementation

We have implemented voice source emulator in C programming language. The idea behind a practical implementation is to use two independent emulators in the same application. The first emulator works on one UDP port and it emulates N voice sources in accordance with equation (18). The second emulator works on the other UDP port and it emulates only one voice source but with same statistical parameters as the first emulator. Channel generated by the second emulator has special purpose and we will refer to it in the rest of the article as the measurement channel. The measurement channel carries information for real time delivery and measurements like sequence number, time stamp, number of emulated sources, etc.

Sink, which is in fact the same application as source, but with different command line switches, performs measurements only on the measurement channel. Measurements results are generalized to the rest of generated channels. This approach is possible because we can say that the measurement channel is randomly selected from an aggregated voice traffic flow. This channel has the same statistical parameters as the rest of generated channels, and accordingly, it can represent all of them. One could ask why to bother with the rest of channels at all. The answer is simple: "Because we need to load network with traffic similar to real voice transport to achieve authentic measurements".

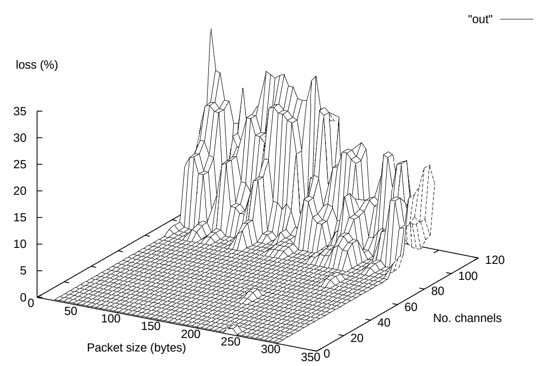

Figure 3: Loss measurements on DiffServ node.

Further, emulator application has a feature to automatically change the maximum number of users and packet length. For example, the application can be configured to generate 100 packets for each combination of packet length and number of users, where both parameters are varied within ranges defined at the application startup. Thanks to this approach, measurement results can be presented in three dimensional space, where X is packet length, Y is number of active channels and Z is measured value (for example packet loss or jitter). This graphical representation gives better insight in phenomena and makes easier to identify the level of service that can be offered to a customer. Example of such representation for measurements on DiffServ node is shown on figure 3.

5. Further Work and Conclusion

Currently, the same packet lengths are used for all emulated channels. The future development plan is to implement support for varying packet lengths. In practice we could expect that customers use different compression algorithms. As a consequence, traffic sources generate packets of varying lengths. It has an influence on queuing disciplines in network nodes, what is directly connected to service quality. Hence, emulation would be more precise if we could define mapping between packet length and specific emulated channel.

References

[1] Željko Pauše. *Vjerojatnost, informacija stohastički procesi*. Školska knjiga Zagreb, 1974.

[2] Nobuhiko Kitawaki and Kenzo Itoh. Pure Delay Effects on Speech Quality in Telecommunications. *IEEE Journal on Selected Areas In Communications*, 9(4):[586]–[593], May 1991.

[3] Daniel Minoli and Emma Minoli. *Delivering voice over IP networks*. John Wiley and Sons,Inc., 1998.

[4] H. Schulzrinne. RTP: A Transport Protocol for Real-Time Applications, Jan 1996. Status: STANDARD.

[5] PAUL T.BRADY. A Statistical Analysis of On-Off Patterns in 16 Conversations. *THE BELL SYSTEM TEHNICAL JOURNAL*, 1968.

[6] Dimitrije Ugrin-Šparac. *Primjenjena teorija vjerovatnosti*. Sveučilišna naklada, 1990.