

An Inter-arrival Delay Jitter Model using Multi-Structure Network Delay Characteristics for Packet Networks

Edward J. Daniel, Christopher M. White, and Keith A. Teague

School of Electrical and Computer Engineering
Oklahoma State University
Stillwater, OK

ABSTRACT

Data traversing packet networks experience varying delays, resulting in inter-arrival jitter. This can result in degraded performance in real-time multimedia communications applications if the jitter delays are large or unaccounted for in the receiver application. This paper examines modeling and simulation of network jitter delay for real-time multimedia communications applications. We examine the multi-structure characteristics of network delay and develop a model for simulation of jitter. The model is confirmed empirically using collected packet network jitter delay statistics.

1. INTRODUCTION

Packet-switched networks are being used to provide interactive multimedia communications, including real-time voice, video and data services. However, packet networks are not ideal for real-time multimedia applications such as voice communications. Packet networks exhibit non-ideal behavior that may seriously degrade the performance of real-time communications systems. Particularly, as packets are transmitted from source to destination, they may experience different delays. As a result, packets arrive at the destination with varying delays (between packets) referred to as 'jitter'.

Packet delay jitter may result from packets taking different paths to their destination to avoid congested areas or failed links. However, jitter is primarily caused by varying queuing delays encountered by packets at routers (nodes). Network packets compete with other networks traffic at routers. Routers statistically multiplex incoming packets, which results in the varying delay.

In a voice over IP (VoIP) application, large inter-arrival jitter leads to starvation of the audio playback system. If there is not adequate buffering on the received system these delays will lead to packet loss and a corresponding loss of voice data (seen at playback). Data

buffering to overcome this problem can lead to excessive end to end delay in the system.

In this paper, we examine the characteristics of network delay jitter to develop a model for network inter-arrival jitter. Section 2 describes the characteristics of packet network inter-arrival delay. Section 3 presents our network delay jitter model. Section 4 gives comparisons of our model to empirically collected network jitter data. Finally, in Section 5 summary and conclusions are given.

2. NETWORK JITTER DELAY PROPERTIES

When a real-time voice communication application uses a wide area packet network employing IP (Internet Protocol) to transmit a stream of voice data, referred to as a *tagged stream*, it is multiplexed at network nodes (routers) with the other traffic on the network called background traffic. IP network nodes have no priorities, so network traffic is serviced on a First in First out (FIFO) basis. Therefore, queuing delays and jitter experienced by packets are a direct result of wait and service times in the queue. Jitter delays of tagged streams on packet networks are a sample of the queuing delays of the background traffic at the nodes [1] [2]. The relative size of the tagged stream is small compared to the background traffic, so it does not significantly contribute to the queuing delays. As tagged stream packets travel across a network path they sample the busy periods¹ (and queuing delays) of the nodes along the path [3]. If the send rate, or the time between packets, of the tagged stream is smaller than the busy periods of the node, the tagged stream gets a sample of the queuing delays within the busy periods (more than one sample from the same busy period), Figure 1(a). However, when the send rate is larger than the busy

¹ When several packets arrive at the node during a given interval (possibly simultaneously), it is referred to as a busy period.

periods, the tagged stream samples delays of different busy periods (many different busy periods), Figure 1(b). Essentially, larger send rates sample the busy periods while smaller send rates sample the delays within the busy periods.

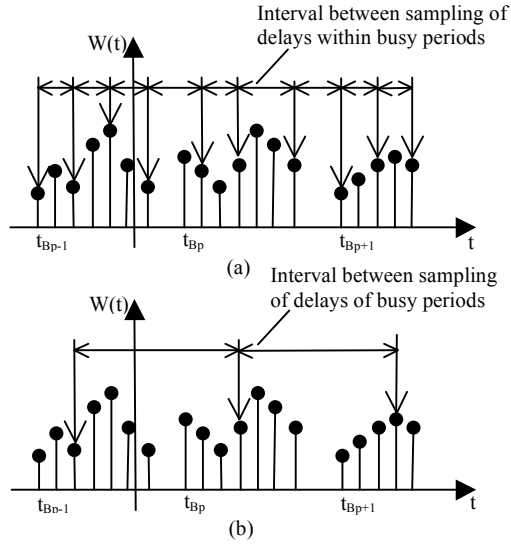


Figure 1. (a) Tagged stream sampling of queuing delay within busy periods and (b) Tagged stream sampling of busy periods of a queue. $W(t)$ is an instance of a queuing delay process and t_{Bp} represents the start time of a busy period. [3]

According to Li and Mills [3], this sampling describes the multi-structure of delay, which consist of a short term non-stationary component, the delays within busy periods, and a long-term stationary component, the delays of different busy periods. The long-term stationary component can be modeled as a Gaussian noise process because there is no correlation between the occurrences of busy periods of a queue. Moreover, the delays sampled within the queue for each busy period have no correlation with delays sampled from other busy periods. The multi-structure characteristic of network delay is used in the next section to develop a model for jitter.

3. NETWORK JITTER DELAY MODEL

Li and Mills' ([4]) study and analysis of network delay states that smaller sampling intervals can be modeled as a non-stationary process, while the longer time intervals can be modeled as a stationary process. The point at which the delay changes from its small time scale

characteristic to its large time scale characteristic is referred to as the crossover.

Several researchers have developed complex models and performed empirical simulation studies of network jitter including [1] [2] [5]. These studies show that network jitter follows a Laplacian distribution. Yletyinen and Kantola ([5]) showed that the probability distribution function widens and shortens based on increases in traffic load, utilization, burstiness, and burst length. Widening of the Laplacian pdf signifies higher probabilities of long jitter delays, while shortening of the pdf implies higher probability for small jitter delays.

While investigating an adaptive jitter buffering algorithm, Ramjee *et al.* [6] reported the delay spike phenomena, a condition where a delay spike in a packet stream is preceded by a series of packets arriving very close together. When a packet experiences a large delay, the packets following get “held up” and as a result they all arrive with very small inter-arrival times, sometimes almost simultaneously. Ramjee *et al.* found that the delay spike phenomena decayed exponentially, and incorporated an exponential decaying function into the spike detection jitter buffering algorithm. When his jitter buffering algorithm detected a delay spike it was allowed to decay exponentially to follow the slope of the spike until it leveled off at approximately zero slope, signaling the end of the delay spike. This obviously indicates a correlation attribute of inter-arrival jitter delays.

In our proposed jitter model, delays are generated from probability distributions that have no correlation between samples, thus not simulating delay spike characteristics. Therefore, an exponential decaying function is used to simulate this phenomenon when the Laplacian or Gaussian distribution produces a delay spike.

The Laplacian distribution is used to model the small time scale, non-stationary, process, and the large time scale stationary process is modeled with additive Gaussian white noise. Jitter delay spikes and subsequent jitter delays follow the delay spike adjustment procedure noted above. The combination of the Laplacian distribution, Gaussian white noise, and delay spike adjustment provide a means to model the multi-structure characteristics of packet delay. The proposed model is shown in figure 2.

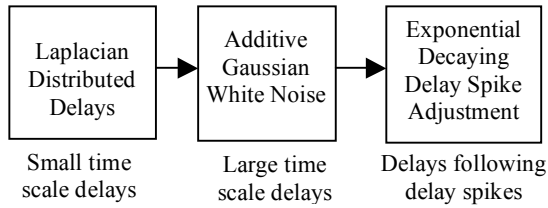


Figure 2. Proposed Jitter Model

The Laplacian distribution is used to generate packet jitter delay values based on a specified mean and variance. The characteristic equations (1) and (2) for the Laplacian distribution function are given as follows:

$$F(X) = \frac{1}{2} e^{-\frac{|X-\alpha|}{\beta}}, \quad x \leq \alpha, \quad (1)$$

$$F(X) = 1 - \frac{1}{2} e^{-\frac{|X-\alpha|}{\beta}}, \quad x > \alpha, \quad (2)$$

where,

$$\alpha = \text{mean} \quad \text{and} \quad \beta = \sqrt{\frac{\text{var}}{2}}.$$

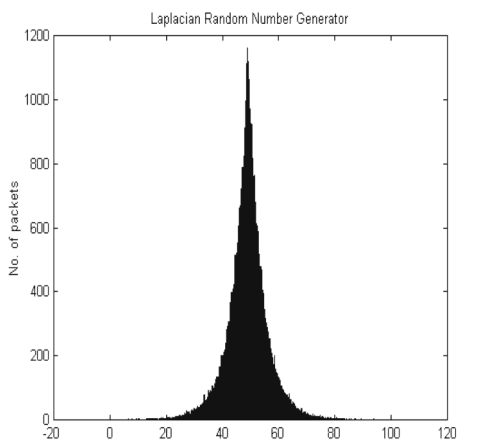


Figure 3. Generated Laplacian Distribution using Laplacian Random Number Generator.

Figure 3 shows the Laplacian distribution function implementation simulating 10,000 packets with $\alpha = 49$ and $\beta = 5$.

Gaussian white noise is added to the delay values generated by the Laplacian distribution function to introduce the large time scale properties. This is done based on a sampling that follows the crossover value. For example, delay values are altered by the Gaussian white noise process at intervals based on the crossover from small time scale to large time scale. This process models sampling of the busy periods of the queue.

When a delay spike is detected, based on a threshold, delay values following the delay spike are reduced exponentially, allowing them to “catch up” with the delayed (spike) packet. Equation (3) gives the exponential decay function used to model this characteristic. The number of proceeding packets that are affected by the delay spike is denoted as the *spike interval*. In order for the packets following the spike to “catch up” with the delay spike the average jitter delays within a spike interval should be less than or equal to the mean.

$$\tilde{d}(t) = d(t)e^{-xt}, \quad (3)$$

where, $d(t)$ is the inter-arrival jitter at time t . Figure 4 provides a pictorial description of this delay jitter adjustment process.

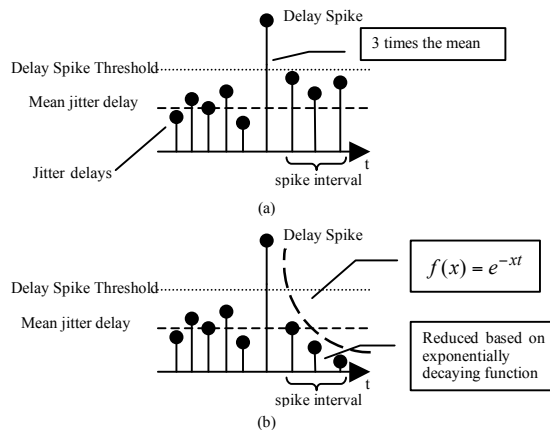


Figure 4. Jitter delay spike adjustment procedure. (a) delay spike produced from model and (b) reduction of delays following delay spike.

4. JITTER MODEL COMPARISONS

Experiments were conducted with a Network Performance Application (*NetPerf*) that was specifically developed to collect packet network statistics for VoIP data. *NetPerf* is written in Visual C++ to operate under the MS Windows operating systems. The program runs as a server/client pair between two computers connected to an IP network (possibly concatenated networks). The application was specifically designed to send simulated voice compressed data over IP networks, with the flexibility to accommodate various user-specified data sizes and send rates, including IP packet parameters to simulate VoIP traffic such as frame sizes, packet size (or frames per packet), send rate, duration of simulation, etc. During a simulation *NetPerf* appends packet sequence numbers on all outgoing packets, and records all packet send and arrival times (keeping track of received packet numbers). This information is used to calculate packet statistical information, such as packet inter-arrival time (jitter), end-to-end packet delay, packet loss rate, etc. Accurate timing is critical for this application; therefore the two computer's (client and server) clocks must be carefully synchronized using Network Time Protocol (NTP) [7]. *NetPerf* is a multithreaded architecture that uses multiple TCP connections for control and a UDP connection for data transfer.

NetPerf was used to collect network inter-arrival delays between a PC connected to the Wide Area Internet via an office LAN (Local Area Network) and a remote PC connected to the Internet. The packet trace (stream of voice data) sent simulated voice data every 45ms, at 14 bytes of payload per packet, from server to client. The resulting inter-arrival delay jitter is compared with the data simulated from the network delay jitter model. The model was used to generate jitter delay values based on the statistical data reported by the *NetPerf* simulation. The jitter data collected empirically using *NetPerf* had a mean of 45 ms and a standard deviation of 5.59. These values were used in determining the jitter model inputs to yield comparable results. A crossover of 180 ms, and a spike scale of 3 times the mean delay was used for Gaussian noise sampling and spike detection adjustment, respectively. Figure 5 compares the collected jitter delays from *NetPerf* and the jitter delays produced by the proposed multi-structure jitter model. The model returned a mean jitter of 45 ms with a standard deviation of 5.5, values equaling the *NetPerf* data.

Figure 5 shows that both the measured and modeled jitter are bursty, with the majority of the jitter delays appearing near the mean as expected. This is also evident in the inter-arrival delay jitter distribution shown in Figure

6. The delay distributions follow a Laplacian distribution about the mean of 45ms.

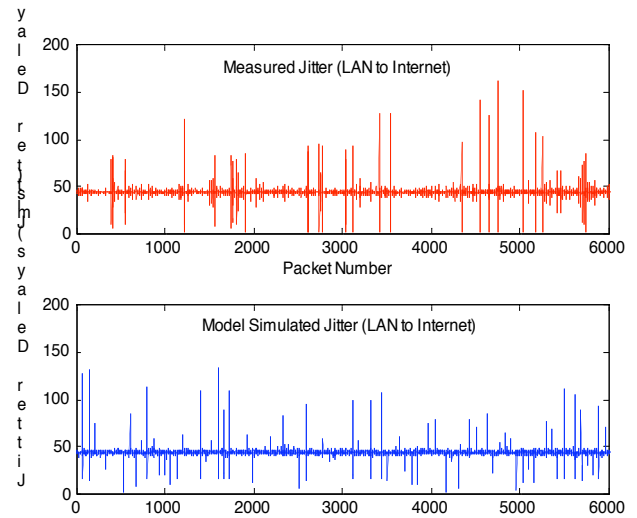


Figure 5. Measured vs. Model Jitter Delays

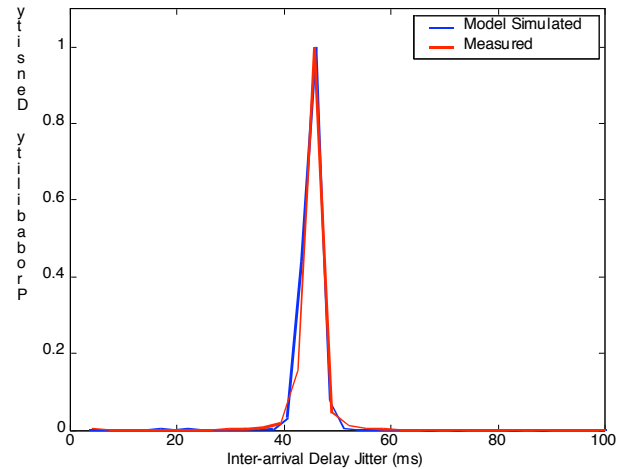


Figure 6. Jitter Delay Distribution for Measured and Modeled Jitter Delays.

The data generated from the above simulation and jitter model was compared with trace data runs over LAN (connected to the Wide Area Internet) to a CDMA data network connection. Trace runs for several times of day were compared for the LAN to CDMA data trace to get a representation of different levels of busy periods. Normalized variance characteristics are compared by averaging variance values of m -aggregated time series. This technique is typically used for self-similarity determination, as illustrated in [8]. The m -aggregated time series $x^{(m)} = \{x_k^m, k = 0, 1, 2, \dots\}$, of a stationary time series x , can be defined by summing the original time series over

non-overlapping, adjacent blocks of size m . The aggregated variance calculations essentially compress the time scales, with $m=1$ representing the highest magnification and $m=10$ a factor of 10 reduction in magnification. This can be expressed as,

$$x_k^{(m)} = \frac{1}{m} \sum_{i=km-(m-1)}^{km} x_i \quad (4)$$

In this comparison an m -aggregate time series of the variance of the jitter delays is used, illustrated in equation (5)

$$Var(x_k^{(m)}) = \frac{Var(x_{mk-(m-1)}) + Var(x_{mk-(m-2)}) + \dots + Var(x_{mk-(m-m)})}{m} \quad (5)$$

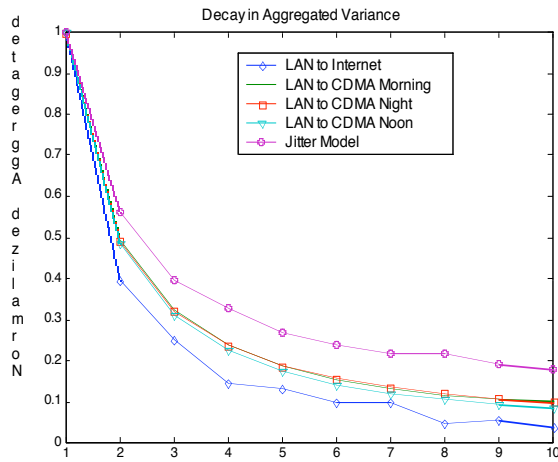


Figure 7. Aggregated Variance Comparison.

Figure 9 shows the results of the aggregate variance comparisons. For self-similar traffic the aggregate variance would decay slowly over the different time scales, maintaining a somewhat constant normalized aggregated variance value [9]. This would indicate that the variance statistics for the data series would be preserved over the various time scales and thus would describe a self-similar process. However, the simulated model and empirically measured data each decay rapidly, almost exponentially, with the jitter model data. The jitter model data does however decay slightly slower than the measured data, but not enough to conclude it is self-similar. In comparison, the overall variance statistics of the modeled data and the collected data appear to have “exponential like” decay consistent with one another. The jitter modeled data follows the short and long term bursty trends of the jitter delay collected from actual packet data networks.

5. SUMMARY AND CONCLUSION

The proposed network jitter model has promise for use in modeling and simulating packet network jitter. It captures the multi-structure characteristic of network inter-arrival delay and preserves the bursty characteristics of delay on several time scales. This model can be used to study the effects of network jitter on the operation of real-time voice and multimedia applications in packet network environments, and its affects on speech and audio quality. A direct application of this model is to evaluate the performance of adaptive jitter buffering algorithms to improve loss performance in the face of network jitter, while minimizing overall system delay.

6. REFERENCES

- [1] L. Zheng, L. Zhang, D. Xu, “Characteristics of Network Delay and Delay Jitter and its Effect on Voice over IP (VoIP)” *IEEE International Conference on Communications ICC 2001*, Vol. 1, pp. 122-126, 2001.
- [2] C. Fulton, S. Li, “Delay Jitter First-Order and Second-Order Statistical Functions of General Traffic on High-Speed Multimedia Networks” *IEEE/ACM Transactions on Networking*, Vol. 6, No. 2, April 1998.
- [3] Q. Li, D. L. Mills, “Jitter-Based Delay-Boundary Prediction of Wide-Area Networks” *IEEE/ACM Transactions on Networking*, Vol. 9, No. 5, October 2001.
- [4] Q. Li, D. L. Mills, “Investigating the Scaling Behavior, Crossover and Anti-persistence of Internet Packet Delay Dynamics,” *Proceedings of IEEE Globalcom*, Vol. 3, pp. 1843-1852, Rio de Janeiro, Brazil, 1999.
- [5] T. Yletyinen, R. Kantola, “Voice Packet Interarrival Jitter Over IP Switching,” *SBT/IEEE International Telecom Symposium ITS '98*, Vol. 1, pp. 16-21, 1998.
- [6] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne, “Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks,” *Proceedings IEEE Networking for Global Communications INFOCOM '94*, Vol. 2, pp. 680-688, 1994.
- [7] D. Mills, “Internet Time Synchronization: the Network Time Protocol” *RFC-1129*, Internet Engineering Task Force, Oct. 1989.
- [8] W. Stallings, *High-speed Networks and Internets*, Prentice-Hall, New Jersey, 2002.
- [9] M. E. Crovella, A. Bestavros, “Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes”, *IEEE/ACM Transactions on Networking*, Vol. 5, No. 6, pp 835-846, December 1997.