Correlation of Packet Delay and Loss in the Internet [1]

Sue B. Moon[2]†, Jim Kurose†, Paul Skelly‡, Don Towsley†

†Department of Computer Science
University of Massachusetts
Amherst MA 01003 USA
{sbmoon,kurose,towsley}@cs.umass.edu

‡GTE Laboratories, Inc.
40 Sylvan Road
Waltham, MA 02254
pskelly@gte.com

Technical Report 98-11
Department of Computer Science
January 1998

Abstract

In this paper we examine the correlation between packet delay and packet loss experienced by a continuous-media traffic source on the Internet. Our goal is to study the extent to which one performance measure can be used to predict of the future behavior of the other (e.g., whether observed increasing delay is a good predictor of future loss) so that an adaptive continuous media application might take *anticipatory* action based on observed performance. We ran numerous hour-long experiments in which continuous media traffic was sent from a source to a destination. We measured the per-packet delay and packet loss and then analyzed our measurements off-line. Our results provide a quantitative study of the extent to which such correlation exists. Interestingly, we observe periodic phenomena in the correlation that we had initially not expected. We discuss our results, speculate as to the reason for the observed behaviors, and discuss their implications for adaptive continuous media applications.

1 Introduction

Beginning in the 1980's, there have been a number of efforts to use the Internet for continuous-media applications, particularly audio applications [Coh77, Jay80, WF83, CD92, Sch92, JM]. The arrival of the World-Wide Web brought a flood of new continuous-media applications to the Internet, including stored video and audio servers and Internet telephony tools with video enhancement, to name but a few. These continuous-media applications differ significantly from traditional Internet applications in that they can tolerate a small amount of packet loss but also have stringent timing constraints. As the name "continuous-

[1]This research was supported in part by GTE and the National Science Foundation under Grants No. CDA-9502639, NCR-9508274.
[2]Research supported by a gift from GTE.

media" application suggests, for the receiver to play out the packets continuously, the packets must arrive at the receiver before their so called "playout time;" otherwise they will be considered "lost" by the receiver.

The current Internet does not provide any form of guarantee on packet delay or loss. Packets may be dropped or delayed at routers along a source-to-destination path as a result of network congestion. There are efforts to change this "best-effort" service paradigm in the Internet to provide integrated service for different types of applications[Wro97, SPG97, CW97]. Given today's best-effort Internet service (or even alternate service models such as differential service or controlled load service in which performance is not guaranteed), however, continuous media applications will continue to suffer impairments due to varying packet loss and network delay within a connection. One approach for handling such impairments is to adapt application behavior in response to changing network conditions. A first step in designing such adaptive mechanisms is to understand the dynamics of the environment in which they must operate.

In this paper we report on the correlation between delay and loss observed by a continuous-media traffic source on the Internet. Our goal in undertaking this study is to determine the extent to which one performance measure could be used as a predictor of the future behavior of the other (e.g., whether observed increasing delay is a good predictor of future loss) so that an adaptive continuous media application might take *anticipatory* action based on observed performance. In this study, we ran numerous hour-long experiments in which packets were periodically sent from a source to a destination. We measured the per-packet delay and packet loss and then analyzed our measurements off-line. Our results provide a quantitative study of the extent to which such correlation exists. Interestingly, we observe periodic phenomena in the correlation that we had initially not expected. We discuss our results, speculate on the observed behaviors, and discuss their implications for adaptive continuous media applications.

In the following section we discuss the background and motivation for our work. In Section 3, we describe the tool used to collect measurements, and discuss the technical issues surrounding the measurement process and the evaluation of the empirical results. We introduce a quantitative measure for correlation between delay and loss in Section 4, and apply it to analyze the measurements in Section 5. Section 6 concludes this report with a summary and a discussion of future research.

2 Motivation

Traditional Internet applications such as *telnet* and *ftp* use TCP as their transport layer protocol, and depend on TCP congestion control when there is congestion in the network. In TCP, the sender increases its transmission rate additively until it experiences a packet loss, which is taken as an indication of congestion. TCP then decreases its transmission rate multiplicatively, thus reacting quickly to the inferred congestion. As a result of this behavior, the transmission rate of the TCP sender is determined by the level of congestion in the network.

Continuous-media applications, on the other hand, usually react to network congestion with less flexibility (if at all), due to their more stringent timing constraints. Figure 1 illustrates these constraints. Here, packets from continuous-media applications must arrive at the receiver before their "playout time;" otherwise they are considered "lost." Figure 1 indicates how packets can be lost either in the network or at the receiver due to late arrival. Packet losses in continuous-media applications, whether from late arrivals or dropped packets, degrade the quality perceived at the receiver. Applications can choose to ignore lost packets or recover from them via retransmission [PP96] or via receiver-based error recovery/masking [BG96].

Continuous media applications can be not only loss-adaptive, but delay-adaptive as well. That is, they can have either a fixed playout time, or adapt to changes in packet delay and set the playout time accordingly [RKTS94]. Such adaptive applications keep track of packet delays, and reflect any change in packet delays in their calculation of "playout time."

Given the above discussion, it is clear that packet loss and delay have tremendous impact on continuous

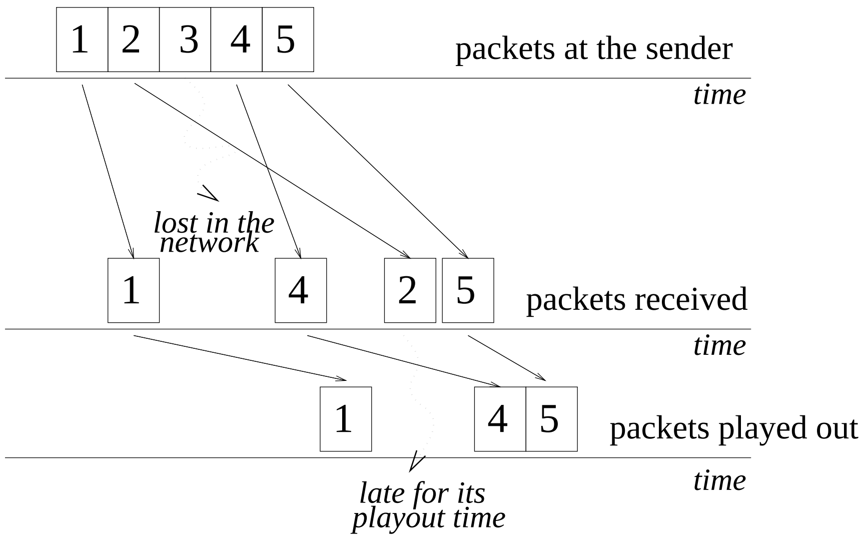


Figure 1: Timing constraints and packet loss

media applications. Both loss and delay result from buffering within the network. As packets traverse the network, they are queued in buffers (thus adding to their end-end delay) and from time to time are dropped due to buffer overflow. Consider then a continuous media packet stream at a buffer that is filling up fast with packets from other traffic sources as well. The packets from the continuous-media application continue to be queued up in the growing packet queue together with the packets from other sources. Continuous media packets arriving at the receiver experience progressively higher end-end delays than earlier packets. When the buffer reaches its capacity, packet losses begin to occur. The receiver of the continuous-media application thus sees increased delay, and eventually losses.

Consider next a scenario in which packets from a continuous-media application arrive at a buffer that is already full. In this case, they are dropped. As other sources (e.g, TCP connections) detect congestion and decrease their transmission rate, the queue length at the buffer will decrease, and packets from the continuous-media application will start to be queued, rather than dropped. In this scenario, the receiver sees losses followed by high, but possibly decreasing, packet delays.

The two examples above are plausible scenarios. In the first example, the receiver could have taken the increased delay as an indication of likely future packet loss; in the second scenario, the decreasing delay following loss indicates a future uncongested period of time. In both cases, the application could adapt its behavior accordingly. But do we expect such scenarios to occur (or be detectable) often in practice? Note that there are a number of implicit assumptions in the discussion above, including a single bottleneck link and presumed behavior by other competing applications. If delay is indeed affected by loss, or vice versa, how temporally close are they? What is causing such correlated behavior in the network? How can we exploit such information at the end-system? These are some of the issues that we touch on in the following sections.

3 Measurements

To collect end-end delay and loss data for a continuous media source, we built a tool that generated packets with RTP headers [SCFJ96, Sch96] at a fixed, periodic rate. The default RTP header has a fixed length of 12 bytes, and includes the version number, sequence number, media-specific timestamp, and source identifier. The sequence number is increased by one per packet, while the increment of a timestamp is dependent on

Trace	Destination	Start time(Source)	Duration	Loss Prob.
1	float-71a.cs.virginia.edu	Tue, 2/11/97 11:12pm	1800sec	0.027
2	maria.wustl.edu	Mon, 3/24/97 7:31pm	2348sec	0.010
3	alps.cc.gatech.edu	Tue, 3/26/97 7:10pm	1864sec	0.041
4	edgar.cs.washington.edu	Fri, 3/28/97 7:58pm	2343sec	0.013
5	cedar.csres.utexas.edu:9898	Tue, 2/18/97 7:23pm	1808sec	0.091
6	anhur.sics.se	Mon, 3/24/97 9:53pm	2228sec	0.23

Table 1: Trace Descriptions

the payload type. We kept a log at both the source and the destination of real timestamps from the system clock along with the RTP sequence number and media-dependent timestamps; these are used to calculate end-end delay. To minimize the load inside the network, each packet had only the 12-byte RTP header, and did not carry any audio data.

To generate packets at a periodic interval of less than 100ms, we use the workstation's audio device for timing. Most workstations provide a timer in their operating systems, but its granularity is usually in hundreds of milliseconds, and not fine enough for our measurement purpose. Audio devices on most workstations interrupt the operating system when the audio buffer is full, with the interval between two consecutive interrupts being in the tens of milliseconds. The application programming interface to the audio device on most workstations allows the fine-tuning of the interrupt interval by changing the audio buffer size. We used the audio device interrupt to generate packets periodically at 20 ms intervals.

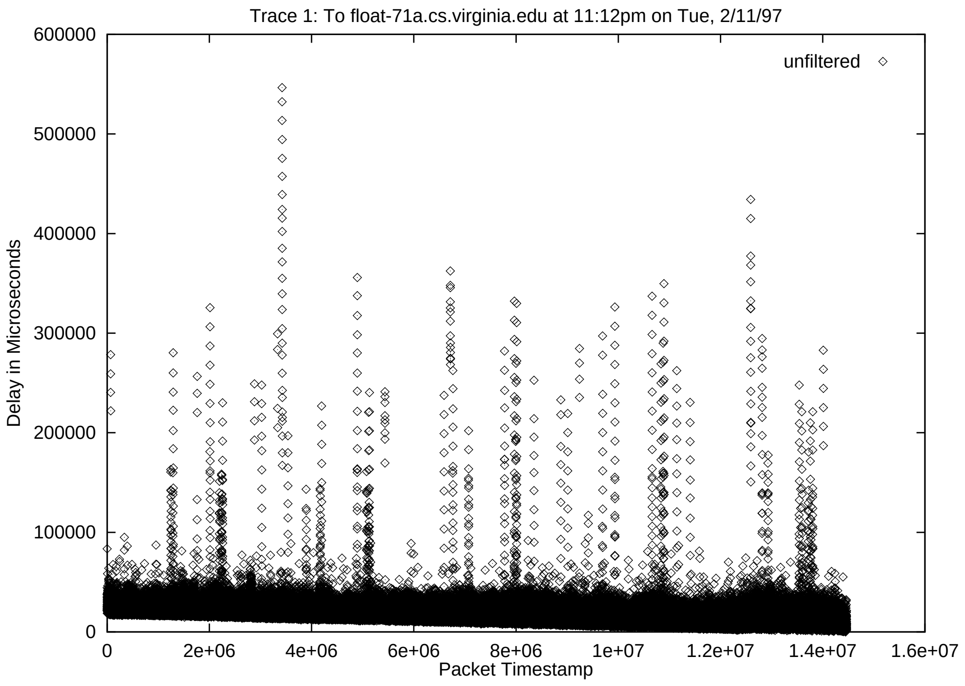
Because timing and delay were so critical to our measurements, it was important for us to verify that packets were indeed being sent at 20 ms intervals, and not being inordinately delayed in the operating system or network interface. We thus first ran a 30-minute-long experiment between two SGI Indy machines on the same branch of a LAN. The results showed that there was no loss between the two machines, and that sender packet inter-departure times were kept almost constant at 20ms.

We collected six sets of measurements between February and March of 1997. Table 1 describes when and where the measurements took place. All the traces originated from two machines at the University of Massachusetts. Traces 1 to 3 were to sites on the east coast, Traces 4 and 5 to sites on the west coast, and Trace 6 to an European site.

Before analyzing the data, two important aspects of the measurement process bear further discussion. The first issue is clock drift. If the clocks at the source and destination are not synchronized, and have a frequency drift, the delay measurements show a gradual increase or decrease over time depending on which clock is faster. Figure 2 plots the packet timestamp versus per-packet delay, showing a gradual decline of delay over time. In this case, the sender clock is faster than the receiver clock. We used linear regression to filter out such clock drift in delay measurements. The filtered results for the raw data shown in Figure 2 is shown in Figure 3.

Another issue to be considered is routing. If packets are routed through different paths to the destination, the resulting delays and losses at the destination come from different queues in the network. This would complicate our analysis of correlation between losses and delays. A recent study by Paxson[Pax97] suggests that route changes can be detected by out-of-order deliveries of packets (although out-of-order packets do not necessarily imply a route change). In his traces, less that 1% of packets were delivered out of order. In all our traces, out-of-order packets constitute less than 0.1% of the total number of packets. We do not consider out-of-order packets in our analysis below.

Figure 2: Before Linear Regression



4 Correlation between Delay and Loss

Let us now introduce the notation to be used in our data analysis. We define the following:

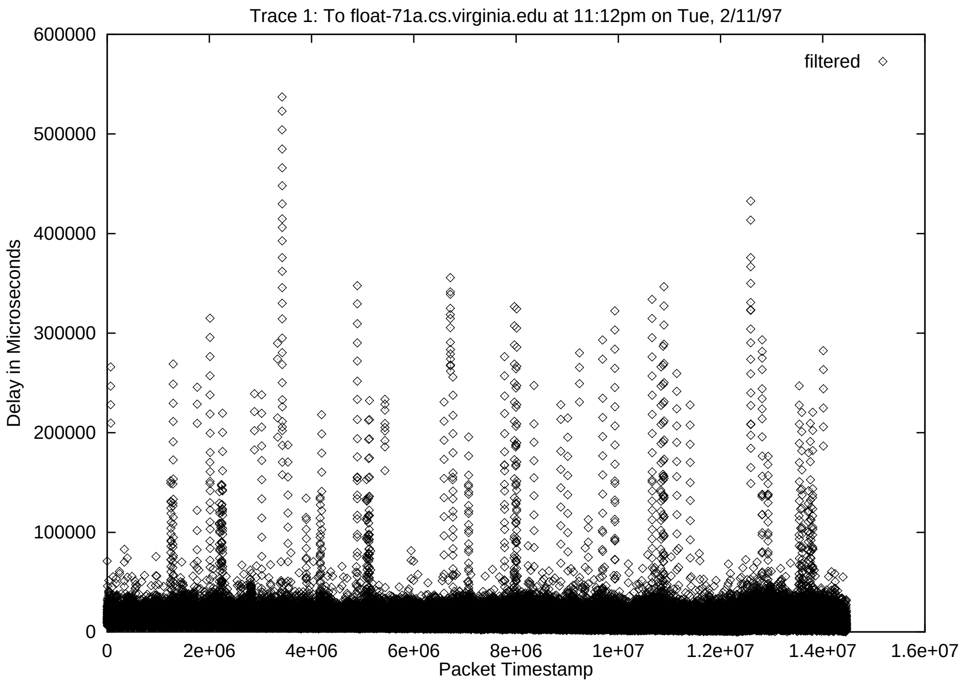
- N: length of a trace.
- i: packet sequence number, $1 \leq i \leq N$
- l_i: indicates a loss.

$$l_i = \begin{cases} 1, & \text{if the packet is lost.} \\ 0, & \text{if the packet is delivered to the destination.} \end{cases}$$

- d_i: delay of the i-th packet. $d_i \geq 0$. If $l_i = 1$, i.e., if the i-th packet is lost, arbitrarily take d_i to be zero.
- M: number of packets that are delivered to the destination. Note that $M \leq N$ and $M + \sum_{i=1}^{N} l_i = N$.
- $\bar{d} \equiv \sum_{i=1}^{N} d_i/M$: average delay.
- $\bar{l} \equiv \sum_{i=1}^{N} l_i/N$: loss frequency.

If we plot d_i and l_i together as a function of i, we may see a temporal correlation between delay and loss. As the d_i's increase over a period of time due to congestion, there may be increasingly more l_i's that take a value 1 near (before, after, or within) intervals of high delay. As a means to quantify this correlation between delay and loss, we can consider the average delay conditioned on loss, as discussed below. We cannot use

Figure 3: After Linear Regression



the cross-correlation between delay and loss, because they represent different aspects of a phenomenon, and their values are not compatible. Furthermore, since a packet either arrives at a receiver with delay, or does not arrive at all, we cannot calculate the average delay of packets conditioned on their own loss.

We introduce a *lag* in calculating the average delay conditioned on loss. Specifically, the average packet delay, conditioned on a loss occurring at a time lag j packets in the past, is the average delay of all packets in the trace that have a loss j packets before them in the trace. That is,

$$E[d_i | l_{i-j} = 1] = \sum_{k \in P} d_k / |P|, \quad \text{where} \quad P = \{k : l_{k-j} = 1 \text{ and } l_k = 0\}. \tag{1}$$

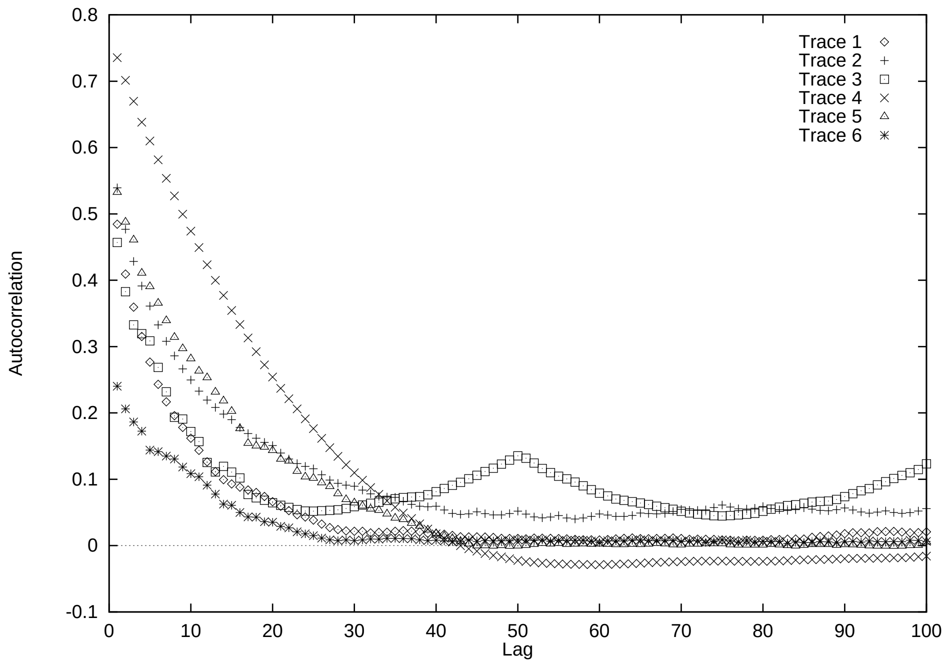
We will refer to the above quantity as the *loss-conditioned average delay.* If the loss-conditioned average delay at a positive lag of j is higher than the unconditional average delay (i.e., the delay averaged over all received packets), it means that packets that arrive j packets after a loss have a higher average delay than the unconditional average delay. That is, a loss occurring j packets in the past can be taken as a precursor to a higher delay later. A literal interpretation of a higher conditional average delay at a negative lag is less intuitive. In such a case, a loss in the future can be thought of as an indicator of higher delay at the current moment. We revisit this issue later in this section.

Before proceeding to quantitatively evaluate the loss-conditioned average delay, we first look at the autocorrelation of delay. The autocorrelation of delay at lag j is:

$$r_j = \frac{\sum_{i=1}^{N} (d_i - \bar{d})(d_{i+j} - \bar{d})}{\sum_{i=1}^{N} (d_i - \bar{d})^2}$$

As shown in Figure 4, the autocorrelation of delay eventually dies out over time as the lag increases. Trace 3, however, stands out from others. Since the autocorrelation of Trace 3 shows that delays with a lag of approximately 50 are correlated, Trace 3 shows periodicity in other measures of performance, as we

Figure 4: Autocorrelation of Delay



discuss it in the next section. The autocorrelation of delay provides another measure of performance to look at the temporal correlation of data.

The loss-conditioned average delay at a positive lag allows us to look at delays of future packets following a loss event. To look at future loss conditioned on current packet delay, we can count the number of packets lost at a specific lag from a packet experiencing a given delay. This is expressed by the following:

$$C_j = |\{l_i : d_{i-j} > T\}| \tag{2}$$

Here, we are counting the losses conditioned on a packet at time lag j having a delay above a given threshold, T. By varying T within the range of $\bar{d}$ and $2\bar{d}$, we can observe how sensitive the losses are to the delays. Figure 5 indicates that the conditional loss count as a function of lag, computed via (2), is not particularly sensitive to the threshold value. Trace 3 stands out from other traces due to its definite periodicity.

5 Measurement Analysis: Loss-conditioned Average Delay

Figure 6 plots the loss-conditioned average delay of all six traces listed in Table 1, using (1). The range of the lag is between -500 and 500, which corresponds to 10 seconds in both positive and negative directions. Except for Trace 4, all traces show higher loss-conditioned average delay near lag 0. This can be explained by an FCFS buffering process where packets that enter the queue just before an overflow experience a large queueing delay (since the queue is nearly full). Similarly, those packets successfully entering the buffer soon after an overflow event will also see a large queueing delay, since the queue is still nearly full

Another general observation of the graph is that the loss-conditioned average delay shows periodic behavior, even dropping *below* the value of the unconditional delay at a given time lag and then rising again above that value for a larger lag. Traces 1, 2, and 4 have a trace loss probability of less than 0.03. Their graphs show a well-pronounced rise between lags of 50 and 100, a few more after lag 100 and when lags are negative. As the loss probability increases from those traces to Traces 3, 5, and 6 in an increasing order, the correlation graph flattens out.

These observations raise several interesting questions. First, what causes the oscillations in Traces 1, 2, and 4? Why is it located between lag of 50 and 100. Why does it disappear as the loss probability increases?

We can suggest a few possible answers these questions if we first make a few assumptions. First, we assume that there is just one congested link along the path from a source to a destination, and that packets are delayed or dropped mostly at that link. This allows us to focus on single-hop behavior, and the impact of other traffic sources on the observed performance of our probe traffic. Second, let us assume that a significant number of the sources that are competing in this bottleneck queue use some form of TCP congestion control. All applications on the Internet use either TCP or UDP as their transport protocol. Theoretically applications that use UDP can consume unlimited bandwidth, and their behavior is unpredictable. However, most continuous-media applications using UDP try to transmit at a fixed rate that is typically determined by the bottleneck bandwidth.

These two assumptions allow us to focus on the impact of other traffic sources on our probe traffic at a single congested link. Returning back to our earlier observations regarding oscillations, we conjecture that some form of synchronization among traffic sources causes the oscillations in the loss-conditioned average delay graphs with the loss probability less than 0.05. When congestion is detected in the network, all TCP sources whose packets are routed through the congested link initiated congestion avoidance when a loss is detected and cut down the size of their congestion windows. Depending on the round-trip time from those sources to their destinations, they will then start to increase their window size, until congestion again occurs at this link. In this scenario, *the loss events in the congested link serves to synchronize the increase and decrease in traffic* offered by the TCP sources passing through this link. The continuous-media source, which can be thought of as a periodic "probe" that samples the congested queue state, thus observes the periodic rise and fall of traffic (and the concomitant rise and fall in delay) that occurs as a result of loss-induced synchronization amount TCP connections passing through this link.

Earlier simulation work reported that TCP sources can behave in such a synchronized fashion[SZC90]. A recent study observed in simulations that a UDP source sees more losses if the interfering traffic from TCP sources is synchronized[SS97]. We intend to to further investigate this phenomenon in the second year of this project to see if such a synchronization is indeed the cause of what we have observed in our traces.

Trace 6 stands out from other traces because it shows very little fluctuation in its loss-conditioned average delay over the entire range of lags. This can be explained by noting that the trace was collected over a very lossy link between UMass and Sweden. The loss probability is over 0.20 – more than an order of magnitude larger than that of any other traces. The high loss probability is an indication of heavy congestion on the bottleneck link. We conjecture that as more losses are detected by the sources, the sources adjust their congestion windows more often, and their self-synchronization becomes less prominent.

6 Conclusions and Future Work

Our analysis of the delay and loss measurements using the loss-conditioned average delay raises several interesting questions. We can guess about the causes of the observed oscillatory behavior, but we can not be sure. If it indeed results from synchronization among other TCP sources, can we validate our claim using simulations? As part of our second year project, we plan to use ns[ns97] as a simulation tool, run several TCP sources with a UDP source through one common bottleneck link, and see if we observe similarly correlated behavior between delay and loss as in our traces.

The synchronized behavior among sources is not desirable in several respects. First, the utilization level of the congested link fluctuates severely. When the link is congested, packets are dropped, and when the link is under-utilized, the bandwidth is wasted. RED (Random-Early Detection) has been proposed to address this lock-out phenomenon of synchronization[FJ93]. We plan to run simulations of the RED queue management to test if it breaks down the synchronization, and if correlation is no longer detectable by our measure of loss-conditioned average delay.

We note that the traces we took can be used in a trace-driven IP voice simulator. Another approach is to develop and validate an analytical model for the delay and loss, and use it in simulations. This in an additional interesting area for further investigation.

Finally, it is of interest to examine the implications of delay and loss correlation for the various application-level congestion control and repair mechanisms for continuous-media applications. If delay and loss are closely correlated temporally, can we make use of this knowledge to enhance the control at the end systems? What kind of control is effective given the correlation? Is retransmission an effective mechanism given the timing constraints and the observed correlation? On what time-scale is such a reactive approach towards congestion control practical?

Acknowledgments

We would like to thank Jorg Liebeherr at the University of Virginia at Charlottesville, Chuck Cranor of Washington University, St. Louis, Peter Wan of Georgia Institute of Technology, Linda Prime, Renee Reed, and Erik Lundberg at University of Washington, Simon S. Lam and David K. Y. Yau at University of Texas at Austin, and Stephen Pink of Swedish Institute of Computer Science for providing us the computer accounts on which we took the measurements.

References

[BG96] J. Bolot and A. Vega Garcia. Control mechanisms for packet audio in the internet. In *Proceedings of IEEE INFOCOM '96*, pages 232–239, 1996.

[CD92] Stephen Casner and Stephen Deering. First IETF Internet Audiocast. *ACM Computer Communication Review*, pages 92–97, July 1992.

[Coh77] Danny Cohen. Issues in transnet packetized voice communication. In *Proc. Fifth Data Communications Symposium*, Snowbird, UT, September 1977.

[CW97] D. Clark and J. Wroclawski. An approach to service allocation in the internet. Internet-Draft draft-clark-diff-svc-alloc-00, Internet Engineering Task Force, 1997.

[FJ93] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1993.

[Jay80] N. S. Jayant. Effects of packet loss on waveform coded speech. In *Proc. Fifth Int. Conference on Computer Communications*, Atlanta, GA, October 1980.

[JM] V. Jacobson and S. McCanne. `ftp://ftp.ee.lbl.gov/conferencing/vat/`.

[ns97] Ucb/lbnl/vint network simulator. http://www-mash.cs.berkeley.edu/ns/, 1997.

[Pax97] Vern Paxson. End-to-end internet packet dynamics. In *Proceedings of SIGCOMM '97*, 1997.

[PP96] C. Papadapolous and G. Parulkar. Retransmission based error control for continuous media applications. In *Proceedings of NOSSDAV 1996*, 1996.

[RKTS94] Ramachandran Ramjee, Jim Kurose, Don Towsley, and Henning Schulzrinne. Adaptive playout mechanisms for packetized audio applications in wide-area networks. In *Proceeding of INFO-COM '94*, 1994.

[SCFJ96] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A transport protocol for real-time applications. RFC 1889, Internet Engineering Task Force, jan 1996.

[Sch92] Henning Schulzrinne. Voice communication across the Internet: A Network Voice Terminal. Technical report, Dept. of ECE, Dept. of CS, University of Massachusetts, Amherst, MA 01003, July 1992.

[Sch96] H. Schulzrinne. RTP profi le for audio and video conferences with minimal control. RFC 1890, Internet Engineering Task Force, jan 1996.

[SPG97] S. Shenker, C. Partridge, and R. Guerin. Specifi cation of guaranteed quality of service. RFC 2212, Internet Engineering Task Force, 1997.

[SS97] Hidenari Sawashima and Yoshiaki Hori Hideki Sunahara. Characteristics of UDP packet loss: Effect of tcp traffi c. In *Proceedings of INET '97: The Seventh Annual Conference of the Internet Society*, Kuala Lumpur, Malaysia, June 1997.

[SZC90] Scott Shenker, Lixia Zhang, and David D. Clark. Some observations on the dynamics of a congestion control algorithm. In *Proceedings of SIGCOMM '90*, pages 30–39, 1990.

[WF83] Clifford Weinstein and James W. Forgie. Experience with speech communication in packet networks. *IEEE Journal on Selected Areas in Communications*, 6(1):963–980, 1983.

[Wro97] J. Wroclawski. Specifi cation of the controlled-load network element service. RFC 2211, Internet Engineering Task Force, 1997.

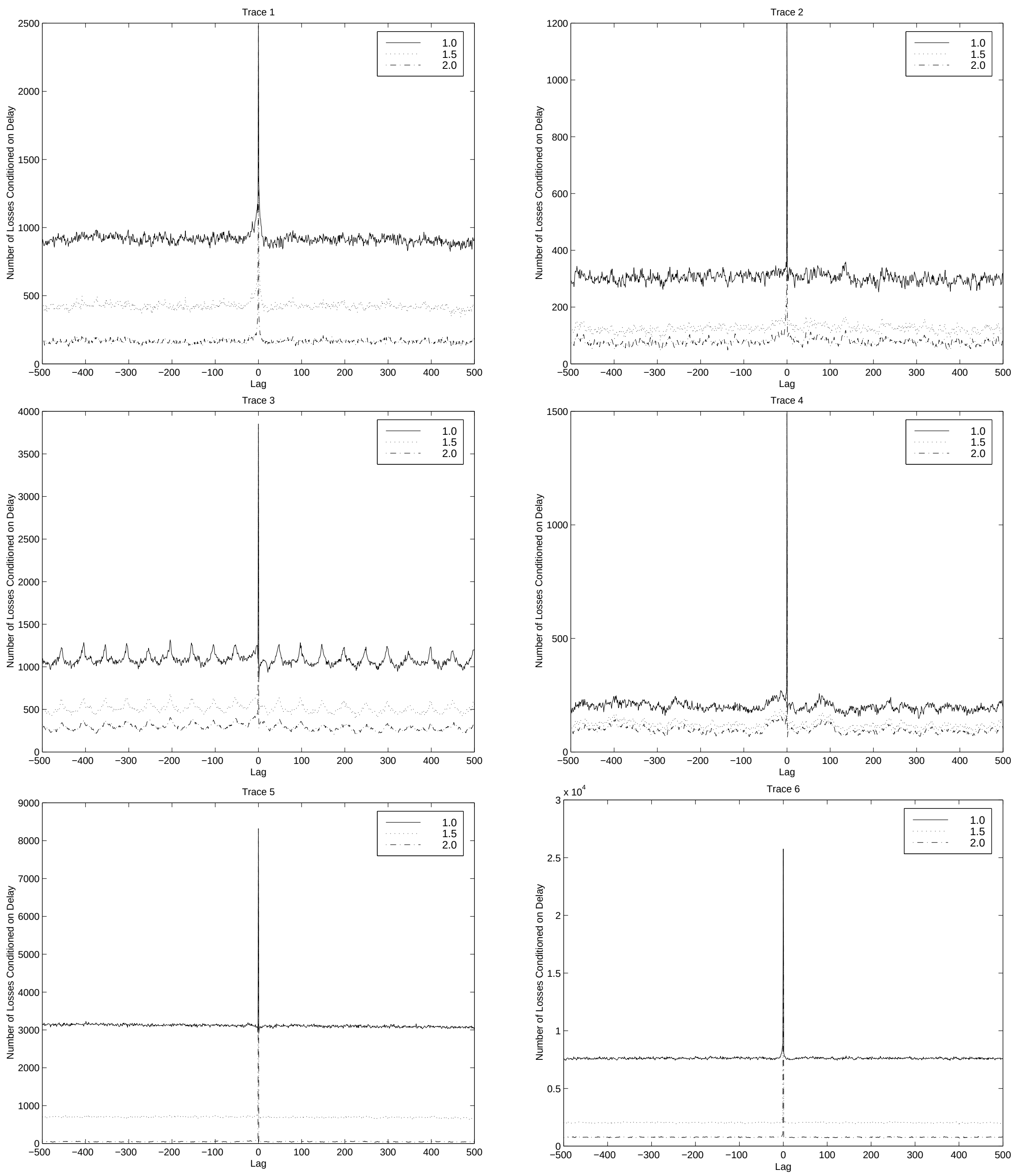


Figure 5: Number of Packets Lost Conditioned on Delay

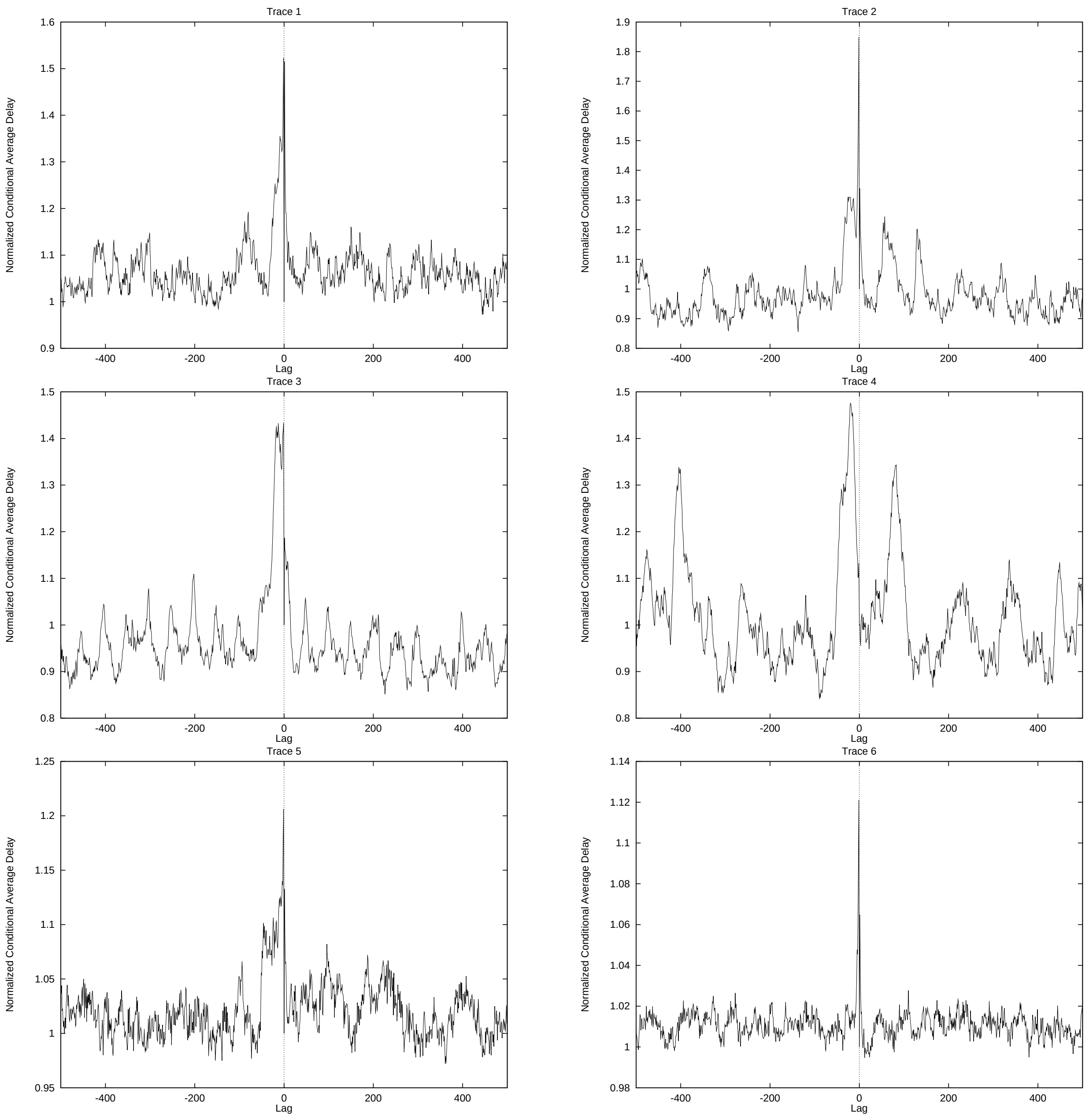


Figure 6: Normalized Conditional Average Delay on Loss