Measurement and Interpretation of Internet Packet Loss

Michael S. Borella*
3Com Corp.
1800 W. Central Rd.
Mt. Prospect IL 60056
USA

Abstract

We analyze a month of Internet packet loss statistics for speech transmission using three different sets of transmitter/receiver host pairs. Our results indicate that packet loss is highly bursty, with the majority of individual losses occurring in a relatively small number of bursts. We find that loss exhibits dependence in most cases, but is not always well-modeled as dependent. We introduce an analytical technique for measuring loss dependency. We also consider the asymmetry of round-trip packet loss, and find that most loss on a round-trip path occurs in either one direction or the other. We introduce a normalized metric for measuring loss asymmetry and apply it to our measurements. Finally we discuss the implications of our study for the next generation of real-time voice services in the Internet.

Keywords: Internet packet loss, real-time services, quality of service, dependence, asymmetry.

1 Introduction

As the Internet continues to grow, the need for tools and benchmarks for empirical performance analysis becomes more important. Ideally, empirical data should be collected not only at end points, but also at the intermediate hops of a one-way or round-trip Internet path. Unfortunately, fine-grained logging at Internet routers or Network Access Points (NAPs) would require increasing the load on systems that may already be very busy. As an alternative, end-to-end measurements are commonly used. A number of end-to-end measurements have been performed for round-trip UDP delay and loss [1] [2] [3], route dynamics [4] as well as unidirectional and round-trip TCP delay and loss [5] [6]. These studies reveal interesting and often surprising Internet behavior and serve to reinforce the importance of Internet performance sampling.

In this paper, we focus on measurement and analysis of UDP packet loss in the Internet. Both unidirectional and round-trip loss is considered. Packet loss can negatively impact applications whether they use UDP or TCP for transport. Real-time applications using UDP, such as Internet telephony or video

*Tel: 847 342-6093, Fax: 847 222-0573, Email: mike_borella@3com.com

conferencing, suffer quality of service (QoS) degradation when loss is excessive. While TCP provides reliable end-to-end transmission through its timeout-and-retransmission algorithms, a single packet loss causes TCP to decrease its transmission rate with either its *congestion avoidance* or *slow start* techniques [7]. Thus, it is difficult for TCP to maintain high throughput over a lossy path. In particular, both UDP- and TCP-based applications are often highly sensitive to loss over paths for which the delay-bandwidth product is large.

We have collected a month of packet loss statistics from speech transmission over three different Internet paths. The parameters of our experiments are similar to those of the ITU G.723.1 recommendation for compressed voice transmission over packet-switched networks [8]. The results of our analysis indicate that individual packet losses exhibit a varying amount of dependence on previous packet losses, but are, in general, very bursty. For example, on one path less than 1% of all loss bursts accounted for nearly 50% of all individual losses. We examine the dependence and predictability of loss through the use of the conditional loss graph. Finally, we find that the majority of loss on a round-trip path is asymmetric, and introduce a normalized metric to measure loss asymmetry.

This paper is organized as follows. Section 2 describes the software we used to collect data statistics and the design of our month-long experiment. Section 3 discusses packet loss burstiness. Section 4 analyzes the dependence structure of the measured loss. Section 5 quantifies the asymmetry of this loss. Section 6 concludes with a discussion of the implications of these results.

2 Experimental Design

This section discusses the software used to collect packet loss statistics and discusses measured loss over a one month period for three paths in the Internet.

2.1 Software

We have developed a set of tools to measure packet loss in the Internet. Our goal was to be able to measure the application-level loss that a real-time UDP traffic stream would experience. Thus we wrote a distributed application layer program consisting of a client and a server to be run on two different hosts (see Figure 1). The client program consists of two processes running in parallel; a transmitter and a receiver. The client's transmitter process uses the UNIX interval timer to schedule transmission of a stream of UDP packets to the server with regular inter-departure times. Since an operating system timer sends the transmitter process an interrupt at these regular intervals, self-synchronization effects are minimized[1].

[1]Self-synchronization *will* occur if the transmitter process schedules each interrupt itself after servicing the previous one.

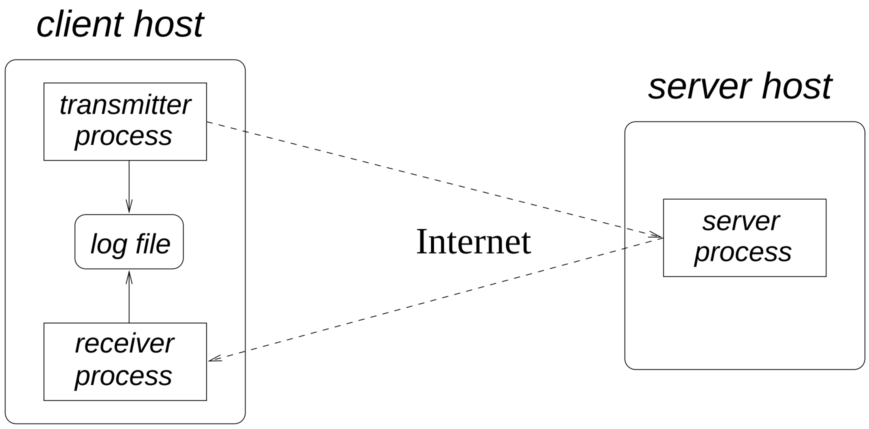

Figure 1: Client/server architecture for packet loss measurement.

Each packet contains a sequence number and a client identifier (CID), which is a random 10-bit string chosen at the beginning of a session. The server saves the last CID that it has received. When the server receives a packet with a CID different than the one that it has stored, it sets its packet count (PC) register to 0. When the server receives a packet with a CID identical to the one it has stored, it increments its PC by 1. Each packet received by the server is echoed back to the client, and, if it is not lost, is received by the client's receiver process. The receiver process logs the sequence number of all packets that it receives. When the sender process has completed sending a stream, it transmits an end-of-transmission token (EOT) to the server, which responds with the current contents of its PC register. If either the EOT or PC packets are lost, the transmitter will timeout and retransmit the EOT until it successfully receives the server's response. Using this software, we can measure client-to-server loss rate (λ_c) and server-to-client loss rate (λ_s) as well as round-trip loss rate (λ_r). Note that λ_r is the geometric mean of λ_c and λ_s. Our method is robust as long as no more than one client attempts to communicate with a server simultaneously. The goal of this software is to measure Internet characteristics and behavior *as would be experienced by an application-layer service.*

2.2 Data Collection

During the month of April, 1997, we measured packet loss between three sites on the Internet. Our hosts were `tcres.cs.depaul.edu` and `dbis.eecs.uic.edu`, both in Chicago, Illinois, and `tabasco.cs.ucdavis.edu`, in Davis, California (see Figure 2). Each site ran both a client and a server, and statistics were collected on the following round-trip paths: (i) Path TC-DB: `tcres` to `dbis` to `tcres`, (ii) Path DB-TA: `dbis` to `tabasco` to `dbis`, and (iii) Path TA-TC: `tabasco` to `tcres` to `tabasco`.

For example, on path TC-DB, `tcres` ran the client and `dbis` ran the server, so round-trip delay was

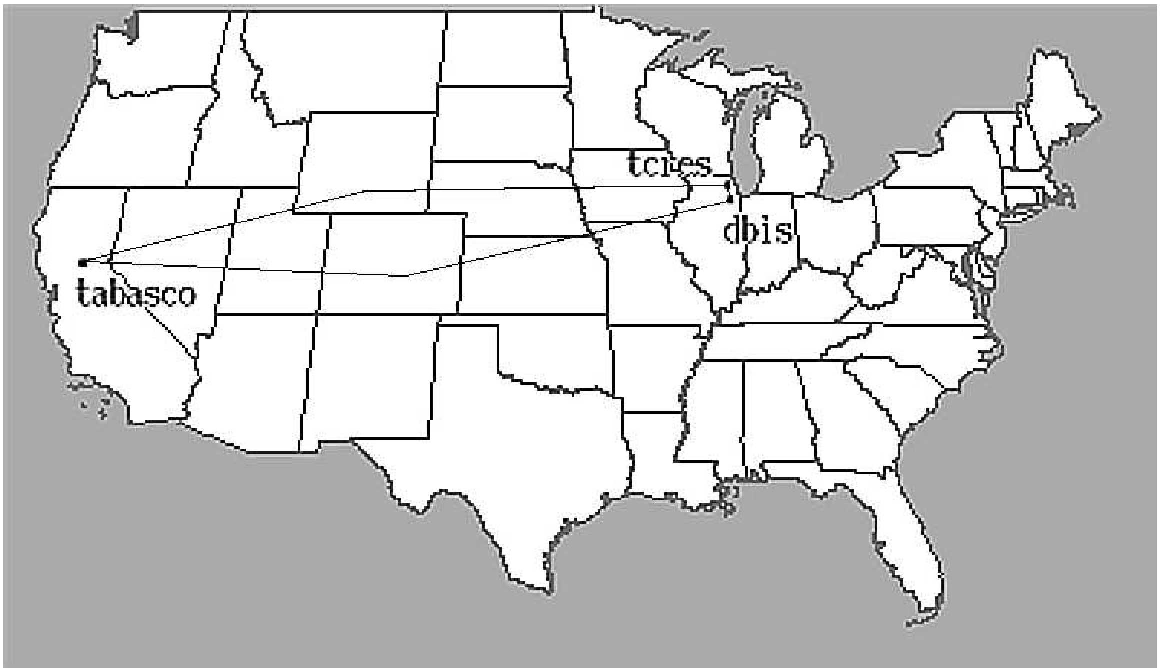

Figure 2: Hosts and paths used for loss measurements.

measured from `tcres` to `dbis` and back to `tcres`. These paths were chosen so that we could measure loss on both wide-area (between Chicago and California) and metropolitan (between DePaul and UIC) portions of the Internet.

During each day of the month, each client transmitted a run of packets to its associated server for a three minute interval, once per hour[2]. Packets were 80 bytes in length and inter-departure times were regular intervals of 30 ms. These transmission parameters were chosen to correspond to those used for voice transmission using G.723.1 [8], which specifies 30 ms inter-departure times and a 20- or 24-byte payload. Ideally, each run should consist of exactly 6000 packets, but in practice background load on the client host and on the client host's LAN occasionally prohibited a packet from being transmitted as scheduled. As a result, the runs generally consisted of between 5900 and 6000 transmissions. Occasionally, there was no data collected for a particular path, during a particular hour. This could have happened for a number of reasons: planned or unplanned network service outages, a host being taken down or rebooted, or the software failing to run properly for some reason. Our measurement mechanism was robust in the sense that we never had to manually restart it[3] and the frequency of failed measurements was reasonably low –

[2] We made sure that each path ran the software at non-overlapping intervals within the hour so that interference between two sets of measurements would not be an issue. Each client's transmissions were one hour apart.

[3] In fact, after setting the measurement mechanism, which consisted of a number of `crontab` entries and Perl scripts, in motion during late December 1996, we did not touch it again until early July 1997, when we completed our goal of a six-month study.

less than 3.6% of the runs over the course of the month.

3 Loss Burst Lengths

Table 1 summarizes the round-trip loss statistics for each path. Runs refers to the total number of runs that successfully transmitted at least one packet. Given the 30 days in April, at best 720 runs would have been transmitted on each path. Packets transmitted, packets lost, and loss rate are cumulative measurements. We note that overall, packet loss was not excessive.

Path	Runs	Packets transmitted	Packets lost	Loss rate	Loss bursts
TC-DB	704	4,091,827	14,907	0.36%	1,001
DB-TA	682	4,045,161	24,825	0.61%	3,892
TA-TC	697	3,801,185	134,687	3.54%	20,385

Table 1: Summary of round-trip loss statistics for each path.

We define a *loss burst* to be an event of one or more consecutive losses. The results of Table 1 indicate that loss is quite bursty – the mean number of packets lost in a single burst is 6.9. However, these numbers do not tell the whole story. We also found that the distribution of burst length has a very heavy upper tail; thus, most individual packet losses occur in a relatively small number of bursts. In particular, path TC-DB had a loss burst consisting of 5476 consecutively lost packets; in other words, 0.1% of the bursts on this path accounted for 36.7% of the individual packets lost.

To capture this heavy-tailed behavior, we modeled the distribution of loss burst lengths, with the Pareto distribution, which has a CDF defined as

$$F(x) = 1 - x^{-\alpha} \quad \alpha > 0 \tag{1}$$

The Pareto CDF obeys a power-law on both axes; thus, when plotted on log-log coordinates, it appears as a straight line. Pareto distributions have been used to model many physical, sociological and economic phenomena, such as the distribution of city populations and the worldwide distribution of wealth. It is interesting to note that when $1 < \alpha \leq 2$ the distribution exhibits infinite variance, while when $0 < \alpha \leq 1$, it exhibits infinite variance *and* infinite mean. This indicates that empirical distributions modeled as Pareto with $\alpha \leq 2$ exhibit extreme characteristics that cannot be captured by "traditional" distributions, such as the exponential.

Figure 3a shows the complementary cumulative distribution plot of loss burst length, conditioned on a single packet loss. Performing a regression on the log-log coordinates of this plot gives us the α parameter

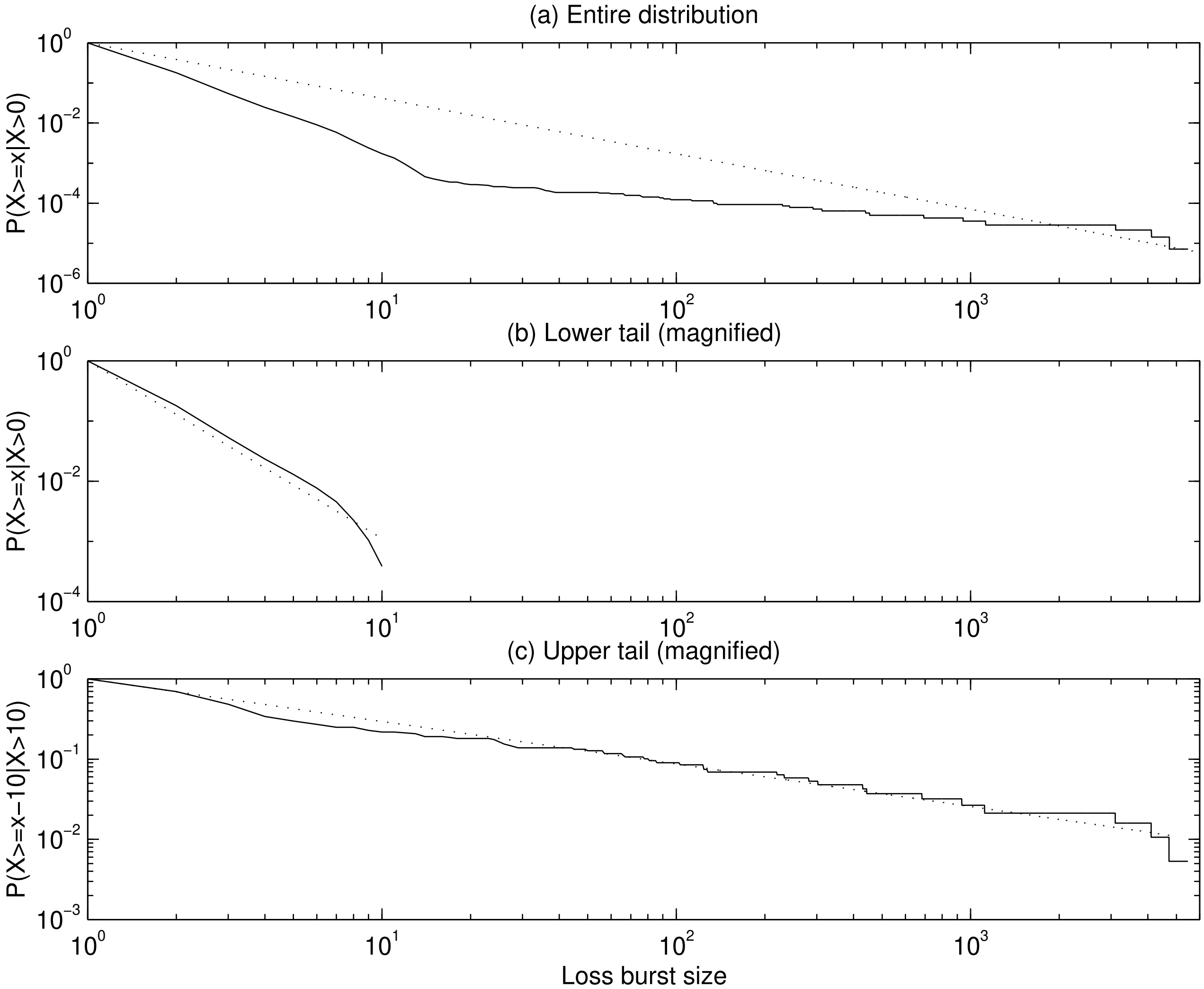

Figure 3: Complementary cumulative distribution plots of loss burst length over all three paths.

for a fit to a Pareto distribution. This fit is shown with a dotted line. Visual inspection of this plot indicates that there is a significant nonlinearity at a burst length of about 10 packets. The log-log regression produces $\alpha = 1.38$, but the associated Pareto distribution does not describe the empirical distribution very well at all.

In Figures 3b and 3c, we examine the distribution on either side of the non-linearity individually. Figure 3b shows a Pareto fit to the lower tail (burst lengths ≤ 10 packets, conditioned on a single loss) and Figure 3c shows a Pareto fit to the upper tail (burst lengths > 10 packets, conditioned on a burst of 10 consecutive packets lost). Note that both Figures 3b and 3c have been magnified. The lower and upper tails are well-modeled with Pareto distributions of 2.84 and 0.53, respectively. While the lower tail exhibits a great deal of burstiness, the upper tail is *extremely* bursty – the fit to $\alpha = 0.53$ indicates that

Path	Entire	Lower tail	Upper tail
TC-DB	1.11	2.75	0.40
DB-TA	2.13	2.78	0.84
TA-TC	1.39	3.03	0.42
All	1.38	2.84	0.53

Table 2: Pareto parameter α fits to loss burst length distributions.

the distribution is well-modeled with an infinite mean! These results agree to some extent with those of [5], where it was found that the upper tail (though not the extreme upper tail) of TCP packet loss burst lengths are well-modeled with infinite variance, but not infinite mean. If we examine the extreme upper tail shown in Figure 3c, we find that it does not fit the Pareto distribution quite as well as the rest of the tail. This is due to the fact that our transmission runs were at most 6000 packets long, thus artificially truncating the burst distribution at that point.

The sudden nonlinearity of the loss burst distribution at 10 packets is curious. To investigate further, we examined the loss burst length distributions of each path individually. In all three cases (not shown), we found the non-linearity at about 10 packets, though it was not as striking in the case of path DB-TA, which only had loss bursts of 137 packets or less. Upon further examination, we found that all three paths were best described by fitting their lower and upper tails separately to Pareto distributions (although path DB-TA's distribution is reasonably modeled in its entirety). The result of these fits are shown in Table 2.

A phenomenological explanation of the loss burst nonlinearity is not immediately obvious, but we propose the following theory. It has been noted that modest loss bursts for a stream of packets can be caused by the drop-tail queueing mechanism that many routers employ [9]. In particular, when routers are overloaded, they will drop all packets that arrive at the tails of their busy queues until the congestion is resolved. A packet flow that passes through this router may lose a number of consecutive packets. This behavior does seem to explain the loss bursts of 1 to 10 or so packets (encompassing 30 to about 300 ms in time) that we have measured. For loss bursts beyond about 10 packets, we consistently see a different distributional character. These much longer bursts (300 ms to about 3 minutes) are likely to be due to local effects resulting from routine maintenance or unexpected outages, and/or host or router reboots.

4 Dependence and Predictability

In this section we look at what our measurements tell us about the dependence of packet loss, which manifests itself in bursts of lost packets. Similar research was performed in [2] and [6]. In [2], the unconditional loss rate and the conditional loss rate for a stream of UDP transmissions were calculated for various packet sizes. It was observed that the conditional loss rate is greater than the unconditional loss rate for all values of packet sizes. This tells us that once a packet is lost, the probability that the next packet is also lost increases, and that UDP packet loss is not well-modeled as independent.

In [6], loss of TCP packets was studied. The conclusion was that TCP packet loss is also not well-modeled as independent, but reported much higher conditional loss probabilities, which indicates more dependence. Even for small TCP ACK packets, the loss rate increased by a factor of seven if the previous ACK was lost. This research suggests that a considerable amount of UDP and TCP packet loss occurs in bursts, a result that agrees with our findings.

In this section we extend this technique for determining packet loss dependence so that we condition on the probability of finding i packets lost in the last k packet transmissions. We present our results in a *conditional loss graph* for $k = 1, ..., 8$ and summarize the conditional loss rates seen on all three paths throughout the month.

We consider a run of m transmitted packets to be a vector of (not necessarily independent) Bernoulli random variables, $\vec{x}$, where $\vec{x} = \{x_m, x_{m-1}, ..., x_1\}$ and $x_i \in \{0, 1\}$. Furthermore, let k take on values between 1 and m, and define I to be an indicator function as follows

$$I(A) \quad = \quad \begin{cases} 1 & \text{assertation } A \text{ is true} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Then,

$$b_{\vec{x},k}(i) = \sum_{l=k}^{m} I\left(\left[k - \sum_{j=l-k+1}^{l} x_j\right] = i\right) \tag{3}$$

is the number of times that i lost packets were found within k consecutive packets in the sequence $\vec{x}$. Furthermore, for $a \in \{0, 1\}$,

$$b_{\vec{x},k}(i, a) = \sum_{l=k}^{m-1} I\left(\left[k - \sum_{j=l-k+1}^{l} x_j\right] = i\right) I\left(x_{l+1} = a\right) \tag{4}$$

is the number of times that i lost packets were found within k consecutive packets in the sequence $\vec{x}$, *and were followed by* event a (where a is either a successful or lost packet). Then the probability, given the vector $\vec{x}$, of some event a following i lost packets within the last k consecutive packets is

$$P_{\vec{x},k}(i,a) = \begin{cases} \frac{b_{\vec{x},k}(i,a)}{b_{\vec{x},k}(i)} & \text{if } b_{\vec{x},k}(i) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The power of this expression is that it allows us to calculate the conditional probability of either a loss or a success given the number of losses that occurred in the last k packets. It is a generalization of the concept of conditional loss rate discussed in the last section. In fact, the expression used in [2] and [6] was simply $P_{\vec{x},1}(1,0)$.

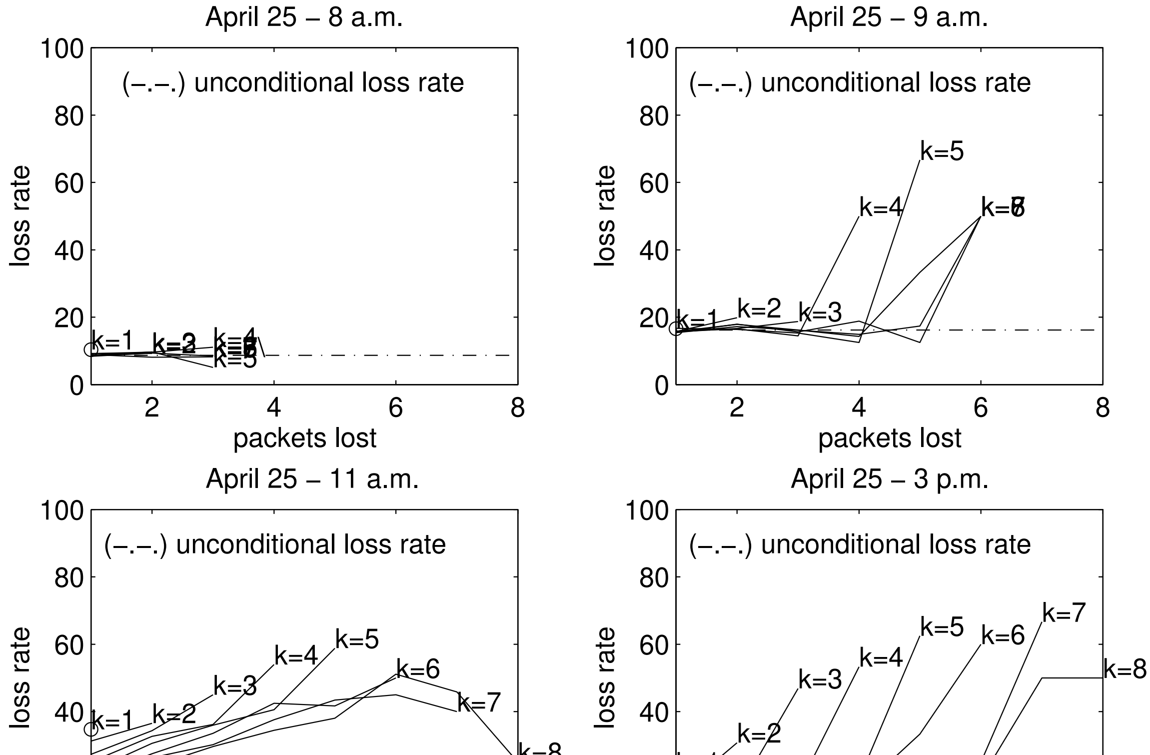

Figure 4: Conditional loss graph for four hours (runs) on April 25th, path TA-TC.

In order to visualize the dependence of packet loss, we introduce the *conditional loss graph*. On this graph we plot $P_{\vec{x},k}(i,0)$ for all $1 \leq i \leq k \leq 8$. Figure 4 shows four of these graphs, all from path TA-TC on April 25th. The unconditional packet loss rate is plotted with a dash-dot line. Consider the 11 AM graph. Looking back 1 packet ($k = 1$), we find that the probability that a loss is followed by another loss is about 35%, much greater than the unconditional loss rate of about 20%. Looking back two packets ($k = 2$), if one of those packets are lost ($i = 1$), the probability that the next packet is lost is about 32%, but if both are lost ($i = 2$), the probability that the next packet is lost becomes about 37%. The power of the conditional loss graph is in its ability to depict how the dependence of packet loss changes with k.

From Figure 4, we find that over the course of the day, from 8 AM to 3PM, the dependence of packet loss tends to grow. Even though at 8 AM we had almost 10% loss, there is very little dependence of loss. In fact, for this run, loss is probably well-modeled as independent. Note that in the 9 AM and 3 PM runs, we see the conditional loss rates spike when $i = k$. This generally indicates that much of the loss in that run came in one or more large bursts.

(δ)	**TA–TC**	**DB–TA**	**TC–DB**
Units	runs	runs	runs
> 0.900	1	6	2
> 0.800	4	22	5
> 0.700	6	64	8
> 0.600	10	115	16
> 0.500	32	137	23
> 0.400	70	195	42
> 0.300	116	257	66
> 0.200	190	314	112
> 0.100	342	405	206
> 0.090	371	422	216
> 0.080	408	434	220
> 0.070	436	442	224
> 0.060	469	453	228
> 0.050	503	464	233
> 0.040	530	469	239
> 0.030	552	482	239
> 0.020	577	488	240
> 0.010	593	493	240
> 0.005	595	496	241
> 0.0025	612	515	247
> 0.0010	675	558	289
> 0.00010	695	634	585
> 0.000010	696	634	585
> 0.0000010	696	634	585
< 0.0000010	1	48	119

Table 3: Cumulative difference δ, between the conditional and unconditional loss probabilities for each run. Cumulative number of runs in April that had a δ value of greater than the value given in the first column.

For purposes of comparison to [2] and [6], we have computed $P_{\vec{x},1}(1,0)$ for each run and then calculated the difference (δ) between $P_{\vec{x},1}(1,0)$ and the run's unconditional loss rate. Thus, δ is an indication of the dependence of packet loss. Table 3 shows, for each path, the number of times δ was greater than[4] a given value in the first column. We note that almost half of the runs exhibit a $\delta > 0.2$ which indicates strong dependence. Path DB-TA exhibits the overall greatest loss dependence even though path TA-TC has the most loss bursts. This is probably because path DB-TA's bursts were less frequent, but quite large. If we consider $\delta > 0.01$ to indicate significant dependence and $\delta \leq 0.01$ to indicate negligible dependence, we find that the majority of wide-area runs exhibit significant dependence, but the majority of metropolitan runs do not. For path TC-DB, which was the least bursty path (maximum burst length was 137), about one-third of the runs were significantly dependent. From these results it is clear that the packet loss process of the Internet exhibits a greatly varying amount of dependence.

4.1 Conditional Entropy

Returning for a moment to the dependency model of Section 4, we define the empirical conditional entropy of $\vec{x}$ as

$$H(P_{\vec{x},k}) = -\sum_{i=1}^{k} P_{\vec{x},k}(i) \sum_{a\in\{0,1\}} P_{\vec{x},k}(i,a) \log_2 P_{\vec{x},k}(i,a) \tag{6}$$

where

$$P_{\vec{x},k}(i) = \frac{b_{\vec{x},k}(i)}{m-k+1} \tag{7}$$

and 0 log 0 is, by convention, set equal to 0. $H(P_{\vec{x},k})$ gives the *conditional entropy* of a run represented by $\vec{x}$. In other words, it measures the uncertainty of whether the next packet will be lost or not, given the number of losses in the last k packets transmitted.

For an independent Bernoulli random variable X with parameter p, we define the binary entropy function as

$$H(X) = -p \log_2 p - (1-p) \log_2 (1-p) \tag{8}$$

By plotting $H(P_{\vec{x},k})$ for various runs versus $H(X)$ for $p = 0, ..., 1$, we can visually determine the uncertainty of the outcome of a packet transmission. We call this plot a *conditional entropy graph.*

Figure 5 shows the conditional entropy curve for April 25th of the TA-TC path. Each point ('+') is conditional entropy of a run. If a point is on or close to the binary entropy curve, then the patterns of

[4] The last row shows the number of times δ was less than 0.000001, indicating little or no dependence.

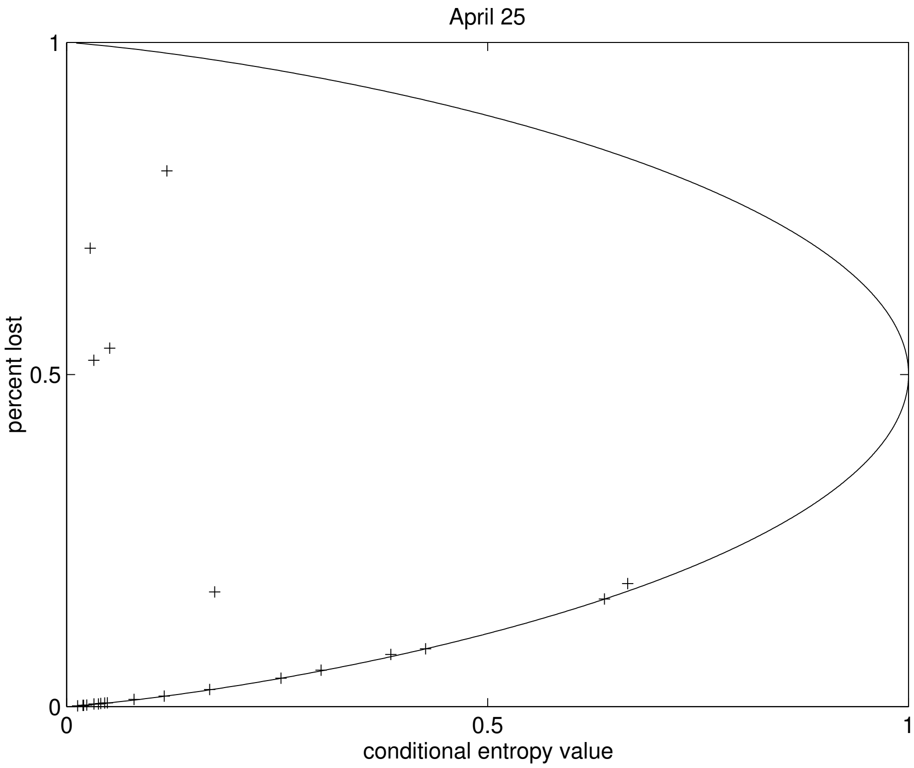

Figure 5: Conditional entropy graph for April 25th, path TA-TC.

successes and losses in that run did not make the uncertainty of the successive packets decrease. If a point is in the concave section of the graph, then the associated run exhibited less uncertainty than is warranted for its loss rate. Runs which were not on the binary entropy curve always have a significant burst of loss.

The conditional entropy graph provides us with the weighted conditional loss metric, that we discussed in the previous section. Points on the binary entropy curve may exhibit dependence, but the bursts of loss in the associated runs were relatively small compared to the number of packets transmitted.

5 Asymmetry

At this point, we direct our attention to unidirectional packet loss characteristics. In particular, we study the asymmetry of packet loss rates on a per run basis. Unidirectional packet delay is difficult to measure accurately due to the requirement of highly-synchronized clocks at both ends (see [5] for a discussion), and has traditionally been estimated as one-half of the round-trip delay of a path. However, recent research [10] has determined that unidirectional delays are highly asymmetric. Therefore, it would not be surprising if loss is asymmetric as well.

Loss asymmetry has performance implications for both TCP- and UDP-based applications. For example, most TCP transactions consist of a sender, who transmits mostly data segments, and a receiver, who transmits mostly acknowledgement (ACK) segments. If most of the loss occurs on the path from the receiver back to the sender, the sender will throttle its transmission rate, even though the path from the sender to the receiver may have more than enough capacity to handle the data segment transmissions. For a UDP-based application, such as packet telephony, asymmetric loss may result in a transaction in which one party has an acceptable quality of service while the other does not.

5.1 Asymmetry of Loss Indicator (ALI)

Recall that our software (Section 2.1) measures client-to-server loss rate (λ_c) and server-to-client loss rate (λ_s) as well as round-trip loss rate (λ_r). Loss asymmetry is present when $\lambda_s \neq \lambda_c$. It is desirable to be able to characterize this asymmetry with a single parameter. While the *presence* of asymmetry of loss over two paths can be objectively measured, the *degree* of this asymmetry is actually quite subjective. For example, consider two round-trip paths in the Internet, one between host A and host B, the other between host C and host D. Suppose that the unidirectional path AB exhibits a 20% loss rate, while path BA exhibits a 10% loss rate. Likewise, suppose that path CD exhibits a 1% loss rate and path DC exhibits a 10% loss rate. Which round-trip path has a higher degree of loss asymmetry? One may argue that round-trip path ABA is more asymmetric because the difference between the unidirectional loss rates is greater. However, one may also argue that round-trip path CDC is more asymmetric because the loss rate of path DC is 10 times that of path CD.

We feel that both of these interpretations of the degree of asymmetry are equally valid, but for purposes of analysis we chose to represent loss asymmetry with a normalized parameter which, in our opinion, captures the essentials of both arguments. We define the *Asymmetry of Loss Indicator* (ALI) as

$$ALI = |\lambda_c - \lambda_s| e^{-min(\lambda_c, \lambda_s)} \tag{9}$$

The range of values that ALI can take on is always between 0 and 1, the former representing no asymmetry while the latter represents the maximum possible asymmetry (i.e., 100% loss in one direction). The ALI contains two parts, the absolute difference of the unidirectional losses, and an exponentially-decaying weight factor. We use the absolute difference as the basis of the ALI, but it is weighted by the minimum loss rate of the two paths. Intuitively, when both loss rates are low, the ALI is roughly the absolute difference of the loss rates. However, the same difference for much higher loss rates indicates less asymmetry. We have found that, in practice, the ALI is a very good indicator of loss asymmetry (see below).

5.2 Loss Asymmetry in the Internet

In order to put the ALI to work, we have plotted unidirectional packet loss percentages of paths TC-DB for Tuesdays, Wednesdays and Thursdays of three consecutive weeks of April 1997, in Figure 8. Each subgraph has two parts – the upper section shows the unidirectional loss rates for each hourly transmission, while the lower section shows the loss asymmetry for each transmission, as represented by the ALI. The solid lines of the loss plots of upper graphs show the loss rate from the client to the server (λ_c), and the dashed lines show the loss rate from the server to the client (λ_s).

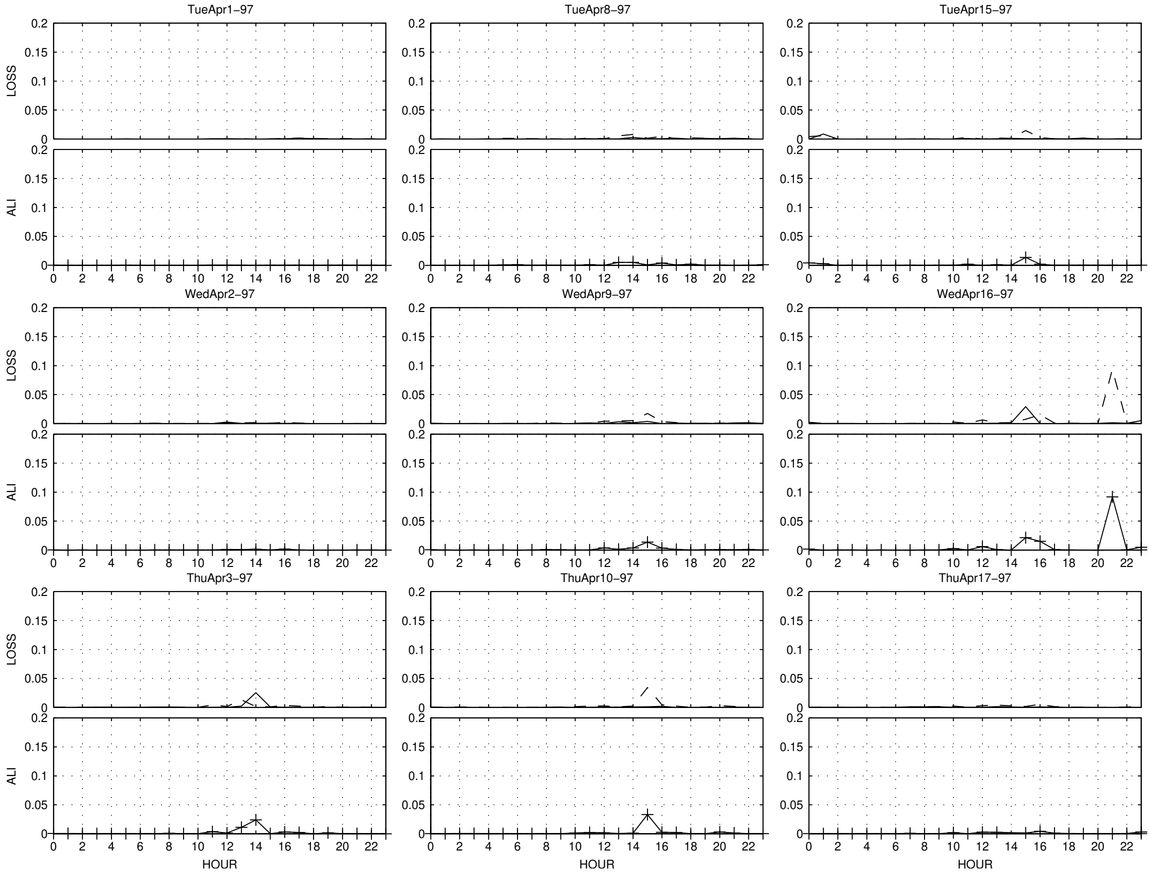

Figure 6: Daily plots of loss and ALI for path TC-DB. The solid line of the upper graphs is the client to server (λ_c) loss and the dashed line is the server to client (λ_s) loss for each hour of the days. The lower graph indicates the ALI.

From these graphs it is clear that ALI values capture the asymmetry of loss quite well. When the loss rate in one direction is negligible but the loss rate for the other direction is not, the magnitude of the ALI is that of the absolute difference between the loss rates (for an example, see April 1 of Figure 8). However, if both directions exhibit non-negligible loss, then the ALI decreases accordingly.

The distribution of ALI values for each path throughout the month is given in Table 4. For each path,

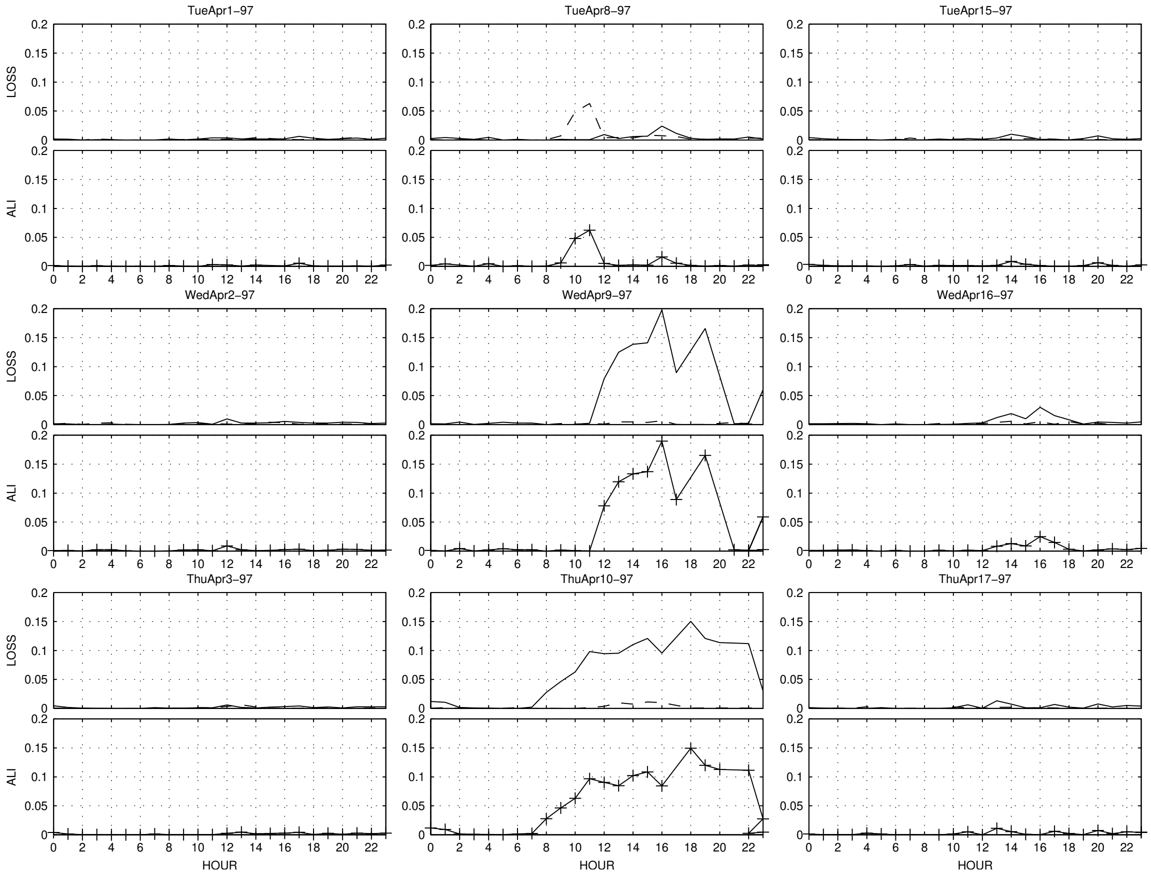

Figure 7: Daily plots of loss and ALI for path DB-TA. The solid line of the upper graphs is the client to server (λ_c) loss and the dashed line is the server to client (λ_s) loss for each hour of the days. The lower graph indicates the ALI.

the number of runs with an ALI greater than n and the number of days which had *at least one ALI* greater than n, are shown for $n = 0.01, ..., 0.15, 0.20, ..., 0.70$. For example, path TA-TC had 236 (out of 697) runs with an ALI greater than 0.01, and path DB-TA had only three days in which there was an ALI greater than 0.15.

Overall, we find that path TA-TC exhibits the greatest loss asymmetry followed by path DB-TA, and finally path TC-DB. It is not surprising that path TA-TC is the most asymmetric, since it also has the most loss. Naturally, in order for there to be loss asymmetry, there must be loss! Path TA-TC and DB-TA are both wide-area, while path TC-DB is metropolitan. We find significantly more asymmetry (and more loss) on the wide-area paths. Packet loss in general seems to follow a daily cycle – most loss occurs in working hours (9AM - 6PM), though a significant amount can occur during leisure hours (6PM - midnight). Note that Figure 8 shows only three days of the week (Tuesdays, Wednesdays, and Thursdays). However, Mondays and Fridays exhibited similar characteristics, and so are not shown. We found that there was

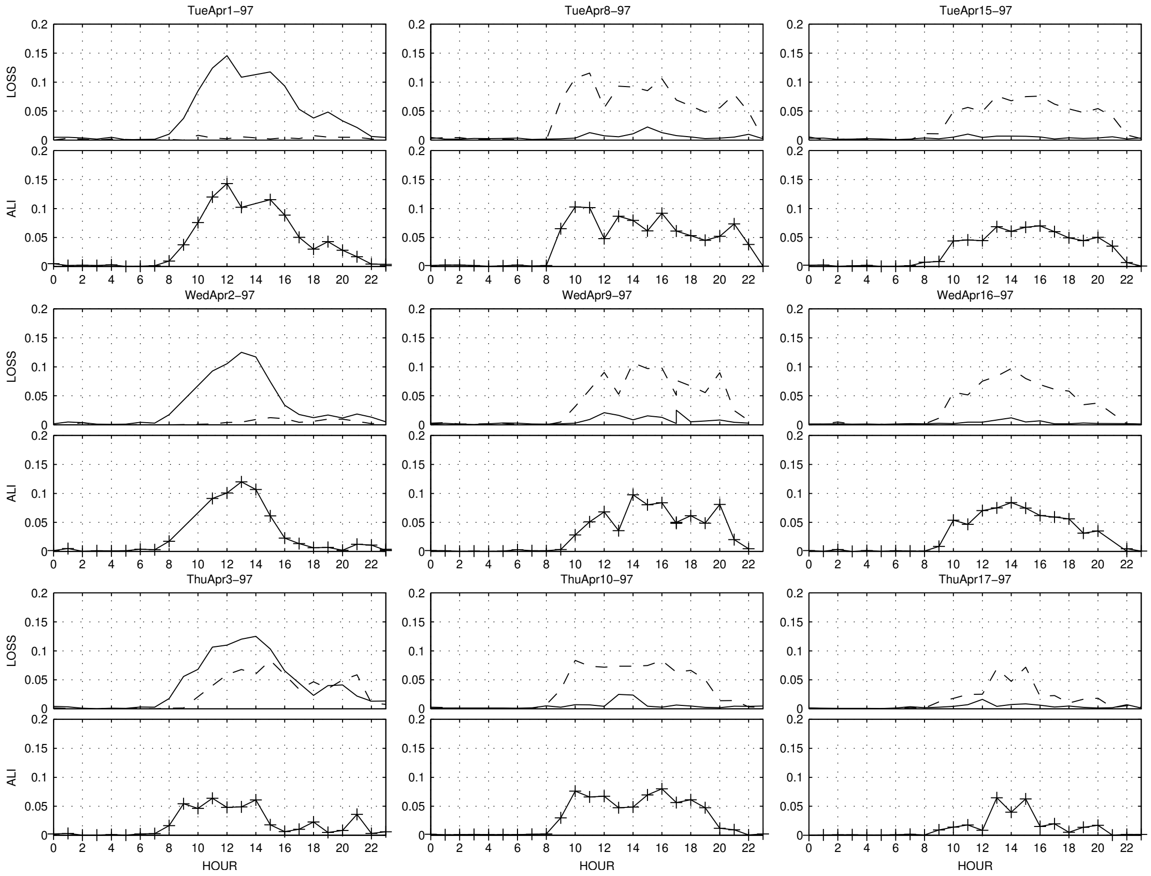

Figure 8: Daily plots of loss and ALI for path TA-TC. The solid line of the upper graphs is the client to server (λ_c) loss and the dashed line is the server to client (λ_s) loss for each hour of the days. The lower graph indicates the ALI.

very little loss on our three paths during the weekends, so to save space, we did not include weekends in our graphs.

5.3 Causes of Asymmetry

Packet loss asymmetry may have any number of causes, including differing loads on client and server hosts and LANs, and asymmetric router queue congestion. However, a major cause of loss asymmetry is that, for many round-trip paths in the Internet, the path from the client to the server and from the server to the client are different. This routing asymmetry was explored in [4].

In our experiments, the wide-area paths exhibited considerably more loss asymmetry than the metropolitan-area path. Using the traceroute tool, we found that the dominant route from `tcres.cs.depaul.edu` to `tabasco.cs.ucdavis.edu` was very different from the return route. Tables 5 and 6 show these two routes. DePaul's wide-area Internet traffic is passed to MCI (through local provider CICNet) and U. C. Davis's

ALI	**TC–DB**		**DB–TA**		**TA–TC**	
Units	runs	days	runs	days	runs	days
> 0.01	14	10	53	16	236	25
> 0.02	6	6	45	12	201	25
> 0.03	4	4	37	10	178	25
> 0.04	4	4	35	8	151	22
> 0.05	3	3	28	7	111	21
> 0.06	3	3	27	7	87	19
> 0.07	2	2	27	7	72	18
> 0.08	2	2	22	7	55	14
> 0.09	1	1	19	6	47	12
> 0.10	1	1	15	5	39	10
> 0.11	1	1	9	4	36	10
> 0.12	1	1	7	4	31	8
> 0.13	1	1	6	4	30	8
> 0.14	1	1	5	4	29	7
> 0.15	1	1	4	3	27	7
> 0.20	1	1	1	1	19	5
> 0.30	0	0	1	1	14	4
> 0.40	0	0	1	1	10	4
> 0.50	0	0	1	1	3	2
> 0.60	0	0	1	1	2	2
> 0.70	0	0	1	1	0	0

Table 4: The cumulative number of days and runs for which the *ALI* was greater than a given value in the first column.

wide-area traffic is passed to Sprint (via University of California Operations). On the path from `tcres` to `tabasco`, MCI routes the packet immediately to Sprint (at the Chicago NAP) which then carries them across the country, while on the path from `tabasco` to `tcres`, Sprint routes the packets quickly to MCI (at the San Francisco NAP), which then carries them across the country.

This behavior, known as *shortest-exit* or *hot-potato* routing, is done by many national backbone providers in order to minimize the amount of traffic carried by their networks. Since the east-to-west and west-to-east packets were carried by different networks with different capacities, policies and administration, it is not surprising that the packet loss characteristics for these wide-area paths exhibit asymmetry.

Although we were unable to run traceroute from `dbis`, the fact that UIC and DePaul both share CICNet as a local provider indicates that they are likely to have very similar routes to `tabasco`. This also implies that two autonomous systems sharing a single ISP may suffer less packet loss and less packet loss asymmetry because that ISP is able to perform capacity planning and management at all points between

Hop	Host or router
1	140.192.32.1
2	cicrtr.depaul.edu
3	dgc-dep.chicago.cic.net
4	dgf-fddi5-0.chicago.cic.net
5	bordercore3-hssi0-0.WillowSprings.mci.net
6	core4.WillowSprings.mci.net
7	sl-chi-15-H1/0-T3.sprintlink.net
8	sl-chi-1-P0/0/0-155M.sprintlink.net
9	sl-stk-23-H5/0-T3.sprintlink.net
10	sl-stk-16-F0/0.sprintlink.net
11	sl-ucberkeley-1-H1/0-T3.sprintlink.net
12	dgty2.ucop.edu
13	128.120.249.1
14	128.120.252.1
15	eu2-gw.ucdavis.edu
16	* * *
17	* * * (tabasco.cs.ucdavis.edu)

Table 5: Dominant route from `tcres` to `tabasco`.

the two systems. Such capacity planning would be much more difficult to manage across two or more ISPs.

6 Discussion and Conclusion

The implications of this research for practitioners wishing to develop Internet telephony applications, and network engineers who want to enable such applications, is twofold. The extreme heavy-tailed behavior of packet loss indicates that while long bursts of loss are not common, their appearance can be catastrophic. Telephony customers are often willing to tolerate the slightly clipped speech and static resulting from loss bursts of one to four or so G.723.1 packets. However, outages of a few seconds or more will probably not be tolerated. From the user's point of view, this would be as bad as circuit-switched connection being suddenly dropped. Since loss burst distributions observed in this paper seem to be inherent to the Internet architecture, carriers providing voice over IP may have to build fault tolerance or loss-smoothing mechanisms (such as Random Early Detection [11]) into their backbones.

Our measurements of loss asymmetry indicate that customers at either end of an interactive voice application may observe radically different qualities of service. Thus, the overall QoS for a bi-directional call will be the lower QoS of the two unidirectional paths, as the user who is receiving the higher QoS may be asked to repeat herself often. The fact that loss occurs asymmetrically indicates a certain locality of packet loss in the Internet. If we look more carefully at a unidirectional path we may find that a large portion of its overall loss rate is caused by a small number of routers. Since a single autonomous network

Hop	Host or router
1	ph-254-subnet56.cs.ucdavis.edu
2	eu2-gw.ucdavis.edu
3	area-gw.ucdavis.edu
4	border-gw.ucdavis.edu
5	dgty2.ucdavis.edu
6	bgty2.ucop.edu
7	sl-stk-16-H11/0-T3.sprintlink.net
8	144.228.40.22
9	sl-stk-1-P10/0/0-155M.sprintlink.net
10	core4-hssi1-0.SanFrancisco.mci.net
11	bordercore3-loopback.WillowSprings.mci.net
12	cicnet.WillowSprings.mci.net
13	dgc-fddi5-0.chicago.cic.net
14	dep-dgc.chicago.cic.net
15	140.192.1.120
16	tcres.cs.depaul.edu

Table 6: Dominant route from `tabasco` to `tcres`.

can upgrade its facilities to reduce loss, localized packet loss is likely to occur at peering points between two local or backbone service providers. If this is indeed a major cause of Internet packet loss, then reducing this loss locality may require a business solution rather than a technical solution.

In this paper, we have described a robust methodology for collecting and analyzing Internet packet loss statistics. In particular, we find that packet loss is highly bursty, with a relatively small number of bursts accounting for most of the loss rate. We have also shown that packet loss exhibits a varying amount of dependence. Finally, we have shown that packet loss is highly asymmetric on two wide-area paths, and we have introduced a normalized metric to measure the asymmetry of loss.

References

[1] D. Sanghi, A. K. Agrawala, O. Gudmundsson, and B. N. Jain, "Experimental assessment of end-to-end behavior on Internet," in *Proceedings, IEEE INFOCOM '93*, pp. 867–874, Mar. 1993.

[2] J.-C. Bolot, "Characterizing end-to-end packet delay and loss in the Internet," *Journal of High Speed Networks*, vol. 2, pp. 305–323, 1993.

[3] A. Mukherjee, "On the dynamics and significance of low frequency components of Internet load," *Internetworking: Research and Experience*, vol. 5, pp. 163–205, 1994.

[4] V. Paxson, "End-to-end routing behavior in the Internet," in *Proceedings, ACM SIGCOMM '96*, Aug. 1996.

[5] V. Paxson, "End-to-end Internet packet dynamics," in *Proceedings, ACM SIGCOMM '97*, Sept. 1997.

[6] V. Paxson, *Measurements and Analysis of End-to-End Internet Dynamics*. PhD thesis, University of California, Berkeley, Apr. 1997.

[7] W. R. Stevens, *TCP/IP Illustrated*, vol. I. Addison-Wesley, 1994.

[8] International Telecommunication Union, "Recommendation G.723.1," 1996. http://www.itu.int.

[9] S. Floyd and V. Jacobson, "On traffic phase-effects in packet-switched gateways," *Internetworking: Research and Experience*, vol. 3, no. 3, 1992.

[10] K. C. Claffy, G. C. Polyzos, and H.-W. Braun, "Measurement considerations for assessing unidirectional latencies," *Internetworking: Research and Experience*, vol. 4, no. 3, pp. 121–132, 1993.

[11] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397–413, Aug. 1993.