Improving Accuracy in End-to-end Packet Loss Measurement

Joel Sommers
University of Wisconsin-Madison
jsommers@cs.wisc.edu

Paul Barford
University of Wisconsin-Madison
pb@cs.wisc.edu

Nick Duffield
AT&T Labs-Research
duffield@research.att.com

Amos Ron
University of Wisconsin-Madison
amos@cs.wisc.edu

ABSTRACT

Measurement and estimation of packet loss characteristics are challenging due to the relatively rare occurrence and typically short duration of packet loss episodes. While active probe tools are commonly used to measure packet loss on end-to-end paths, there has been little analysis of the accuracy of these tools or their impact on the network. The objective of our study is to understand how to measure packet loss episodes accurately with end-to-end probes. We begin by testing the capability of standard Poisson-modulated end-to-end measurements of loss in a controlled laboratory environment using IP routers and commodity end hosts. Our tests show that loss characteristics reported from such Poisson-modulated probe tools can be quite inaccurate over a range of traffic conditions. Motivated by these observations, we introduce a new algorithm for packet loss measurement that is designed to overcome the deficiencies in standard Poisson-based tools. Specifically, our method creates a probe process that (1) enables an explicit trade-off between accuracy and impact on the network, and (2) enables more accurate measurements than standard Poisson probing at the same rate. We evaluate the capabilities of our methodology experimentally by developing and implementing a prototype tool, called BADABING. The experiments demonstrate the trade-offs between impact on the network and measurement accuracy. We show that BADABING reports loss characteristics far more accurately than traditional loss measurement tools.

Categories and Subject Descriptors: C.2.3 [Network Operations]: Network monitoring, C.2.5 [Local and Wide-Area Networks]: Internet (e.g., TCP/IP), C.4 [Performance of Systems]: Measurement Techniques

General Terms: Algorithms, Design, Experimentation, Measurement

Keywords: Active Measurement, BADABING, Network Congestion, Network Probes, Packet Loss

SIGCOMM'05, August 21–26, 2005, Philadelphia, Pennsylvania, USA.
Copyright 2005 ACM 1-59593-009-4/05/0008 ...$5.00.

1. INTRODUCTION

Measuring and analyzing network traffic dynamics between end hosts has provided the foundation for the development of many different network protocols and systems. Of particular importance is understanding packet loss behavior since loss can have a significant impact on the performance of both TCP- and UDP-based applications. Despite efforts of network engineers and operators to limit loss, it will probably never be eliminated due to the intrinsic dynamics and scaling properties of traffic in packet switched network [19]. Network operators have the ability to passively monitor nodes within their network for packet loss on routers using SNMP. End-to-end active measurements using probes provide an equally valuable perspective since they indicate the conditions that application traffic is experiencing on those paths.

The most commonly used tools for probing end-to-end paths to measure packet loss resemble the ubiquitous PING utility. PING-like tools send probe packets (*e.g.,* ICMP echo packets) to a target host at fixed intervals. Loss is inferred by the sender if the response packets expected from the target host are not received within a specified time period. Generally speaking, an active measurement approach is problematic because of the discrete *sampling* nature of the probe process. Thus, the accuracy of the resulting measurements depends both on the characteristics and interpretation of the sampling process as well as the characteristics of the underlying loss process.

Despite their widespread use, there is almost no mention in the literature of how to tune and calibrate [28] active measurements of packet loss to improve accuracy or how to best interpret the resulting measurements. One approach is suggested by the well-known PASTA principle [36] which, in a networking context, tells us that Poisson-modulated probes will provide unbiased time average measurements of a router queue's state. This idea has been suggested as a foundation for active measurement of end-to-end delay and loss [3]. However, the asymptotic nature of PASTA means that when it is applied in practice, the higher moments of measurements must be considered to determine the validity of the reported results. A closely related issue is the fact that loss is typically a rare event in the Internet [39]. This reality implies either that measurements must be taken over a long time period, or that average rates of Poisson-modulated probes may have to be quite high in order to report accurate estimates in a timely fashion. However, increasing the mean probe rate may lead to the situation that the probes themselves skew the results. Thus, there are trade-offs in packet loss measurements between probe rate, measurement accuracy, impact on the path and timeliness of results.

The goal of our study is to understand how to accurately measure loss characteristics on end-to-end paths with probes. We are interested in two specific characteristics of packet loss: *loss episode frequency*, and *loss episode duration* [39]. The commonly referred to notion of *loss rate* [8, 27] can be estimated directly from these two measurements. Our study consists of three parts: (*i*) empirical evaluation of the currently prevailing approach, (*ii*) development of estimation techniques that are based on novel experimental design, novel probing techniques, and simple validation tests, and (*iii*) empirical evaluation of this new methodology.

We begin by testing standard Poisson-modulated probing in a controlled and carefully instrumented laboratory environment consisting of commodity workstations separated by a series of IP routers. Background traffic is sent between end hosts at different levels of intensity to generate loss episodes thereby enabling repeatable tests over a range of conditions. We consider this setting to be ideal for testing loss measurement tools since it combines the advantages of traditional simulation environments with those of tests in the wide area. Namely, much like simulation, it provides for a high level of control and an ability to compare results with "ground truth." Furthermore, much like tests in the wide area, it provides an ability to consider loss processes in actual router buffers and queues, and the behavior of *implementations* of the tools on commodity end hosts. Our tests reveal two important deficiencies with simple Poisson probing. First, individual probes often incorrectly report the absence of a loss episode (*i.e.,* they are successfully transferred when a loss episode is underway). Second, they are not well suited to measure loss episode duration over limited measurement periods.

Our observations about the weaknesses in standard Poisson probing motivate the second part of our study: the development of a different approach for end-to-end loss measurement. There are three key elements in this new approach. First, we design a probe process that assesses the likelihood of loss experienced by other flows that use the same path, rather than merely reporting its own packet losses. Second, we design a new experimental framework with estimation techniques that directly estimate the mean duration of the loss episodes without estimating the duration of any individual loss episode. Our estimators are proved to be consistent, under mild assumptions of the probing process. Third, we provide simple validation tests (that require no additional experimentation or data collection) for some of the statistical assumptions that underly our analysis.

The third part of our study involves the empirical evaluation of our new loss measurement methodology. To this end, we developed a one-way active measurement tool called BADABING. BADABING sends fixed-size probes at specified intervals from one measurement host to a collaborating target host. The target system collects the probe packets and reports the loss characteristics after a specified period of time. We also compare BADABING with a standard tool for loss measurement that emits probe packets at Poisson intervals. The results show that our tool reports loss episode estimates much more accurately for the same number of probes.

The most important implication of these results is that there is now a methodology and tool available for wide-area studies of packet loss characteristics that enables researchers to understand and specify the trade-offs between accuracy and impact. Furthermore, the tool is self-calibrating [28] in the sense that it can report when estimates are poor. Practical applications could include its use for path selection in peer-to-peer overlay networks and as a tool for network operators to monitor specific segments of their infrastructures.

The remainder of the paper is structured as follows. In Section 2, we consider related work in the areas of loss measurement techniques and loss measurement studies. In Section 3, we provide definitions for loss episode frequency and duration which provide critical context for our probe process. Our laboratory testbed configuration and results of our tests of basic Poisson-modulated probing are reported in Section 4. Details on our new probe process and an analytic treatment of the validity of the resulting loss estimators is provided in Section 5. A description of the tool that we built to implement our probe process and results of its evaluation experiments are described in Section 6. We provide a discussion of practical considerations for using our tool in Section 7. A concluding summary and future directions for this work are provided in Section 8.

2. RELATED WORK

It is well known that packet loss can have a substantial impact on the performance of a wide range of Internet protocols and applications. Understanding the characteristics and impact of packet loss led, for example, to the development of the NewReno [15] and SACK [21] versions of TCP. Loss characteristics are also a fundamental component of TCP throughput modeling [10, 22, 24].

There have been many studies of packet loss behavior in the Internet. Bolot [8] and Paxson [27] evaluated end-to-end probe measurements and reported characteristics of packet loss over a selection of paths in the wide area. Yajnik *et al.* evaluated packet loss correlations on longer time scales and developed Markov models for temporal dependence structures [37]. Zhang *et al.* characterized several aspects of packet loss behavior in [39]. In particular, that work reported measures of *constancy* of loss episode rate, loss episode duration, loss free period duration and overall loss rates. The authors in [25] used a sophisticated passive monitoring infrastructure inside Sprint's IP backbone to gather packet traces and analyze loss episodes frequency and duration. Finally, Sommers and Barford pointed out some of the limitations in standard end-to-end Poisson probing tools by comparing the loss rates measured by such tools to loss rates measured by passive means in a fully instrumented wide area infrastructure [6].

The foundation for the notion that Poisson Arrivals See Time Averages (PASTA) was developed by Brumelle in [9], and later formalized by Wolff in [36]. Adaptation of those queuing theory ideas into a network probe context to measure loss and delay characteristic began with Bolot's study in [8] and was extended by Paxson in [27]. Of particular relevance to our work is Paxson's recommendation and use of Poisson-modulated active probe streams to reduce bias in delay and loss measurements. Several studies include the use of loss measurements to estimate network properties such as bottleneck buffer size and cross traffic intensity [4, 29] . The Internet Performance Measurement and Analysis efforts [16, 17] resulted in a series of RFCs that specify how packet loss measurements should be conducted. However, those RFCs are devoid of details on how to tune probe processes and how to interpret the resulting measurements. We are also guided by Paxson's recent work in [28] where he advocates rigorous calibration of network measurement tools.

ZING is a tool for measuring end-to-end packet loss in one direction between two participating end hosts [2, 20]. ZING sends UDP packets at Poisson-modulated intervals with fixed mean rate. Savage developed the STING [30] tool to measure loss rates in both forward and reverse directions from a single host. STING uses a clever scheme for manipulating a TCP stream to measure loss. Allman *et al.* demonstrated how to estimate TCP loss rates from passive packet traces of TCP transfers taken close to the sender. A related study using passive packet traces taken in the middle of the network was presented in [7]. Network tomography based on using

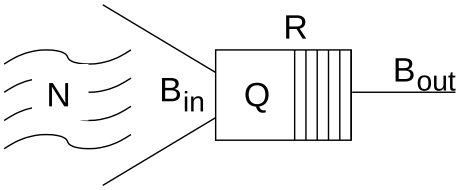


Figure 1: Simple model for system under consideration. N **flows on input links with aggregate bandwidth** B_{in} **compete for a single output link on router** R **with bandwidth** B_{out} **where** $B_{in} > B_{out}$**. The output link has** Q **seconds of buffer capacity.**

both multicast and unicast probes has also been demonstrated to be effective for inferring loss rates on internal links on end-to-end paths [11, 12].

Finally, controlled laboratory environments like the one used in this paper have begun to emerge as effective arenas for network protocol and system evaluation (*e.g.,* [18]). Environments such as WAIL [1], DETER [32], and Emulab [35] are openly available to the research community, often include IP routers as well as general purpose workstations, and are ideal for measurement tool testing.

3. DEFINITIONS OF LOSS CHARACTERISTICS

There are many factors that can contribute to packet loss in the Internet. We will describe some of these in detail as a foundation for understanding our active measurement objectives. The environment that we consider is modeled as a set of N flows that pass through a router R and compete for a single output link with bandwidth B_{out} as depicted in Figure 1. The aggregate input bandwidth (B_{in}) must be greater than the shared output link (B_{out}) in order for loss to take place. The typical round trip time for the N flows is M seconds. Router R is configured with Q bytes of packet buffers to accommodate traffic bursts, with Q typically sized on the order of $M \times B$ [5, 34]. We assume that the traffic includes a mixture of short- and long-lived TCP flows as is common in today's Internet, and that the value of N will fluctuate over time.

Figure 2 is an illustration of how the occupancy of the buffer in router R might evolve over time. Congestion occurs when the aggregate sending rate of the N flows exceeds the capacity of the shared output link. The onset of congestion results in filling of the output buffer which is seen as a positive slope in queue length graph. The rate of increase of the queue length depends both on the number N and on sending rate of each source. A *loss episode* begins when the aggregate sending rate has exceeded B_{out} for a period of time sufficient to load Q bytes into the output buffer of router R (*e.g.,* at times a and c in Figure 2). A loss episode ends when the aggregate sending rate drops below B_{out} and the buffer begins a consistent drain down to zero (*e.g.,* at times b and d in Figure 2). This typically happens when TCP sources sense a packet loss and halve their sending rate, or simply when the number of competing flows N drops to a sufficient level. In the former case, the duration of a loss episode is related to M, depending whether loss is sensed by a timeout or fast retransmit signal. We define *loss episode duration* as the difference between start and end times (*i.e.,* $b-a$ and $d-c$). While this definition and model for loss episodes is somewhat simplistic and dependent on well behaved TCP flows, it is important for any measurement method to be robust to flows that do not react to congestion in a TCP-friendly fashion.

This definition of loss episodes can be considered a "router-centric" view since it says nothing about when any one end-to-end flow actually loses a packet or senses a lost packet. This contrasts with most of the prior work discussed in Section 2 which consider only losses of individual or groups of probe packets. In other words, in our methodology, a loss episode begins when the probability of some packet loss becomes positive. During the episode, there might be transient periods during which packet loss ceases to occur, followed by resumption of some packet loss. The episode ends when the probability of packet loss stays at 0 for a sufficient period of time (longer than typical RTT). Thus, we offer two definitions for *packet loss rate*:

- **Router-centric loss rate.** With L the number of dropped packets on a given output link on router R during a given period of time, and S the number all successfully transmitted packets through the same link over the same period of time, we define the router-centric loss rate as $L/(S+L)$.
- **End-to-end loss rate.** We define end-to-end loss rate in exactly the same manner as router-centric loss-rate, with the caveat that we only count packets that belong to a specific flow on interest.

It is important to distinguish between these two notions of loss rates since packets are transmitted at the maximum rate B_{out} during loss episodes. The result is that during a period where the router-centric loss rate is non-zero, there may be flows that do not lose any packets and therefore have end-to-end loss rates of zero. This observation is central to our study and bears directly on the design and implementation of active measurement methods for packet loss.

4. SIMPLE POISSON PROBING FOR PACKET LOSS

We begin by evaluating the capabilities of simple Poisson-modulated loss probe measurements using the ZING tool [2, 20]. We tested ZING in a series of experiments conducted in a laboratory environment consisting of commodity workstation end hosts and a series of IP routers. We consider this to be an environment ideally suited to understanding and calibrating end-to-end loss measurement tools. Laboratory environments do not have the weaknesses typically associated with ns-type simulation (*e.g.,* abstractions of measurement tools, protocols and systems) [13], nor do they have the weaknesses of wide area *in situ* experiments (*e.g.,* lack of control, repeatability, and complete, high fidelity end-to-end instrumentation). We address the important issue of testing the tool under "representative" traffic conditions by using a combination of the Harpoon IP traffic generator [31] and Iperf [33] to evaluate the tool over a range of cross traffic and loss conditions.

4.1 Testbed Setup

The laboratory testbed used in our experiments is shown in Figure 3. It consisted of commodity end hosts connected to a dumbbell-like topology comprised of Cisco GSR 12000 routers. Both probe and background traffic was generated and received by the end hosts. Traffic flowed from the sending hosts on separate paths via Gigabit Ethernet to separate Cisco GSRs (hop B in the figure) where it transitioned to OC12 (622 Mb/s) links. This configuration was created in order to accommodate our measurement system, described below. Probe and background traffic was then multiplexed onto a single OC3 (155 Mb/s) link (hop C in the figure) which formed the bottleneck where loss episodes took place. We used a hardware-based propagation delay emulator on the OC3 link to add 50 milliseconds delay in each direction for all experiments, and configured the bottleneck queue to hold approximately 100 milliseconds

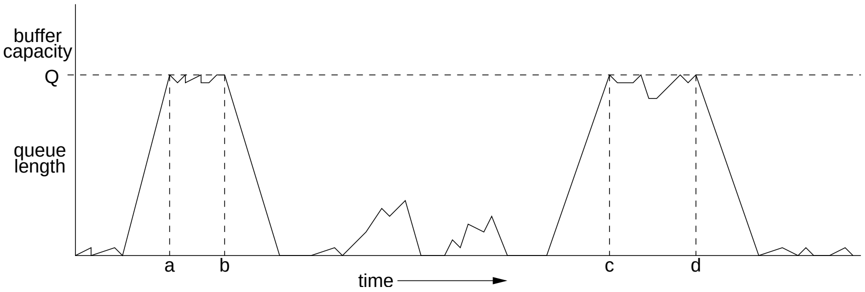


Figure 2: Example of the evolution of the length of a queue over time. The queue length grows when aggregate demand exceeds the capacity of the output link. Loss episodes begin (points a and c) when the maximum buffer size Q is exceeded. Loss episodes end (points b and d) when aggregate demand falls below the capacity of the output link and the queue drains to zero.

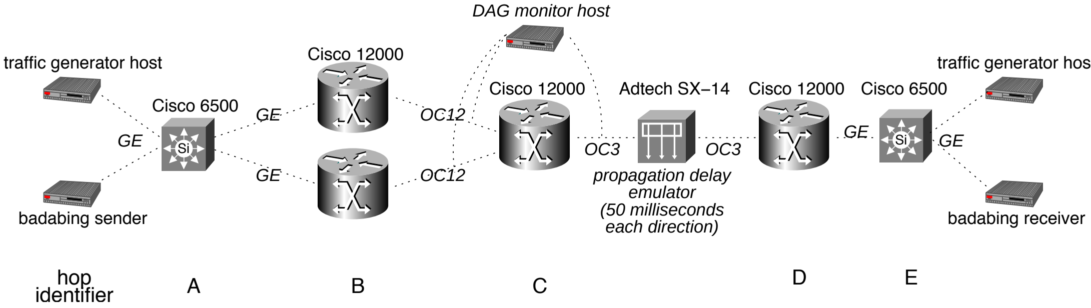


Figure 3: Experimental testbed. Cross traffic scenarios consisted of constant bit-rate traffic, long-lived TCP flows, and Harpoon web-like traffic. Cross traffic flowed across one of two routers at hop B, while probe traffic flowed through the other. Optical splitters connected Endace DAG 3.5 and 3.8 passive packet capture cards to the testbed between hops B and C, and hops C and D. Probe traffic flowed from left to right and the loss episodes occurred at hop C.

of packets. Packets exited the OC3 link via another Cisco GSR 12000 (hop D in the figure) and passed to receiving hosts via Gigabit Ethernet.

The probe and traffic generator hosts consisted of identically configured workstations running Linux 2.4. The workstations had 2 GHz Intel Pentium 4 processors with 2 GB of RAM and Intel Pro/1000 network cards. They were also dual-homed, so that all management traffic was on a separate network than depicted in Figure 3.

One of the most important aspects of our testbed was the measurement system we used to establish "ground truth" for our experiments. Optical splitters were attached to both the ingress and egress links at hop C and Endace DAG 3.5 and 3.8 passive monitoring cards were used to capture traces of packets entering and leaving the bottleneck node. DAG cards have been used extensively in many other studies to capture high fidelity packet traces in live environments (*e.g.,* they are deployed in Sprint's backbone [14] and in the NLANR infrastructure [23]). By comparing packet header information, we were able to identify exactly which packets were lost at the congested output queue during experiments. Furthermore, the fact that the measurements of packets entering and leaving hop C were time-synchronized on the order of a single microsecond enabled us to easily infer the queue length and how the queue was affected by probe traffic during all tests.

4.2 Performance of Poisson Probes

ZING is a tool for measuring packet delay and loss in one direction on an end-to-end path. The ZING sender emits UDP probe packets at Poisson-modulated intervals with timestamps and unique sequence numbers and the receiver logs the probe packet arrivals. Users specify the mean probe rate λ, the probe packet size, and the number of packets in a "flight."

To evaluate simple Poisson probing, we configured ZING using the same parameters as in [39]. Namely, we ran two tests, one with $\lambda = 100ms$ (10 Hz) and 256 byte payloads and another with $\lambda = 50ms$ (20Hz) and 64 byte payloads. To determine the duration of our experiments below, we selected a period of time that should limit the variance of the loss rate estimator $\bar{X}$ where $Var(\bar{X_n}) \approx \frac{p}{n}$ for loss rate p and number of probes n.

We conducted three separate experiments in our evaluation of simple Poisson probing. In each test we measured both the frequency and duration of packet loss episodes. Again, we used the definition in [39] for loss episode, namely, "a series of consecutive packets (possibly only of length one) that were lost." The first experiment uses 40 infinite TCP sources with receive windows set to 256 full size (1500 bytes) packets. Figure 4 shows the time series of the queue occupancy for a portion of the experiment; the expected synchronization behavior of TCP sources in congestion avoidance is clear. The experiment was run for a period of 15 minutes which should have enabled ZING to measure loss rate with standard deviation within 10% of the mean.

Results from the experiment with infinite TCP sources are shown in Table 1. The table shows that ZING performs poorly in measuring both loss frequency and duration in this scenario. For both probe rates, there were no instances of consecutive lost packets, which explains the inability to estimate loss episode duration.

In the second set of experiments, we used Iperf to create a series of (approximately) constant duration (about 68 milliseconds) loss episodes that were spaced randomly at exponential intervals with mean of 10 seconds over a 15 minute period. The time series of the queue length for a portion of the test period is shown in Figure 5.

Results from the experiment with randomly spaced, constant duration loss episodes are shown in Table 2. The table shows that ZING measures loss frequencies and durations that are closer to the true values.

In the final set of experiments, we used Harpoon to create a series of loss episodes that approximate loss resulting from web-like traffic. Harpoon was configured to briefly increase its load in order to induce packet loss, on average, every 20 seconds. The variability of traffic produced by Harpoon complicates delineation of loss episodes. To establish baseline loss episodes to compare against, we found trace segments where the first and last events were packet losses, and queuing delays of all packets between those losses were above 90 milliseconds (within 10 milliseconds of the maximum). We ran this test for 15 minutes and a portion of the time series for the queue length is shown in Figure 6.

Results from the experiment with Harpoon web-like traffic are shown in Table 3. For measuring loss frequency, neither probe rate results in a close match to the true frequency. For loss episode duration, the results are also poor. For the 10 Hz probe rate, there were no consecutive losses measured, and for the 20 Hz probe rate, there were only two instances of consecutive losses, each of exactly two lost packets.

Table 1: Results from ZING experiments with infinite TCP sources.

	frequency	duration μ (σ) (seconds)
true values	0.0265	0.136 (0.009)
ZING (10Hz)	0.0005	0 (0)
ZING (20Hz)	0.0002	0 (0)

Table 2: Results from ZING experiments with randomly spaced, constant duration loss episodes.

	frequency	duration μ (σ) (seconds)
true values	0.0069	0.068 (0.000)
ZING (10Hz)	0.0036	0.043 (0.001)
ZING (20Hz)	0.0031	0.050 (0.002)

Table 3: Results from ZING experiments with Harpoon web-like traffic.

	frequency	duration μ (σ) (seconds)
true values	0.0093	0.136 (0.009)
ZING (10Hz)	0.0014	0 (0)
ZING (20Hz)	0.0012	0.022 (0.001)

5. PROBE PROCESS MODEL

The results from our experiments described in the previous section show that simple Poisson probing is generally poor for measuring loss episode frequency and loss episode duration. These results, along with deeper investigation of the reasons for particular deficiencies in loss episode duration measurement, form the foundation for a new measurement process.

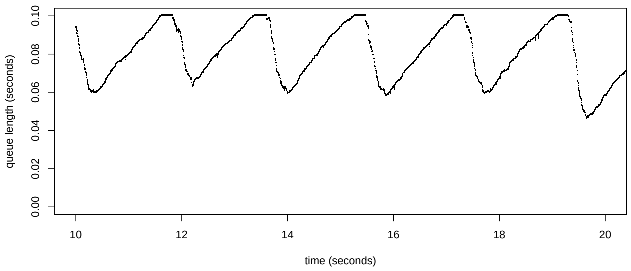


Figure 4: Queue length time series for a portion of the experiment with 40 infinite TCP sources.

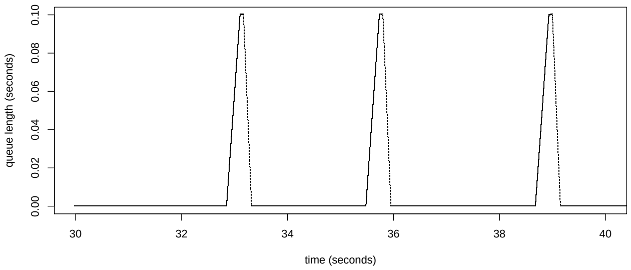


Figure 5: Queue length time series for a portion of the experiment with randomly spaced, constant duration loss episodes.

5.1 General Setup

Our methodology involves dispatching a sequence of probes, each of which contains one or more very closely spaced packets. The aim of a probe is to get a snapshot of the congestion state of the network at the instant of probing. To this end, the record for each probe indicates whether or not it encountered congestion, as indicated by either the loss or sufficient delay of any of the packets within a probe, as described in § 6. The reason for using multi-packet probes is that not all packets passing through a congested link are subject to loss; using multiple packets enables a more accurate determination to be made.

The probes themselves are organized into what we term *basic experiments*, each of which comprises a number of probes sent in

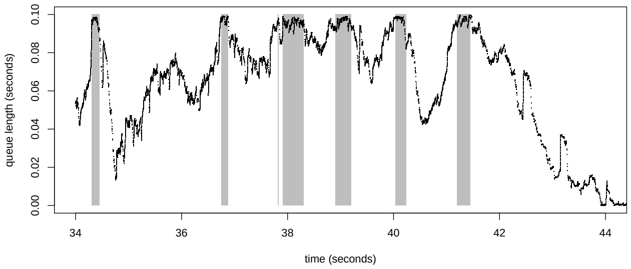


Figure 6: Queue length time series for a portion of the experiment with Harpoon web-like traffic. Time segments in grey indicate loss episodes.

rapid succession. The aim of the basic experiment is to determine the dynamics of transitions between the congested and uncongested state of the network. Below we show how this enables us to estimate the duration of congestion periods.

A *full experiment* comprises a sequence of basic experiments generated according to some rule. The sequence may be terminated after some specified number of basic experiments, or after a given duration, or in an open-ended adaptive fashion, *e.g.,* until estimates of desired accuracy for a congestion characteristic have been obtained, or until such accuracy is determined impossible.

We formulate the probe process as a discrete-time process. This decision is not a fundamental limitation: since we are concerned with measuring congestion dynamics, we need only ensure that the interval between the discrete time slots is smaller than the time scales of the congested episodes. A *congested slot* is simply a time slot during which congestion occurs. A *congestion episode* is a maximal set of consecutive slots that are congested.

There are three steps in the explanation of our loss measurement method (*i.e.,* the experimental design and the subsequent estimation). First, we present the *basic algorithm* version of our design. This model is designed to provide estimators of the frequency of congested slots and the duration of congestion episodes. The frequency estimator is unbiased, and under relatively weak statistical assumptions, both estimators are consistent in the sense they converges to their respective true values as the number of measurements grows.

Second, we describe the *improved algorithm* version of our design which provides loss episode estimators under weaker assumptions, and requires that we employ a more sophisticated experimental design. In this version of the model, we insert a mechanism to estimate, and thereby correct the possible bias of the estimators from the basic design.

Third, we describe simple validation techniques that can be used to assign a level of confidence to loss episode estimates. This enables open-ended experimentation with a stopping criterion based on estimators reaching a requisite level of confidence.

5.2 Basic Algorithm

For each time slot i we decide whether or not to commence a basic experiment; this decision is made independently with some fixed probability p over all slots. We indicate this series of decisions through random variables $\{x_i\}$ that takes the value 1 (if a basic experiment is started in slot i) and 0 otherwise.

If $x_i = 1$, we dispatch *two* probes to measure congestion in slots i and $i+1$. The random variable y_i records the reports obtained from the probes as a 2-digit binary number, *i.e.*, $y_i = 00$ means "both probes did not observe congestion", while $y_i = 10$ means "the first probe observed congestion while the second one did not", and so on. Our methodology is based on the following fundamental assumptions, which, in view of the probe and its reporting design (as described in § 6) are very likely to be valid ones. *These assumptions are required in both algorithmic versions.* The basic algorithm requires a stronger version of these assumptions, as we detail later.

5.2.1 Assumptions

We do **not** assume that the probes accurately report congestion: we allow that congestion present in a given time slot may not be observed by any of the probe packets in that slot. However, we do assume a specific structure of the inaccuracy, as follows.

Let Y_i be the *true congestion state* in slot i, *i.e.,* $Y_i = 01$ means that there is no congestion at $t = i$ and there is congestion at $t = i+1$. Here, true means the congestion that would be observed were we to have knowledge of router buffer occupancy, queueing delays and packet drops. Of course, the value of Y_i is not available to us. Our specific assumption is that y_i is correct, *i.e.,* equals Y_i, at probability p_k that is independent of i and depends only on the number k of 1-digits in Y_i. Moreover, if y_i is incorrect, it must take the value 00. Explicitly,

(1) If $Y_i = 00$ (= no congestion occurs) then $y_i = 00$, too (= no congestion reported), with probability 1.

(2) If $Y_i = 01$ (= congestion begins), or $Y_i = 10$ (= congestion ends), then $\mathsf{P}(y_i = Y_i | (Y_i = 01) \cup (Y_i = 10)) = p_1$, for some p_1 which is independent of i. If y_i fails to match Y_i, then necessarily, $y_i = 00$.

(3) If $Y_i = 11$ (= congestion is on-going), then $\mathsf{P}(y_i = Y_i | Y_i = 11) = p_2$, for some p_2 which is independent of i. If y_i fails to match Y_i, then necessarily, $y_i = 00$.

5.2.2 Estimation

The basic algorithm assumes that $p_1 = p_2$ for consistent duration estimation, and $p_1 = p_2 = 1$ for consistent and unbiased frequency estimation. The estimators are as follows:

Congestion Frequency Estimation is straightforward. Denote the true frequency of congested slots by F. We define a random variable z_i whose value is the first digit of y_i. Our estimate is then

$$\widehat{F} = \sum_i z_i / M,$$

with the index i running over all the basic experiments we conducted, and M is the total number of such experiments.

This estimator is unbiased, $\mathsf{E}[\widehat{F}] = F$, since the expected value of z_i is just the congestion frequency F. Under mild conditions, the estimator is also consistent. For example, if the durations of the congestion episodes and congestion-free episodes are independent with finite mean, then the proportion of congested slots during an experiment over N slots converges almost surely, as N grows, to the congestion frequency F, from which the stated property follows.

Congestion Duration Estimation is more sophisticated. Recall that a congestion episode is one consecutive occurrence of "k congestions" preceded and followed by "no congestion", *i.e.,* its binary representation is written as:

$$01\ldots10.$$

Suppose that an oracle provides us with the state of the router's buffer at all possible time slots in our discretization. We then count all congestion episodes and their durations and find out that for $k = 1, 2, \ldots$, there were exactly j_k congestion episodes of length k. Then, congestion occurred over a total of

$$A = \sum_k k j_k$$

slots, while the total number of congestion episodes is

$$B = \sum_k j_k.$$

The average duration D of a congestion episode is then defined as

$$D := A/B.$$

In order to estimate D, we observe that, with the above structure of congestion episodes in hand, there are exactly B time slots i for which $Y_i = 01$, and there are also B time slots i for which $Y_i = 10$. Also, there are exactly $A+B$ time slots i for which $Y_i \neq 00$. We therefore define

$$R := \#\{i : y_i \in \{01, 10, 11\}\},$$

and

$$S := \#\{i : y_i \in \{01, 10\}\}.$$

Now, let N be the total number of time slots. Then $\mathsf{P}(Y_i \in \{01,10\}) = 2B/N$, hence $\mathsf{P}(y_i \in \{01,10\}) = 2p_1B/N$.
Similarly, $\mathsf{P}(Y_i \in \{01,10,11\}) = (A+B)/N$, and $\mathsf{P}(y_i \in \{01,10,11\}) = (p_2(A-B)+2p_1B)/N$. Thus,

$$\mathsf{E}(R)/\mathsf{E}(S) = \frac{p_2(A-B)+2p_1B}{2p_1B}.$$

Denoting $r := p_2/p_1$, we get then

$$\mathsf{E}(R)/\mathsf{E}(S) = \frac{r(A-B)+2B}{2B} = \frac{rA}{2B} - r/2 + 1.$$

Thus,

$$D = \frac{2}{r} \times \left(\frac{\mathsf{E}(R)}{\mathsf{E}(S)} - 1\right) + 1.$$

In the basic algorithm we assume $r = 1$, the estimator $\widehat{D}$ of D is then obtained by substituting the measured values of S and R for their means:

$$\widehat{D} := 2 \times \frac{R}{S} - 1.$$

Note that this estimator is *not* unbiased for finite N, due to the appearance of R in the quotient. However, it is consistent under the same conditions as those stated above for $\widehat{F}$, namely that congestion is described by an alternating renewal process with finite mean lifetimes D and D' for the congested and uncongested periods, respectively. In this case (with $r = 1$) R/N converges almost surely, as N grows, to $p(D+1)/(D+D')$ while S/N converges to $2p/(D+D')$, and hence $\widehat{D}$ converges almost surely to D.

5.3 Improved Algorithm

The improved algorithm is based on weaker assumptions than the basic algorithm: we no longer assume that $p_1 = p_2$. In view of the details provided so far, we will need, for the estimation of duration, to know the ratio $r := p_1/p_2$. For that, we modify our basic experiments as follows.

As before, we decide independently at each time slot whether to conduct an experiment. With probability 1/2, this is a basic experiment as before; otherwise we conduct an *extended experiment* comprising *three* probes, dispatched in slots $i, i+1, i+2$, and redefine y_i to be the corresponding 3-digit number returned by the probes, *e.g.,* $y_i = 001$ means "congestion was observed only at $t = i+2$", etc. As before Y_i records the true states that our ith experiment attempts to identify. We now make the following additional assumptions.

5.3.1 Additional Assumptions

We assume that the probability that y_i misses the true state Y_i (and hence records a string of 0's), does not depend on the length of Y_i but only on the number of 1's in the string. Thus, $\mathsf{P}(y_i = Y_i) = p_1$ whenever Y_i is any of $\{01, 10, 001, 100\}$, while $\mathsf{P}(y_i = Y_i) = p_2$ whenever Y_i is any of $\{11, 011, 110\}$ (note that we ignore the states 010 and 101, but address them below).

We claim that these additional assumptions are realistic, but defer the discussion until after we describe the reporting mechanism for congestion.

With these additional assumptions in hand, we denote

$$U := \#\{i : y_i \in \{011, 110\}\},$$

and

$$V := \#\{i : y_i \in \{001, 100\}\}.$$

The combined number of states 011, 110 in the full time series is still $2B$, while the combined number of states of the form 001, 100 is also $2B$. Thus, we have

$$\frac{\mathsf{E}(U)}{\mathsf{E}(V)} = r,$$

hence, with U/V estimating r, we employ (5.2.2) to obtain

$$\widehat{D} := \frac{2V}{U} \times \left(\frac{R}{S} - 1\right) + 1.$$

5.4 Validation

When running an experiment, our assumptions require that several quantities have the same mean. We can validate the assumptions by checking those means.

In the basic algorithm, the probability of $y_i = 01$ is assumed to be the same as that of $y_i = 10$. Thus, we can design a stopping criterion for on-going experiments based on the ratio between the number of 01 measurements and the number of 10 measurements. A large discrepancy between these numbers (that is not bridged by increasing M) is an indication that our assumptions are invalid. Note that this validation does not check whether $r = 1$ or whether $p_1 = 1$, which are two important assumptions in the basic design.

In the improved design, we expect to get similar occurrence rate for each of $y_i = 01, 10, 001, 100$. We also expect to get similar occurrence rate for $y_i = 011, 110$. We can check those rates, stop whenever they are close, and invalidate the experiment whenever the mean of the various events do not coincide eventually. Also, each occurrence of $y_i = 010$ or $y_i = 101$ is considered a violation of our assumptions. A large number of such events is another reason to reject the resulted estimations. Experimental investigation of stopping criteria is future work.

5.5 Modifications

There are various straightforward modifications to the above design that we do not address in detail at this time. For example, in the improved algorithm, we have used the triple-probe experiments only for the estimation of the parameter r. We could obviously include them also in the actual estimation of duration, thereby decreasing the total number of probes that are required in order to achieve the same level of confidence.

Another obvious modification is to use unequal weighing between basic and extended experiments. In view of the expression we obtain to $\widehat{D}$ there is no clear motivation for doing that: a miss in estimating V/U is as bad as a corresponding miss in R/S (unless the average duration is very small). Basic experiments incur less cost in terms of network probing load. On the other hand, if we use the reports from triple probes for estimating $\mathsf{E}(S)/\mathsf{E}(R)$ then we may wish to increase their proportion. Note that in our formulation, we cannot use the reported events $y_i = 111$ for estimating anything, since the failure rate of the reporting on the state $Y_i = 111$ is assumed to be unknown. A topic for further research is to quantify the trade-offs between probe load and estimation accuracy involved in using extended experiments of 3 or more probes.

6. PROBE TOOL IMPLEMENTATION AND EVALUATION

To evaluate the capabilities of our loss probe measurement process, we built a tool called BADABING[1] that implements the basic

[1] Named in the spirit of past tools used to measure loss including PING, ZING, and STING. This tool is approximately 800 lines of C++ and is available to the community for testing and evaluation.

algorithm from § 5. We then conducted a series of experiments with BADABING in the same laboratory environment and with the same test traffic scenarios described in § 4.

The objective of our lab-based experiments was to validate our modeling method and to evaluate the capability of BADABING over a range of loss conditions. We report results of experiments focused in three areas. While our probe process does not assume that we always receive true indications of loss from our probes, the accuracy of reported measurements will improve if probes more reliably indicate loss. With this in mind, the first set of experiments was designed to understand the ability of an individual probe (consisting of 1 to N tightly-spaced packets) to accurately report an encounter with a loss episode. The second is to examine accuracy of BADABING in reporting loss episode frequency and duration for a range of probe rates and traffic scenarios. In our final set of experiments, we compare the capabilities of BADABING with simple Poisson-modulated probing.

6.1 Accurate Reporting of Loss Episodes by Probes

An important component of our probe process is dealing with instances where individual probes (where a probe consists of a series of N packets) do not report loss accurately. In other words, ideally, a given probe P_i should report the following:

$$P_i = \begin{cases} 0: & \text{if a loss episode is not encountered} \\ 1: & \text{if a loss episode is encountered} \end{cases}$$

It should be noted that this requirement is only for a probe, not necessarily the individual packets within a probe. Satisfying this requirement is problematic because, as noted in § 3, many packets are successfully transmitted during loss episodes. Thus, we hypothesized that we might be able to increase the probability of probes correctly reporting a loss episode by increasing the number of packets in an individual probe. We also hypothesized that, assuming FIFO queueing, using one-way delay information could further improve the accuracy of individual probe measurements.

We investigated the first hypothesis in a series of experiments using the infinite TCP source background traffic and constant-bit rate traffic described in § 4. For the infinite TCP traffic, loss event duration were approximately 150 milliseconds. For the constant-bit rate traffic, loss episodes were approximately 68 milliseconds in duration. We used a modified version of BADABING to generate probes at fixed intervals of 10 milliseconds so that some number of probes would encounter all loss episodes. We experimented with probes consisting of between 1 and 10 packets. Packets in an individual probe were sent back to back per the capabilities of the measurement hosts (*i.e.,* with approximately 30 microseconds between packets). Probe packet sizes were set at 600 bytes[2].

Figure 7 shows the results of these tests. We see that for the constant-bit rate traffic, longer probes have a clear impact on the ability to detect loss. While about half of single-packet probes do not experience loss during a loss episode, probes with just a couple more packets are much more reliable indicators of loss. For the infinite TCP traffic, there is also an improvement as the probes get longer, but the improvement is relatively small. Examination of the details of the queue behavior during these tests demonstrates why the 10 packet probes do not greatly improve loss reporting ability for the infinite source traffic. As shown in Figure 8, longer probes begin to have a serious impact on the queuing dynamics during loss episodes.

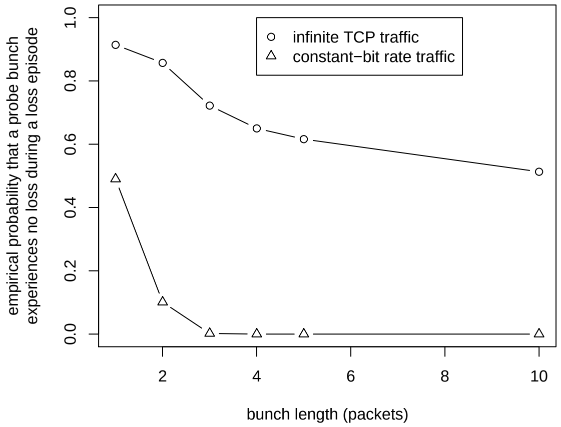


Figure 7: Results from tests of ability of probes consisting of N packets to report loss when an episode is encountered.

This observation, along with our hypothesis regarding one-way packet delays, led to our development of an alternative approach for identifying loss events. Our new method considers both individual packet loss with probes *and* the one-way packet delay as follows. For probes in which any packet is lost, we consider the one-way delay of the most recent successfully transmitted packet as an estimate of the maximum queue depth (OWD_{max}). We then consider a loss episode (or, more generally, a congestion episode) to be delimited by probes within τ seconds of an indication of a lost packet (*i.e.*, a missing probe sequence number) and having a one-way delay greater than $(1-\alpha) \times OWD_{max}$. Using the parameters τ and α, we mark probes as 0 or 1 according to § 5 and form estimates of loss episode frequency and duration.

This formulation of probe-measured congestion assumes that queuing at intermediate routers is FIFO. Also, we can keep a number of estimates of OWD_{max}, taking the mean when determining whether a probe is above the $(1-\alpha) \times OWD$ threshold or not. Doing so effectively filters loss at end host operating system buffers or in network interface card buffers, since such losses are unlikely to be correlated with end-to-end network congestion and delays.

We conducted a series of experiments with constant-bit rate traffic to assess the sensitivity of the loss threshold parameters. Using a range of values for probe send probability (p), we explored a cross product of values for α and τ. For α, we selected 0.025, 0.05, 0.10, and 0.20, effectively setting a high-water level of the queue of 2.5, 5, 10, and 20 milliseconds. For τ, we selected values of 5, 10, 20, 40, and 80 milliseconds. Figure 9(a) shows results for loss frequency for a range of p, with τ fixed at 80 milliseconds, and α varying between 0.05, 0.10, and 0.20 (equivalent to 5, 10, and 20 milliseconds). Figure 9(b) fixes α at 0.10 (10 milliseconds) while letting τ vary over 20, 40, and 80 milliseconds. We see, as expected, that with larger values of either threshold, estimated frequency increases. There are similar trends for loss duration (not shown). We also see that there is a trade-off between selecting a higher probe rate and more "permissive" thresholds. It appears that the break-even point for τ comes around the expected time between probes plus one or two standard deviations. The best α appears to depend both on the probe rate and on the traffic process and level of mul-

[2] This packet size was chosen to exploit an architectural feature of the Cisco GSR so that probe packets had as much impact on internal buffer occupancy as maximum-sized frames. Investigating the impact of packet size on estimation accuracy is a subject for future work.

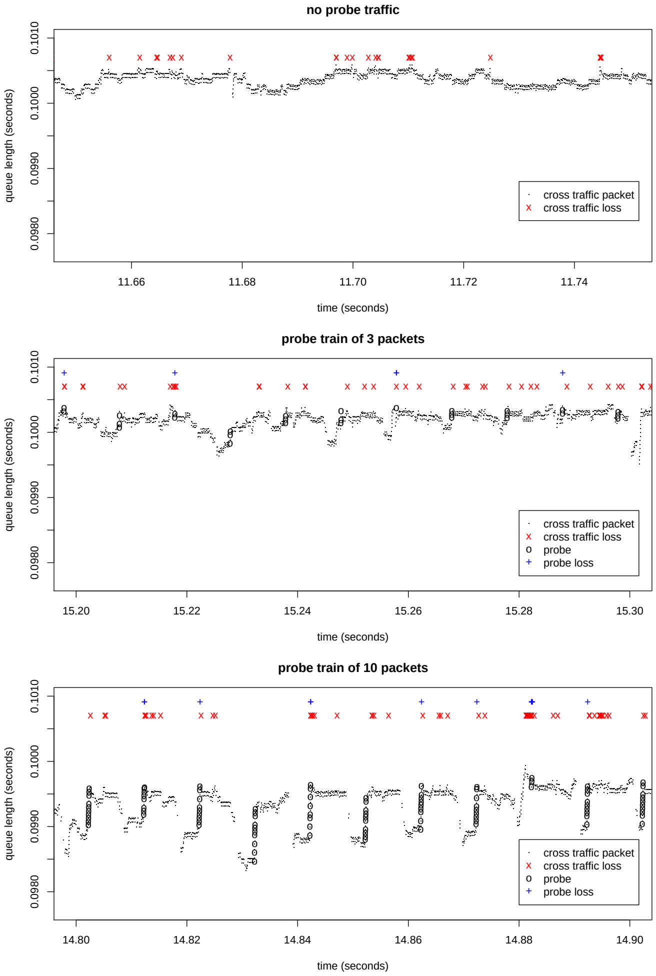


Figure 8: Queue length during a portion of a loss episode for different size loss probes. The top plot shows infinite source TCP traffic with no loss probes. The middle plot shows infinite source TCP traffic with loss probes of three packets, and the bottom plots shows loss probes of 10 packets. Each plot is annotated with TCP packet loss events and probe packet loss events.

tiplexing, which determines how quickly a queue can fill or drain. Considering such issues, we discuss parameterizing BADABING in general Internet settings in § 7.

6.2 Measuring Frequency and Duration

The formulation of our new loss probe process in Section 5 calls for the user to specify two parameters, N and p, where p is the probability of sending a probe at a given interval. In the next set of experiments, we explore the effectiveness of BADABING to report loss episode frequency and duration for a fixed N, and p using values of 0.1, 0.3, 0.5, 0.7, and 0.9 (implying that probe traffic consumed between 0.2% and 1.7% of the bottleneck link). With the time discretization set at 5 milliseconds, we fixed N for these experiments at 180,000, yielding an experiment duration of 900 seconds. We also examine the loss frequency and duration estimates for a fixed p of 0.1 and N of 720,000 from an hour-long experiment.

In these experiments, we used three different background traffic scenarios. In the first scenario, we used Iperf to generate random loss episodes at constant duration as described in Section 4. For the second, we modified Iperf to create loss episodes of three different durations (50, 100, and 150 milliseconds), with an average of 10 seconds between loss episodes. In the final traffic scenario, we used Harpoon to generate web-like workloads as described in § 4. For all traffic scenarios, BADABING was configured with probe sizes of 3 packets and with packet sizes fixed at 600 bytes. The three packets of each probe were sent back-to-back, according to the capabilities of our end hosts (approximately 30 microseconds between packets). For each probe rate, we set τ to the expected time between probes plus one standard deviation. For α, we used 0.2 for a probe rate of 0.1, 0.1 for probe rates of 0.3 and 0.5, and 0.5 for probe rates of 0.7 and 0.9.

For loss episode duration, results from our experiments described below confirm the validity of the assumption made in § 5.4 that the probability $y_i = 01$ is very close to the probability $y_i = 10$. That is, we appear to be equally likely to measure in practice the beginning of a loss episode as we are to measure the end. We therefore use the mean of the estimates derived from these two values of y_i.

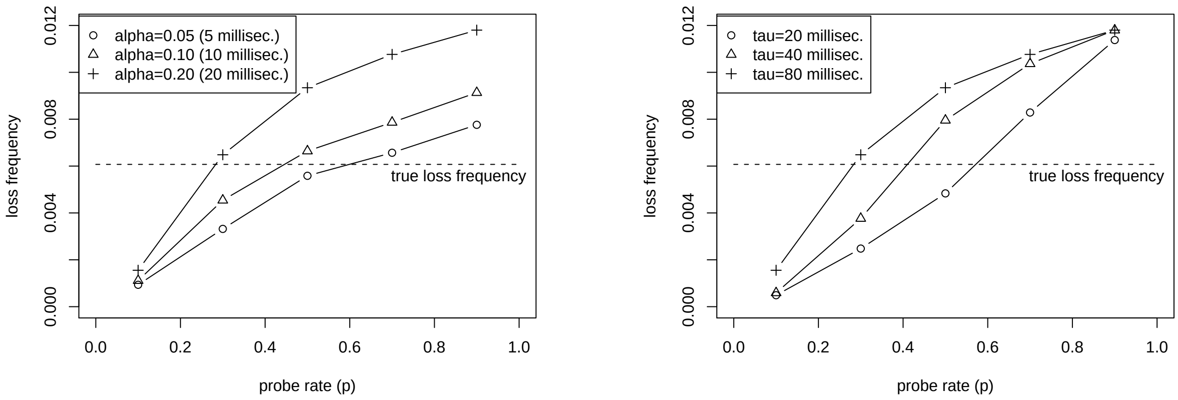
Table 4 shows results for the constant bit rate traffic with loss episodes of uniform duration. For values of p other than 0.1, the loss frequency estimates are close to the true value. For all values of p, the estimated loss episode duration was within 25% of the actual value.

Table 5 shows results for the constant bit rate traffic with loss episodes randomly chosen between 50, 100, and 150 milliseconds. The overall result is very similar to the constant bit rate setup with loss episodes of uniform duration. Again, for values of p other than 0.1 (*i.e.,* very low-impact probing), the loss frequency estimates are close to the true values, and all estimated loss episode durations were within 25% of the true value.

Table 6 displays results for the setup using Harpoon web-like traffic to create loss episodes. Since Harpoon is designed to generate average traffic volumes over relatively long time scales [31], the actual loss episode characteristics over these experiments vary. For loss frequency, just as with the constant bit rate traffic scenarios, the estimates are quite close except for the case of $p = 0.1$. For loss episode durations, all estimates except for $p = 0.3$ fall within a range of 25% of the actual value. The estimate for $p = 0.3$ falls just outside this range.

In Tables 4 and 5 we see, over the range of p values, an increasing trend in loss frequency estimated by BADABING. This effect arises primarily from the problem of selecting appropriate parameters α and τ, and is similar in nature to the trends seen in Figures 9(a) and 9(b). It is also important to note that these trends are peculiar to the well-behaved CBR traffic sources: such an increasing trend in loss frequency estimation does not exist for the significantly more bursty Harpoon web-like traffic, as seen in Table 6. We also note that no such trend exists for loss episode duration estimates. Empirically, there are somewhat complex relationships among the choice of p, the selection of α and τ, and estimation accuracy. While we have considered a range of traffic conditions in a limited, but realistic setting, we have yet to explore these relationships in more complex multi-hop scenarios, and over a wider range of cross traffic conditions. We intend to establish more rigorous criteria for BADABING parameter selection in our ongoing work.

Finally, Table 7 shows results from an experiment designed to understand the trade-off between an increased value of p, and an increased value of N. We chose $p = 0.1$, and show results using two different values of τ, 40 and 80 milliseconds. The background traffic used in these experiments was the simple constant bit rate traffic with uniform loss episode durations. We see that there is only a slight improvement in both frequency and duration estimates, with



(a) Estimated loss frequency over a range of values for α while holding τ fixed at 80 milliseconds.

(b) Estimated loss frequency over a range of values for τ while holding α fixed at 0.1 (equivalent to 10 milliseconds).

Figure 9: Comparison of the sensitivity of loss frequency estimation to a range of values of α and τ.

most improvement coming from a larger value of τ. Empirically understanding the convergence of estimates of loss characteristics as N grows larger is a subject for future experiments.

Table 4: BADABING loss estimates for constant bit rate traffic with loss episodes of uniform duration.

p	loss frequency		loss duration (seconds)	
	true	BADABING	true	BADABING
0.1	0.0069	0.0016	0.068	0.054
0.3	0.0069	0.0065	0.068	0.073
0.5	0.0069	0.0060	0.068	0.051
0.7	0.0069	0.0070	0.068	0.051
0.9	0.0069	0.0078	0.068	0.053

Table 5: BADABING loss estimates for constant bit rate traffic with loss episodes of 50, 100, or 150 milliseconds.

p	loss frequency		loss duration (seconds)	
	true	BADABING	true	BADABING
0.1	0.0083	0.0023	0.097	0.034
0.3	0.0083	0.0076	0.097	0.076
0.5	0.0083	0.0098	0.097	0.090
0.7	0.0083	0.0102	0.097	0.074
0.9	0.0083	0.0105	0.097	0.059

Table 6: BADABING loss estimates for Harpoon web-like traffic (Harpoon configured as described in § 4).

p	loss frequency		loss duration (seconds)	
	true	BADABING	true	BADABING
0.1	0.0044	0.0017	0.060	0.071
0.3	0.0011	0.0011	0.113	0.143
0.5	0.0114	0.0117	0.079	0.074
0.7	0.0043	0.0039	0.071	0.076
0.9	0.0031	0.0038	0.073	0.062

Table 7: Comparison of loss estimates for $p = 0.1$ and two different values of N and two different values for the τ threshold parameter.

N	τ	loss frequency		loss duration (seconds)	
		true	BADABING	true	BADABING
180,000	40	0.0059	0.0006	0.068	0.021
180,000	80	0.0059	0.0015	0.068	0.053
720,000	40	0.0059	0.0009	0.068	0.020
720,000	80	0.0059	0.0018	0.068	0.041

6.3 Comparing Loss Measurement Tools

Our final set of experiments compares BADABING with ZING using the constant-bit rate and Harpoon web-like traffic scenarios. We set the probe rate of ZING to match the link utilization of BADABING when $p = 0.3$ and the packet size is 600 bytes, which is about 876 kb/s, or about 0.5% of the capacity of the OC3 bottleneck. Each experiment was run for 15 minutes. Table 8 summarizes results of these experiments, which are similar to the results of § 4. (Included in this table are BADABING results from row 2 of Tables 4 and 6.) For the CBR traffic, the loss frequency measured by ZING is somewhat close to the true value, but loss episode durations are not. For the web-like traffic, neither the loss frequency nor the loss episode durations measured by ZING are good matches to the true values. Comparing the ZING results with BADABING, we see that for the same traffic conditions and probe rate, BADABING reports loss frequency and duration estimates that are significantly closer to the true values.

7. USING BADABING IN PRACTICE

There are a number of important practical issues which must be considered when using BADABING in the wide area:

- The tool requires the user to select values for p and N. Let us assume for the sake of the current discussion that the number of loss events is stationary over time. (Note that we allow the duration of the loss events to vary in an almost arbitrary way, and to change over time. One should keep in mind that in our current formulation we estimate the *average* duration and not the distribution of the durations.) Let

Table 8: Comparison of results for Badabing and zing with constant-bit rate (CBR) and Harpoon web-like traffic. Probe rates matched to $p = 0.3$ for badabing (876 kb/s) with probe packet sizes of 600 bytes. (Badabing results copied from row 2 of Tables 4 and 6.)

traffic	tool	loss frequency		loss duration	
scenario		true	measured	true (sec)	measured (sec)
CBR	Badabing	0.0069	0.0065	0.068	0.073
	Zing	0.0069	0.0041	0.068	0.010
Harpoon web-like	Badabing	0.0011	0.0011	0.113	0.143
	Zing	0.0159	0.0019	0.119	0.007

L be the mean number of loss events that occur over a unit period of time. For example, if an average of 12 loss events occur every minute, and our discretization unit is 5 milliseconds, then $L = 12/(60 \times 200) = .001$ (this is, of course, an estimate of the true the value of L). With the stationarity assumption on L, we expect the accuracy of our estimators to depend on the product pNL, but not on the individual values of p, N or L[3]. Specifically, a reliable approximation of the standard deviation in our estimation of duration is given by:

$$StdDev(duration) \approx \frac{1}{\sqrt{pNL}}$$

Thus, the individual choice of p and N allow a trade off between timeliness of results and impact that the user is willing to have on the link. Prior empirical studies can provide initial estimates of L. An alternate design is to take measurements continuously, and to report an when our validation techniques confirm that the estimation is robust. This can be particularly useful in situations where p is set at low level. In this case, while the measurement stream can be expected to have little impact on other traffic, it may have to run for some time until a reliable estimate is obtained.

- Our estimation of duration is critically based on correct estimation of the ratio B/M (*cf.* § 5). We estimate this ratio by counting the occurrence rate of $y_i = 01$, as well as the occurrence rate of $y_i = 10$. The number B/M can be estimated as the average of these two rates. The *validation* is done by measuring the *difference* between these two rates. This difference is directly proportional to the expected standard deviation of the above estimation. Similar remarks apply to other validation tests we mention in both estimation algorithms.

- The recent study on packet loss via passive measurement reported in [25] indicates that loss episodes in backbone links can be very short-lived (*e.g.,* on the order of several microseconds). The only condition for our tool to successfully detect and estimate such short durations is for our discretization of time to be finer, even in a slight way, than the order of duration we attempt to estimate. Such a requirement may imply that commodity workstations cannot be used for accurate active measurement of end-to-end loss characteristics in some circumstances. A corollary to this is that active measurements for loss in high bandwidth networks may require high-performance, specialized systems that support small time discretizations.

- Our classification of whether a probe traversed a congested path concerns not only whether the probe was lost, but how long it was delayed. While an appropriate τ parameter appears to be dictated primarily by the value of p, it is not yet clear how best to set α for an arbitrary path, when characteristics such as the level of statistical multiplexing or the physical path configuration are unknown. Examination of the sensitivity of τ and α in more complex environments is a subject for future work.

- To accurately calculate end-to-end delay for inferring congestion requires time synchronization of end hosts. While we can trivially eliminate offset, clock skew is still a concern. New on-line synchronization techniques such as reported in [26] or even off line methods such as [38] could be used effectively to address this issue.

8. SUMMARY, CONCLUSIONS AND FUTURE WORK

The purpose of our study was to understand how to measure end-to-end packet loss characteristics accurately with probes and in a way that enables us to specify the impact on the bottleneck queue. We began by evaluating the capabilities of simple Poisson-modulated probing in a controlled laboratory environment consisting of commodity end hosts and IP routers. We consider this testbed ideal for loss measurement tool evaluation since it enables repeatability, establishment of ground truth, and a range of traffic conditions under which to subject the tool. Our initial tests indicate that simple Poisson probing is relatively ineffective at measuring loss episode frequency or measuring loss episode duration, especially when subjected to TCP (reactive) cross traffic.

These experimental results led to our development of a probe process that provides more accurate estimation of loss characteristics than simple Poisson probing. The experimental design is constructed in such a way that the performance of the accompanying estimators relies on the total number of probes that are sent, but not on their sending rate. Moreover, simple techniques that allow users to validate the measurement output are introduced. We implemented this method in a new tool, Badabing, which we tested in our laboratory. Our tests demonstrate that Badabing, in most cases, accurately estimates loss frequencies and durations over a range of cross traffic conditions. For the same overall packet rate, our results show that Badabing is significantly more accurate than Poisson probing for measuring loss episode characteristics.

While Badabing enables superior accuracy and a better understanding of link impact versus timeliness of measurement, there is still room for improvement. For example, we have considered adding adaptivity to our probe process model in a limited sense. We are also considering alternative, parametric methods for inferring loss characteristics from our probe process. Another task is to estimate the variability of the estimates of congestion frequency and duration themselves directly from the measured data, under a minimal set of statistical assumptions on the congestion process.

Acknowledgments

We thank David Donoho for valuable discussions and the anonymous reviewers for their helpful comments. This work is supported in part by NSF grant numbers CNS-0347252, ANI-0335234, and CCR-0325653 and by Cisco Systems. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF or of Cisco Systems.

[3] Note that estimators that average individual estimations of the duration of each loss episode are not likely to perform that well at low values of p.

9. REFERENCES

[1] The Wisconsin Advanced Internet Laboratory. http://wail.cs.wisc.edu, 2005.

[2] A. Adams, J. Mahdavi, M. Mathis, and V. Paxson. Creating a scalable architecture for Internet measurement. *IEEE Network*, 1998.

[3] G. Almes, S. Kalidindi, and M. Zekauskas. A one way packet loss metric for IPPM. IETF RFC 2680, September 1999.

[4] S. Alouf, P. Nain, and D. Towsley. Inferring network characteristics via moment-based estimators. In *Proceedings of IEEE INFOCOM '00*, Tel Aviv, Israel, April 2000.

[5] G. Appenzeller, I. Keslassy, and N. McKeown. Sizing router buffers. In *Proceedings of ACM SIGCOMM '04*, Portland, OR, 2004.

[6] P. Barford and J. Sommers. Comparing probe- and router-based packet loss measurements. *IEEE Internet Computing*, September/October 2004.

[7] P. Benko and A. Veres. A passive method for estimating end-to-end TCP packet loss. In *Proceedings of IEEE Globecom '02*, Taipei, Taiwan, November 2002.

[8] J. Bolot. End-to-end packet delay and loss behavior in the Internet. In *Proceedings of ACM SIGCOMM '93*, San Francisco, September 1993.

[9] S. Brumelle. On the relationship between customer and time averages in queues. *Journal of Applied Probability*, 8, 1971.

[10] N. Cardwell, S. Savage, and T. Anderson. Modeling TCP latency. In *Proceedings of IEEE INFOCOM '00*, Tel-Aviv, Israel, March 2000.

[11] M. Coates and R. Nowak. Network loss inference using unicast end-to-end measurement. In *Proceedings of ITC Conference on IP Traffic, Measurement and Modeling*, September 2000.

[12] N. Duffield, F. Lo Presti, V. Paxson, and D. Towsley. Inferring link loss using striped unicast probes. In *Proceedings of IEEE INFOCOM '01*, Anchorage, Alaska, April 2001.

[13] S. Floyd and V. Paxson. Difficulties in simulating the Internet. *IEEE/ACM Transactions on Networking*, 9(4), 2001.

[14] C. Fraleigh, C. Diot, B. Lyles, S. Moon, P. Owezarski, D. Papagiannaki, and F. Tobagi. Design and deployment of a passive monitoring infrastructure. In *Proceedings of Passive and Active Measurement Workshop*, Amsterdam, Holland, April 2001.

[15] J. Hoe. Improving the start-up behavior of a congestion control scheme for TCP. In *Proceedings of ACM SIGCOMM '96*, Palo Alto, CA, August 1996.

[16] Merit Internet Performance Measurement and Analysis Project. http://nic.merit.edu/ipma/, 1998.

[17] Internet Protocol Performance Metrics. http://www.advanced.org/ippm/index.html, 1998.

[18] L. Le, J. Aikat, K. Jeffay, and F. Smith. The effects of active queue management on web performance. In *Proceedings of ACM SIGCOMM*, Karlsruhe, Germany, August 2003.

[19] W. Leland, M. Taqqu, W. Willinger, and D. Wilson. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, pages 2:1–15, 1994.

[20] J. Mahdavi, V. Paxson, A. Adams, and M. Mathis. Creating a scalable architecture for Internet measurement. In *Proceedings of INET '98*, Geneva, Switzerland, July 1998.

[21] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. TCP selective acknowledgement options. IETF RFC 2018, 1996.

[22] M. Mathis, J. Semke, J. Mahdavi, and T. Ott. The macroscopic behavior of the TCP congestion avoidance algorithm. *Computer Communications Review*, 27(3), July 1997.

[23] NLANR Passive Measurement and Analysis (PMA). http://pma.nlanr.net/, 2005.

[24] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP throughput: A simple model and its empirical validation. In *Proceedings of ACM SIGCOMM '98*, Vancouver, Canada, September 1998.

[25] D. Papagiannaki, R. Cruz, and C. Diot. Network performance monitoring at small time scales. In *Proceedings of ACM SIGCOMM Internet Measurement Conference '03*, Miami, FL, October 2003.

[26] A. Pasztor and D. Veitch. PC based Precision timing without GPS. In *Proceedings of ACM SIGMETRICS*, Marina Del Ray, CA, June 2002.

[27] V. Paxson. End-to-end Internet packet dynamics. In *Proceedings of ACM SIGCOMM '97*, Cannes, France, September 1997.

[28] V. Paxson. Strategies for sound Internet measurement. In *Proceedings of ACM SIGCOMM Internet Measurement Conference '04*, Taormina, Italy, November 2004.

[29] K. Salamatian, B. Baynat, and T. Bugnazet. Cross traffic estimation by loss process analysis. In *Proceedings of ITC Specialist Seminar on Internet Traffic Engineering and Traffic Management*, Wurzburg, Germany, July 2003.

[30] S. Savage. Sting: A tool for measuring one way packet loss. In *Proceedings of IEEE INFOCOM '00*, Tel Aviv, Israel, April 2000.

[31] J. Sommers and P. Barford. Self-configuring network traffic generation. In *Proceedings of ACM SIGCOMM Internet Measurement Conference '04*, 2004.

[32] The DETER Testbed. http://www.isi.edu/deter/, 2005.

[33] A. Tirumala, F. Qin, J. Dugan, J. Ferguson, and K. Gibbs. Iperf 1.7.0 – the TCP/UDP bandwidth measurement tool. http://dast.nlanr.net/Projects/Iperf. 2005.

[34] C. Villamizar and C. Song. High Performance TCP in ASNET. *Computer Communications Review*, 25(4), December 1994.

[35] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar. An integrated experimental environment for distributed systems and networks. In *Proceedings of 5th Symposium on Operating Systems Design and Implementation (OSDI)*, Boston, MA, December 2002.

[36] R. Wolff. Poisson arrivals see time averages. *Operations Research*, 30(2), March-April 1982.

[37] M. Yajnik, S. Moon, J. Kurose, and D. Towsley. Measurement and modeling of temporal dependence in packet loss. In *Proceedings of IEEE INFOCOM '99*, New York, NY, March 1999.

[38] L. Zhang, Z. Liu, and C. Xia. Clock Synchronization Algorithms for Network Measurements. In *Proceedings of IEEE Infocom*, New York, NY, June 2002.

[39] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker. On the constancy of Internet path properties. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop '01*, San Francisco, November 2001.