

# Rigorous Statistical Analysis of Internet Loss Measurements

Hung X. Nguyen, *Member, IEEE*, and Matthew Roughan, *Senior Member, IEEE*

**Abstract**—Loss measurements are widely used in today’s networks. There are existing standards and commercial products to perform these measurements. The missing element is a rigorous statistical methodology for their analysis. Indeed, most existing tools ignore the correlation between packet losses and severely underestimate the errors in the measured loss ratios. In this paper, we present a rigorous technique for analyzing performance measurements, in particular, for estimating confidence intervals of packet loss measurements. The task is challenging because Internet packet loss ratios are typically small and the packet loss process is bursty. Our approach, SAIL, is motivated by some simple observations about the mechanism of packet losses. Packet losses occur when the buffer in a switch or router fills, when there are major routing instabilities, or when the hosts are overloaded, and so we expect packet loss to proceed in episodes of loss, interspersed with periods of successful packet transmission. This can be modeled as a simple ON/OFF process, and in fact, empirical measurements suggest that an alternating renewal process is a reasonable approximation to the real underlying loss process. We use this structure to build a hidden semi-Markov model (HSMM) of the underlying loss process and, from this, to estimate both loss ratios and confidence intervals on these loss ratios. We use both simulations and a set of more than 18 000 hours of real Internet measurements (between dedicated measurement hosts, PlanetLab hosts, Web and DNS servers) to cross-validate our estimates and show that they are better than any current alternative.

**Index Terms**—Accuracy, confidence interval, loss measurement, hidden semi-Markov models (HSMMs).

## I. INTRODUCTION

**I** NTERNET performance is a topic of increasing concern. Online gaming, voice over IP, and other advanced Internet applications require better than best-effort quality. Packet loss, in particular, can result in serious degradation in a user’s quality of experience.

There are many ways of improving performance, but measurements are the key to ensuring ongoing quality. As in other areas of industry, we cannot fix problems if we do not know they exist. It is now common for network operators to perform ongoing performance measurements. The most common method to obtain these is to inject probe packets into the network to

sample the end-to-end performance, and there are now standards for such measurements [1], [9], [16] and commercial devices available to perform these tests, and measurements are even supported by some routers.

The missing element is a rigorous statistical methodology for the analysis of performance results. The analysis of packet loss may appear trivial. The loss ratio is just the number of dropped packets divided by the total number of probes. However, we show in this paper that calculating a loss ratio, by itself, is not enough. Isolated estimates can be misleading unless coupled with some measure of the accuracy of the estimate. However, naive confidence intervals calculated using the assumption that probe measurements are independent ignore the bursty structure of packet losses and misrepresent the accuracy of these estimates.

In this paper, we present a rigorous technique for analyzing performance measurements, in particular, for estimating confidence intervals for packet loss measurements. The task is challenging because Internet packet loss ratios are typically small (and estimating small probabilities is always challenging), and the packet loss process is bursty. It is common for a particular set of probes to miss some intervals where loss occurs, and we need to take this into account in estimating the accuracy of our measurements.

Our approach draws on the past experience of modeling packet loss. Various models for packet loss processes have been presented (for instances, see [2], [5], [8], [14], [19], [20], [26], and [29]), usually with the aim of using these models for simulation or for modeling application protocols. These studies have unanimously found correlations between packet losses, but have suggested different approaches for modeling these losses. Our approach is motivated by some simple observations about the mechanism of packet losses. Packet losses occur when the buffer in a switch, router, or an end-host fills, when there are routing instabilities, or when there are software/hardware failures, and so we expect packet loss to proceed in episodes of loss, interspersed with periods of successful packet transmission. This can be modeled as a simple ON/OFF process, and in fact, empirical measurements suggest that an alternating renewal process is a reasonable approximation to the real underlying loss process. Note that this model is more sophisticated than simple exponential models and can model non-Poissonian loss behaviors that occur due to route reconvergence, overloaded hosts, etc.

We can use this structure to build a hidden semi-Markov model (HSMM) of the underlying loss process and, from this, estimate both loss ratios and confidence intervals on these loss ratios. While our approach does use simplifying assumptions about the underlying losses, the proof is in the pudding. We use

Manuscript received November 30, 2011; revised June 19, 2012; accepted July 02, 2012; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor M. Allman. Date of publication July 17, 2012; date of current version June 12, 2013. This work was supported by the Australian Research Council under Grants DP0985063 and DP110103505.

The authors are with the School of Mathematical Sciences, The University of Adelaide, Adelaide, SA 5005, Australia (e-mail: hung.nguyen@adelaide.edu.au; matthew.roughan@adelaide.edu.au).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNET.2012.2207915

a set of real Internet performance measurements to cross-validate our estimates and show that they are more reasonable than any current alternative. The data and code are publicly available at <http://bandicoot.maths.adelaide.edu.au/SAIL/>.

Our work provides a rigorous statistical analysis of Internet packet loss measurements. Such an analysis is helpful to operators for ongoing network measurements, but is also critical in the specification of service level agreements (SLAs). Current SLAs sometimes specify required packet loss ratios and punitive measures should these be exceeded. However, what we have shown in this paper is that without some care, it is easy to write an SLA that could not be defended in court simply because practical measurements are not accurate enough to assess whether we are complying with the SLA. The specification of an SLA must be done in conjunction with a specification of the measurements that will be used for confirmation.

The work presented here has additional impact. A question that is often asked is ‘‘How many probes do I need?’’ This question is, formally, a statistical experimental design question. It cannot be correctly answered without an understanding of the correct approach for the analysis of the results, which we provide here.

## II. BACKGROUND AND RELATED WORK

### A. Active Probing

The typical approach to Internet performance measurement is to actively send *probe* packets across the network and to measure the performance of these probes. As a result of substantial work, tools have been developed for this type of probing [3], [4], [15], [16], [22]. The probes provide a sample of the network losses, and it is the accuracy of this sample that we wish to assess.

The temporal pattern of probes sent has received some interest. Throughout this work, we use Poisson probes, i.e., probes sent at the epochs of a Poisson process. While there are works suggesting that other probing strategies may be more efficient [3], [22], Poisson probes have a number of key advantages, for instance, the Poisson Arrivals See Time Averages (PASTA) result ensures that the sampled measurements will be unbiased (for more detail, see RFC 2330 [16]). In addition, subsamples of Poisson probes will also form a Poisson sequence. We use this fact to cross-validate our results. Note, however, that our HSMM does not assume the probes to be Poisson and works with all probe patterns.

We shall assume that the probes have zero size and do not perturb the loss process. This assumption is reasonable if care is taken to not overload the path by having small probes and by sending probes with low frequency.

### B. Existing Loss Models

Studies of Internet packet losses (for instances, see [2], [5], [8], [14], [19], [20], [26], and [29]) have unanimously agreed that packet loss measurements experience correlations. Several models have been proposed to explain these correlations, for instance, the discrete  $2^k$ -state Markov model [5], [26] where the state represents the loss/success of the previous  $k$  packets. Obviously, the potentially large state space of this model limits us

to smaller values of  $k$ , e.g., when  $k = 1$ , we get as a special case the popular Gilbert model [7]. A simple model that captures only the distribution of loss burst lengths is the  $k$ -state extended Gilbert model [20]. This model, however, does not provide information about the burstiness or clustering of the loss runs and therefore does not fully specify the loss process. A more fundamental problem for the analysis considered here is the fact that these are models for the *observed* loss process, which is discrete. For many applications, this is what is needed. For instance, when using a loss model to understand the impact of packet loss on an application, all that is needed is knowledge of which packets will be lost. Likewise, the extended Gilbert model has been widely used to study performance of forward error correction schemes [8]. However, in this paper, we are trying to assess the accuracy of a loss sample with respect to the real underlying loss process, which is continuous time. For this, we need a continuous-time model to describe the state of the underlying loss process. Moreover, we can use this model to study the performance of applications with different packet patterns than those observed.

Our approach draws from the Gilbert model, but extends it to the continuous-time domain. The main idea of the extended Gilbert model is to base the future loss probability on the previous  $k$  losses but not the previous no-loss data. Several authors [12], [26], [29] studied the properties the loss and success runs. They find that in most of their traces, the loss run lengths are short and uncorrelated. However, in many cases, the arrivals of the loss and no-loss runs cannot be modeled as Poisson. Given such observations, the obvious continuous-time model for our data is an ON/OFF alternating renewal model, which we will discuss in more detail.

## III. MEASUREMENT ACCURACY

### A. Estimating the Loss Ratio From the Probes

We model the loss process on an Internet path as a continuous-time binary stochastic process  $I(t)$ :  $I(t) = 1$  if at time  $t$  an arriving packet would be dropped, and  $I(t) = 0$  otherwise. For estimates to be valid, we must assume that this process is wide-sense stationary, which means that its mean, variance, and autocovariance are all constant with respect to  $t$  for all  $t \geq 0$ . The mean of  $I$  is just the loss ratio, which we denote as  $p$ , and can consequently be written as  $p = \mathbb{E}[I(t)]$ , and in addition, we use the notation  $\text{VAR}(I(t)) = \sigma^2$  and  $\text{Cov}(I(t), I(t+s)) = R(s)$ .

In practice,  $I(t)$  cannot be observed directly. We shall send  $N$  probes at times  $t_1, \dots, t_N$ , resulting in samples  $I_i = I(t_i)$  of the underlying process. Given  $N$  samples,  $I_1, \dots, I_N$ , the standard estimator for the loss ratio is

$$\hat{p} = \frac{1}{N} \sum_{i=1}^N I_i. \quad (1)$$

When the loss process is stationary and ergodic, and we use Poisson probes, the above estimator is unbiased and is guaranteed to converge to the true loss ratio  $p = \mathbb{E}[I(t)]$  (in fact, more general results [10] guarantee convergence in a much wider range of cases, but these are not needed here).

Unfortunately, the theory of Poisson probes says little about the rate of convergence or its variance for finite  $N$ . The naive estimate of the variance (which seems to be the basis for the few occasions where confidence intervals are calculated for loss measurements) can be derived from the assumption that the loss process is a Bernoulli process, i.e., packets losses are independent. In this case, the number of packet losses follows a Binomial distribution, and the variance of  $\hat{p}$  is

$$\text{VAR}(\hat{p}) = p(1 - p)/N \quad (2)$$

where this would be typically estimated using  $\hat{p}$  itself. However, we show in this paper that this variance estimator produces misleadingly small confidence intervals for loss estimates.

The correct variance of  $\hat{p}$  based on samples of  $I(t)$  is

$$\begin{aligned} \text{VAR}(\hat{p}) &= \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \mathbb{E}[I_i I_j] - p^2 \\ &= \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N R(\tau_{ij}) \end{aligned} \quad (3)$$

where  $R(\tau)$  is the autocovariance function of  $I(t)$ , and  $\tau_{ij}$  is the time between the  $i$ th and  $j$ th probes. The variance of  $\hat{p}$  is a function of both the correlations structure of the loss process  $I(t)$  and the structure of the sampling stream.

The formula (3) can be simplified (see [18]), which is useful for calculating the asymptotic variance as  $N \rightarrow \infty$ , but that is not needed here. We know the sample times  $t_i$ , and so if we knew the autocovariance function of  $I(t)$ , we could easily compute the above sum. One approach to estimating variance of  $\hat{p}$  is to empirically estimate  $R(\tau)$  based on our sampled measurements, and from this derive the variance estimator. This suffers from the same problems alluded to earlier for large Markov models of the loss process. The function  $R(\tau)$  represents a large number of parameters to estimate, and thus we are in danger of obtaining high-variance, overfitted estimates. We show later that this approach may even fail simple sanity tests, e.g., nonnegativity.

The standard statistical alternative to direct estimation is to build a model of the data, estimate the model parameters, and use this model to derive the autocovariance.

### B. Alternating ON/OFF Renewal Model

The loss process can be considered to consist of alternating periods of consecutive losses and periods without loss, i.e.,  $I(t) \equiv \{(A_i, B_i)\}_{i=1}^{\infty}$ , where  $A_i$  is the length of the  $i$ th loss period and  $B_i$  is the length of the  $i$ th no-loss period. This process is an alternating *renewal* process iff the  $A_i$  are independent and identically distributed (IID), the  $B_i$  are IID, and  $A_i$  and  $B_j$  are independent for all  $i$  and  $j$ . This assumption was studied in [12], and we validate it on our measurement data in Section VII-B.2 for both in-network and system induced losses. It turns out that, despite not being perfect, it is more reasonable than any (practical) Markov assumption as it allows for arbitrary correlations over time (alternating renewal processes may even display long-range dependence), but also leads to a practical model from the point of view of estimation.

The alternating renewal process is described by the density functions of  $A_i$  and  $B_i$ , which we denote by  $f_A(\cdot)$  and  $f_B(\cdot)$  respectively, and we denote their respective means by  $\mathbb{E}[A_i] = \mu_A$  and  $\mathbb{E}[B_i] = \mu_B$ .

In our approach, we need a parametric description of these distributions. Here, we use the Gamma distribution. Our initial experiments used the exponential distribution because it is the simplest reasonable distribution one can use for such processes. In that case, the resulting process is Markov, and so the resulting analysis is relatively easy. However, this exponential model did not produce satisfactory results. The Gamma distribution's density function is given by

$$\mathbb{P}(X \in [x, x + dx]) = x^{k-1} \frac{e^{-x/\theta}}{\Gamma(k)\theta^k} dx \quad (4)$$

where  $\Gamma(\cdot)$  is the gamma function,  $k$  is the shape parameter, and  $\theta$  is the scale parameter (the mean value of  $X$  is  $k\theta$ , and the variance is  $k\theta^2$ ). It is a natural generalization of the exponential distribution to a two-parameter family that still includes the exponential distribution as a special case (when  $k = 1$ ). As  $k$  increases, the Gamma distribution can approximate a Gaussian distribution, or even a deterministic variable plus jitter. When  $k < 1$ , we can obtain highly skewed distributions with relatively long tails.

Using Gamma distributions, the ON/OFF model has four parameters  $w = \{(k_0, \theta_0), (k_1, \theta_1)\}$ , where  $(k_0, \theta_0)$  are the parameters of the ON (lossy) periods and  $(k_1, \theta_1)$  are the parameters of the OFF (no loss) periods.

### C. Modeling the Sampling Process

The model presented above is continuous time, but our samples are discrete, so we need to construct the sample process from the continuous model as follows. Define the  $i$ th loss run as a sequence of consecutive lost probe packets that arrive to the path on the time interval  $A_i$ . Let  $Y_i$  be the length of the  $i$ th loss run. Note that the loss run can have zero length  $Y_i = 0$  if there are no probes on the time interval  $A_i$ . Similarly, a success run is the sequence of consecutive probe packets that arrive to the path on the time interval  $B_i$ . The length of the success run is denoted by  $Z_i$  and is measured by the number of consecutive successful packets. Also denote by  $L$  the number of pairs of loss and no-loss runs, and note the fact that the resulting process will be a discrete-time alternating renewal process.

Obviously, our observations of  $\{(Y_i, Z_i)\}_{i=1}^L$  will distort our view of  $\{(A_i, B_i)\}_{i=1}^L$  because we do not see the exact start and end of intervals. We could easily correct for this issue if not for the fact that we do not even observe all intervals. If we do not have a probe during some interval (i.e.,  $Y_i$  or  $Z_i = 0$ ), then we do not even know this interval exists. As a result, we will actually observe an alternating process  $I'(t)$  where some (short) periods of loss and no loss are omitted, and the intervals on either side are amalgamated.

Consider the simple example in Fig. 1 with 12 samples of the alternating loss process  $I(t)$  shown on the top line. The samples,  $\{1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0\}$  omit intervals  $A_1$  and  $B_3$ . We ignore periods at the start and end (where we cannot tell the true length of the sequence), so there are only two well-defined loss runs and success runs:  $(Y'_1, Z'_1) = (3, 3)$  and  $(Y'_2, Z'_2) =$

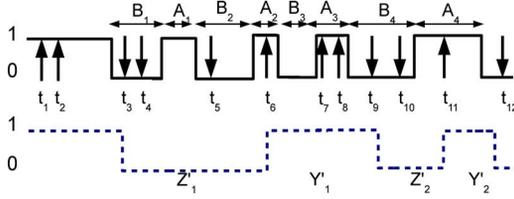


Fig. 1. Continuous time ON/OFF process (top) sampled by probes at times  $t_i$ . The inferred ON/OFF process (bottom) misses some transitions and distorts the size of others.

(1, 2) in the observed data. Note that in this example, even though there are four complete transitions in the continuous time process (i.e.,  $L = 4$ ), there are only two loss and no-loss runs in the measurements. The probes miss several state transitions and overestimate the lengths of the loss and no-loss runs as shown on the bottom line.

When probes are sent as Poisson process with rate  $\lambda$ , we can analytically compute the probability that the probes miss an ON/OFF period using the following lemma.

*Lemma 1:* For the alternating renewal process defined above

$$\begin{aligned}\mathbb{P}(Y_i = 0) &= 1/(\lambda \theta_0 + 1)^{k_0} \\ \mathbb{P}(Z_i = 0) &= 1/(\lambda \theta_1 + 1)^{k_1}.\end{aligned}$$

*Proof:* From (4)

$$\mathbb{P}(A_i \in [t, t + dt)) = t^{k_0-1} \frac{e^{-t/\theta_0}}{\Gamma(k_0)\theta_0^{k_0}} dt.$$

The number of probes  $Y_i$  in an interval of length  $A_i = t$  has distribution

$$\mathbb{P}(Y_i = n | A_i = t) = \frac{e^{-\lambda t} (t\lambda)^n}{n!}.$$

Thus

$$\begin{aligned}\mathbb{P}(Y_i = 0) &= \int_0^\infty \frac{e^{-\lambda t} (t\lambda)^0}{0!} t^{k_0-1} \frac{e^{-t/\theta_0}}{\Gamma(k_0)\theta_0^{k_0}} dt \\ &= \frac{1}{\Gamma(k_0)\theta_0^{k_0}} \int_0^\infty t^{k_0-1} e^{-t(\lambda+1/\theta_0)} dt \\ &= \frac{1}{(\lambda \theta_0 + 1)^{k_0}}\end{aligned}\quad (5)$$

where the integral follows from the definition of the Gamma function. Using the same approach, we can obtain a similar formula for  $\mathbb{P}(Z_i = 0)$ . ■

As we can see, many of the short loss or no-loss bursts will be missed by the probes, especially when  $\lambda \theta_i \ll 1$ , which is common in real measurements where both  $\lambda$  and  $\theta_i$  are small. The missing loss bursts create problems when estimating the length of intervals, and even how many intervals are present. These will impact the accuracy of the estimated loss ratios and their confidence intervals. To overcome this problem, we use a hidden semi-Markov model with missing observations to model the sample process. This model is described next.

#### IV. HIDDEN SEMI-MARKOV MODEL

One standard approach for estimation of an underlying process where we have incomplete samples of the process and its transitions is to use an HSMM [17].

A Markov chain is a sequence of random variables  $\mathcal{Q} = q_1 \dots q_T$  with the Markov property: Given the present state, the future and past states are independent. Consider a Markov chain with  $M$  possible states  $\mathcal{S} = \{s_1, \dots, s_M\}$ . The Markov property is formally defined as

$$\mathbb{P}(q_{t+1} = s_i | q_1, q_2, \dots, q_t) = \mathbb{P}(q_{t+1} = s_i | q_t = s_j).$$

If the states of the Markov process are not directly observed, but rather we see some output sequence that is probabilistically associated with the Markov chain, the process is referred to as a hidden Markov model (HMM) [17]. An HMM is formally defined by the quintuple:

- the set of  $M$  states  $\mathcal{S} = \{s_1, \dots, s_M\}$ ;
- the set of  $V$  observation symbols  $\mathcal{V} = \{v_1, \dots, v_H\}$ ;
- the initial probability

$$\boldsymbol{\pi} = (\pi_1, \dots, \pi_M), \quad \text{where } \pi_i = \mathbb{P}(q_1 = s_i);$$

- the time-independent state transition probability

$$\mathbf{A} = (a_{ij})_{M \times M}, \quad \text{where } a_{ij} = \mathbb{P}(q_{t+1} = s_j | q_t = s_i);$$

- the time-independent observation probability

$$\mathbf{B} = (b_{ik})_{M \times H}, \quad \text{where } b_{ik} = \mathbb{P}(O_t = v_k | q_t = s_i).$$

The HMM presented above is a discrete-time model. To represent the continuous-time loss process in a discrete HMM, we divide the measurement interval into  $T$  slots of length  $\Delta t$ . Ideally,  $\Delta t$  should be set to the length of the smallest loss period so that the probability that a state change occurs within every time-slot is negligible. However, this information is not available from measurements. The alternative is to make  $\Delta t$  sufficiently small, independent of the real loss process, so that it is smaller than the length of any realistic loss burst with high probability. This approach, however, increases the computation time significantly with little gain on the accuracy for two reasons.

First, making  $\Delta t$  small will only help the inference if we have enough measurements so that each discrete interval is measured at least once. Without the measurements to provide information about the loss process, an inference algorithm, at best, provides the most probable estimate among all the possible outcomes for the unobserved intervals. Increasing the number of unobserved intervals by having small  $\Delta t$  therefore will not improve the accuracy of the inference.

Second, for a given measurement interval, the number of discrete intervals  $T$  is inversely proportional to  $\Delta t$ . As HMM inference algorithms have quadratic computation time in  $T$ , making  $\Delta t$  too small will quickly make the HMM infeasible for real-time inference. Finding the optimal value for  $\Delta t$  is a nontrivial optimization problem given the underlying loss process (not available from active measurements) and the sampling rate  $\lambda$ . Our tool is designed to provide accurate estimates of the loss process from any set of measurements. We therefore adopt a pragmatic approach of setting  $\Delta t$  to half the average intersample

time, which provides a reasonable compromise between speed and accuracy. We show in this paper that this value of  $\Delta t$  works well in simulations and a wide range of measured traces. We shall leave the problem of finding the optimal parameter values for future work.

In an HMM, the state durations are treated as either fixed (one time unit) or geometrically distributed. In either case, the state sequence  $\mathcal{Q} = q_1 \dots q_T$  is a Markov process since the state at time  $t$ ,  $q_t$  depends on its past only through the most recent state  $q_{t-1}$ . This model has already been used in the context of packet loss modeling [19], but our model is not Markov. We found that to produce useful results, our model needs to allow *nonexponential* interval times, and so the process is not memoryless. Indeed, a parametric model with Gamma distributions  $p_m(d)$  for the state durations was found to be useful for modeling packet losses [12]. When the duration for state  $m$  is independent of all other durations of all other states, our process falls into the class of *semi-Markov models* [27], i.e., models whose state transitions are Markov, even if the times between transitions are not. A hidden semi-Markov model is therefore defined by the set of parameters  $\{\mathcal{S}, \mathcal{V}, \mathbf{A}, \mathbf{B}, \boldsymbol{\pi}, \{p_m(d)\}\}$ .

The HSMM for Internet losses has two states  $\mathcal{S} = \{1, 2\}$ , where state 1 means loss (ON) and state 2 means no-loss (OFF). As we have only two states, the state transition is straightforward. Once the process finishes state  $m$ , it will move to the other state  $n$  with probability 1, i.e.,

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

The observation set  $\mathcal{V}$  is more complex as in many time-slots there are no probes. In any given slot, we therefore have three possible observations: 1) a loss ( $O_t = 1$ ) if the loss process is ON ( $I(t) = 1$ ) and a probe is sent in that time-slot; 2) a success ( $O_t = 2$ ) if the loss process is OFF ( $I(t) = 0$ ) and a probe is sent in that slot; 3) a *null* ( $O_t = \emptyset$ ) observation if no probe is sent in that interval. The set of possible observations  $O_t$  for any time-slot  $t$  is  $\mathcal{V} = \{v_1 = 1, v_2 = 2, v_3 = \emptyset\}$ .

When the probes are Poisson with rate  $\lambda$ , the observations probabilities are

$$b_{mk} = \mathbb{P}(O_t = v_k | q_t = s_m) = \begin{cases} 1 - e^{-\lambda \Delta t}, & \text{if } v_k = m \\ e^{-\lambda \Delta t}, & \text{if } v_k = \emptyset \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

In the above formula,  $e^{-\lambda \Delta t}$  is the probability that there is no probe in the interval of length  $\Delta t$ , i.e.,  $e^{-\lambda \Delta t}$  is the probability that we observe a null event in time-slot  $t$ .

Our interest is to infer  $(k_m, \theta_m)$  for the Gamma distribution of the duration of state  $m$ , given the observations  $\mathcal{O} = \{O_1, \dots, O_T\}$ . We seek the maximum likelihood estimator of the model parameters, i.e., our goal is to find the parameters  $\omega = \{(k_0, \theta_0), (k_1, \theta_1)\}$  that maximize  $\mathbb{P}(\mathcal{O} | \omega)$ . There are standard algorithms to infer the parameters of the above HSMM. In this paper, we use the algorithms in [28] for this purpose. These are extensions of the well-known Baum–Welch algorithm for standard HMMs, though we have made a number of improvements to the speed of the algorithm in our implementation.

## V. SAIL: STATISTICALLY ACCURATE INTERNET LOSS MEASUREMENTS

### A. Variance and Confidence Intervals for Loss Ratio

Once the parameters  $\theta_0, k_0, \theta_1,$  and  $k_1$  of the continuous-time ON/OFF model are estimated using the HSMM algorithm, we can use the model to predict properties of any set of sample data (it does not have to be Poisson sampled), for instance, the mean, variance, and autocovariance.

The mean of the continuous-time alternating renewal process  $I(t) \equiv \{(A_i, B_i)\}_{i=1}^{\infty}$  with  $\mathbb{E}[A_i] = \mu_A = \theta_0 k_0$  and  $\mathbb{E}[B_i] = \mu_B = \theta_1 k_1$  is given by [6]

$$\mathbb{E}[I(t)] = \frac{\mu_A}{\mu_A + \mu_B} \quad (7)$$

and the unbiased nature of Poisson samples means that they will see this loss rate on average. We can derive the variance of  $\hat{p}$  using (3), but first we must calculate the autocovariance  $R(\tau)$ . We use the following results from [11]. Recall that  $f_A(t)$  and  $f_B(t)$  are the density of the ON and OFF period, respectively. Denote by  $f_A^*(s)$  and by  $f_B^*(s)$  the Laplace transforms of  $f_A(t)$  and  $f_B(t)$ , respectively, e.g.,

$$\text{Laplace}(f_A) = f_A^*(s) = \int_0^{\infty} e^{-st} f_A(t) dt. \quad (8)$$

Then, the Laplace transform  $R^*(s)$  of  $R(\tau)$  is given by [11, Theorem 2] to be

$$R^*(s) = \frac{\mathbb{E}[I(t)]}{s} - \frac{(1 - f_A^*(s))(1 - f_B^*(s))}{s^2 \mathbb{E}[C_i](1 - f_C^*(s))} \quad (9)$$

where  $C_i = A_i + B_i$  is the length of the joint ON/OFF intervals, which has density  $f_C(t)$ , with Laplace transform  $f_C^*(s)$ , and mean  $\mathbb{E}[C_i] = \mu_A + \mu_B$ .

When  $A$  and  $B$  have Gamma distributions, we can rewrite (9) as

$$R^*(s) = \frac{k_0 \theta_0}{s(k_0 \theta_0 + k_1 \theta_1)} - \frac{(1 - (\theta_0 s + 1)^{k_0})(1 - (\theta_1 s + 1)^{k_1})}{s^2 (k_0 \theta_0 + k_1 \theta_1) ((\theta_0 s + 1)^{k_0} (\theta_1 s + 1)^{k_1} - 1)}. \quad (10)$$

At this point, we have to use a numerical method to invert the Laplace transform to find  $R(\tau)$  from  $R^*(s)$ . We use two methods to invert the Laplace transform. We used the Week's method in [24], but we found that Week's algorithm is somewhat sensitive to the discretization sizes used in numerical integrals and does not always converge. In our implementation, when the Week's algorithm fails to converge, we use a slightly slower but more reliable method for computing autocovariances. Given the model parameters, it is trivial to simulate the ON/OFF process above for an arbitrary time period and directly measure the autocovariance of this process to arbitrary accuracy. When using this method, two parameters—the maximum lag and the number of samples—need to be chosen carefully. In our implementation, the maximum lag for estimating  $R(\cdot)$  is chosen to be the maximum lag given by the real samples, i.e., the largest interarrival time between the samples. The simulation duration needs

to be sufficiently large in order to compute the autocovariance function  $R(\cdot)$  accurately. We use a large default value of  $10^6$  samples in each simulation. This number is large enough for most of our measurement traces with  $10^4$  or less samples. Note that occasionally we may not get enough samples even in simulation to accurately estimate  $R(\tau_{ij})$  for large  $\tau_{ij}$ . In these cases, we eliminate any  $R(\tau_{ij})$  that has less than 30 samples in our computation of (3).

Once we know the autocovariance function  $R(\tau)$ , the calculation of the variance of  $\hat{p}$  is straightforward based on (3). We can then compute Gaussian confidence intervals for  $\hat{p}$  (we use 95th percentile intervals in our results), i.e.,

$$\hat{p} \pm z_{\beta/2} \sqrt{\text{VAR}(\hat{p})/N} \quad (11)$$

where  $z_{\beta/2}$  are such that  $\mathbb{P}(-z_{\beta/2} < N(0,1) < z_{\beta/2}) = 1 - \beta$ . For example, to compute the 95% confidence intervals,  $z_{2.5} = 1.96$  is used, and the confidence intervals would be  $\hat{p} \pm 1.96 \sqrt{\text{VAR}(\hat{p})/N}$ . In principle, we might model the distribution of measurements more accurately to refine these confidence intervals, but Gaussian intervals are easy to calculate.

### B. Analysis Algorithm

Our Statistically Accurate Internet Loss measurement (SAIL) algorithm consists of three main components: measurement, modeling, and inference. SAIL is fast and easy to use. It can be applied to any loss trace that has probe sending times and probe outcomes. The details of the algorithm are described as follows.

- 1) INPUT
  - a) The set  $\{t_1, \dots, t_N\}$  of probe sending times.
  - b) The set  $\{I_1, \dots, I_N\}$  of probe outcomes.
- 2) ALGORITHM
  - a) Apply the forward and backward algorithm to find  $\omega = \{(k_0, \theta_0), (k_1, \theta_1)\}$ .
  - b) Apply the inverse Laplace transform (or simulation) to find  $R(\tau)$ .
  - c) Compute the loss ratio and its confidence intervals.
- 3) OUTPUT
  - a) The loss ratio  $\hat{p}$  and its confidence interval.
  - b)  $\omega = \{(k_0, \theta_0), (k_1, \theta_1)\}$ .

## VI. SIMULATION RESULTS

Simulations offer the ground truth needed to verify our analysis techniques. Of course, in simulations we lose the “richness” of real networks, and thus the simulations in this section are only used to show that our inference algorithm works correctly and is essential in the computation of the errors of the estimated loss ratio when the model assumptions are correct. We validate our model and method in real network data in the next section.

### A. Setting

We wrote our simulations in MATLAB. First, we simulate the packet loss process  $I(t)$  as an alternating process consisting of two Gamma random processes  $\{A_i\}$  and  $\{B_i\}$  according to (4). We then sample the resulting process  $I(t)$  using a Poisson sampler with rate  $\lambda$  to obtain  $N = 10\,000$  samples  $I_1, \dots, I_N$  from which we apply our model and inference method to find

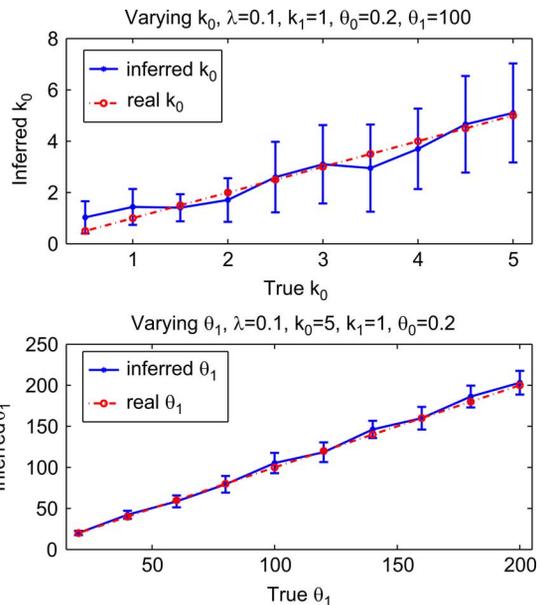


Fig. 2. Parameter estimates versus ground truth.

the following: the loss ratio, its variance, and the parameters of the Gamma distribution of  $A_i$  and  $B_i$ . We compare these results against the real values varying  $\lambda$  and  $w = \{(\theta_0, k_0), (\theta_1, k_1)\}$  in our simulations to show that our SAIL algorithm produces accurate results (when the underlying model is correct).

### B. Results

1) *Parameter Estimation*: The first feature we must validate is that we can correctly identify the parameters of our model using the SAIL algorithm. Fig. 2 shows estimates of model parameters against the real parameters. In the top plot, we vary  $k_0$ , and in the lower plot, we vary  $\theta_1$  (keeping other parameters fixed). We repeat each simulation 10 times and report the average values of the inferred parameters and their confidence intervals. We have performed other experiments with other parameters with similar results (omitted due to space restrictions).

The obvious features of these plots are that the parameter values are all estimated accurately within the limits of statistical accuracy of our measurements. It is, however, noteworthy that our estimates of the scale parameter  $\theta$  have much less variability than those of the shape parameter  $k$ . The obvious question then is “Does this significantly effect the accuracy of estimates derived from these parameters?”

2) *ON/OFF Durations*: In order to answer the previous question, we examine how accurately we can estimate the average ON and OFF durations for our process. In Fig. 3, we plot the estimated mean ON/OFF durations for different values of  $\lambda$ . We compare our estimates to naive estimates based simply on the empirical sampled ON/OFF process  $\{Y'_i, Z'_i\}$ . Both methods generally improve as the sample rate  $\lambda$  increases because the probes miss fewer short ON/OFF intervals. However, for small  $\lambda$ , the naive approach dramatically overestimates interval lengths because it amalgamates many intervals together. On the other hand, the performance of our SAIL algorithm is reasonable and not systematically dependent on the sample rate. The results show that

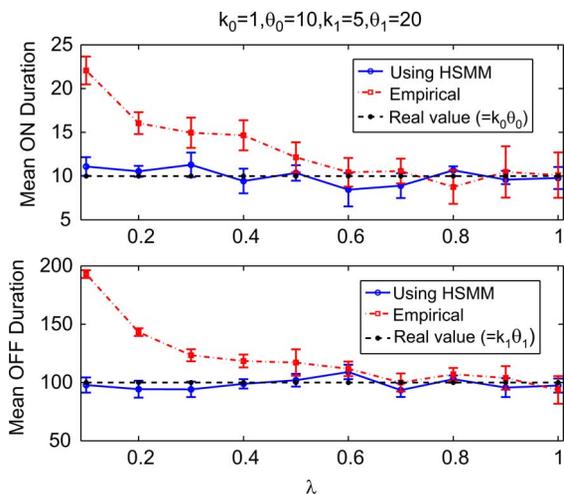


Fig. 3. Estimated lengths of the ON/OFF intervals for different sampling rates.

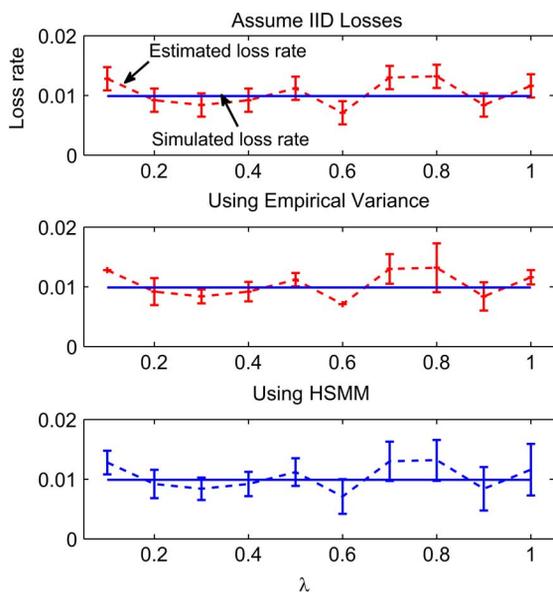


Fig. 4. Inferred loss ratio and its confidence intervals for different sampling rates.

when measuring the ON/OFF durations of a loss process, it is certainly important to compensate for the missing intervals using methods like SAIL.

3) *Comparison to Naive Methods*: Is all of the above work necessary? It would be much easier to estimate the accuracy of  $\hat{p}$  using the Bernoulli estimate (assuming independent samples) or an estimate based on direct empirical estimates of  $R(\tau)$ . In this section, we show that those two methods make loss estimates appear overly optimistic, so that we do require a method like SAIL.

Fig. 4 shows estimates of  $\hat{p}$  along with 95th percentile confidence intervals over a range of probing rates  $\lambda$  with  $k_0 = 5$ ,  $\theta_0 = 0.2$ ,  $k_1 = 1$ , and  $\theta_1 = 100$ . When we assume losses are IID (top plot), the confidence intervals are too small—they do not overlap the true measurement in 40% of cases (as opposed to the 5% expected). The results demonstrate that it is critical to incorporate correlations among the probes into the computation of the confidence intervals.

The middle plot of Fig. 4 shows the results obtained when we use a direct empirical estimate of  $R(\tau)$  to attempt to model correlations. Surprisingly, the confidence intervals in this case perform even worse than using the IID assumption. The reason for this result comes from the fact that by directly estimating the autocovariance function from the samples, we overfitted the data, and these estimations themselves contain errors. Indeed, in 2 out of 10 cases, the empirical variance fails to produce positive results, and we have to approximate them to 0.

Finally, the bottom plot of Fig. 4 shows that the variance computed using the HSMM model and our SAIL algorithm is the most accurate. The 95% confidence intervals contain the real loss rate in 9 out of 10 cases (and is very close in the other case). Certainly, these estimated confidence intervals are more useful than the alternatives.

Note also in this bottom plot that as the sampling rate  $\lambda$  increases, the length of confidence intervals grow. As the probe rate  $\lambda$  increases, we keep the number of probes constant ( $N = 10\,000$ ), and so the probes are more closely packed. This increases the correlations between each measurements, reducing their individual value, leading to less accurate estimates. Neither of the two previous approaches were able to duplicate this effect.

## VII. REAL INTERNET PACKET TRACES

### A. Data Sources

We apply SAIL to four different datasets: 1) loss measurements between dedicated measurement machines; 2) loss measurements between PlanetLab hosts; 3) ping measurements to 100 Web servers in different countries; and 4) DNS server loss measurements obtained by QUEEN [23]. The first two datasets are used to test SAIL under network losses, whereas the other two are used to test SAIL with losses at the DNS and Web servers. Further details of each dataset are described below.

#### 1) Network Losses:

1) *Dedicated measurement hosts*: We set up two measurement hosts, one in Australia and one in Switzerland. These are standalone, special-purposed machines. The machine load is low, with CPU utilization always below 1% and memory utilization below 5%. We send 10 000 Poisson probes between the two machines with a fixed rate  $\lambda = 10$  packets/s. The probes are UDP packets with a payload of 40 B. This payload is used to store the packet sequence numbers, from which the receiver can identify the lost packets. At the sender, we keep track of the sending times of each of the probes. The probe outcomes and probe sending times are used as input to the SAIL algorithm. There are almost no other network activities in these machines except the probes during the measurement period. We collected in total 100 packet traces (each with 10 000 probes) between these two machines from March 22–27, 2009. We call these traces the *dedicated measurement traces*.

2) *PlanetLab data*: To validate SAIL on machines in different locations and under different network conditions, we also performed loss measurements between nodes on PlanetLab [25]. We first select the source and destination

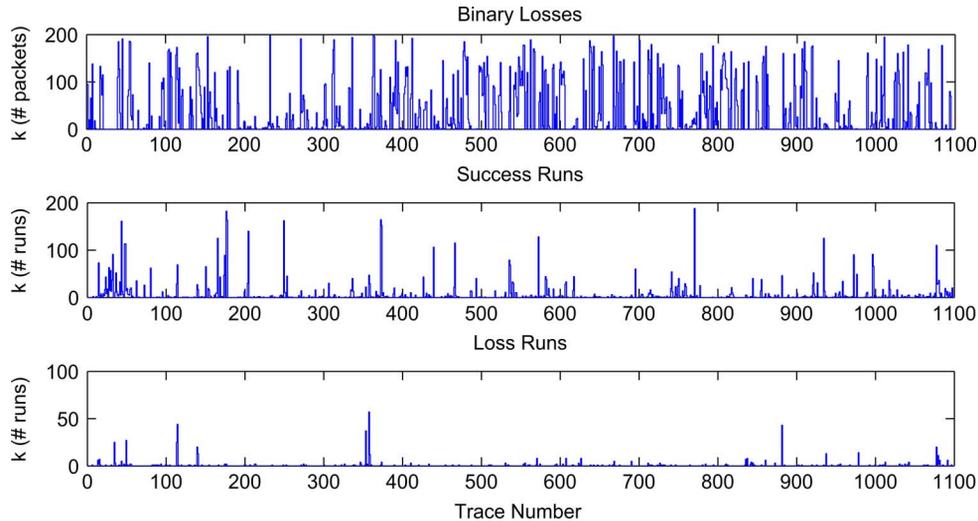


Fig. 5. Correlation timescale  $k$  of the (top) binary loss process ( $\{I_i\}_{i=1}^N$ ), (middle) success run lengths ( $\{Z_i^l\}_{i=1}^l$ ), and (bottom) loss run lengths ( $\{Y_i^l\}_{i=1}^l$ ).

pair randomly among 300 hosts at different sites on PlanetLab, and then send Poisson probes between them similar to the probes used in the dedicated measurements. The traces were obtained during a one-month period between August 7 and September 7, 2008. We call these traces the *PlanetLab* traces. Note that there are several potential problems with loss measurements on PlanetLab as shown in [21]. Most notably, overloaded hosts can drop packets so that losses observed by the probes may not only reflect network losses, but also losses that occur at the host. Even though this artifact of PlanetLab machines affects the ability to measure network losses, it has little impact on the validation of our inference technique. We aim to estimate the accuracy of end-to-end losses, regardless of the causes.

2) *Web Server Losses*: We test the accuracy of SAIL under different loss mechanisms using loss measurements between a host in Australia and the 100 most popular Web sites according to *www.alexacom*, each from a different country. We sent 10 000 ping packets from our measurement host to the Web servers during the 5-days period from August 23–27, 2009. To overcome the ICMP filters that are widely used at Web servers, we implemented our ping probes using TCP SYN packets on port 80. The probes were sent as a Poisson process with rate  $\lambda = 10$  packets/s. In total, we obtained 100 *web server* loss traces.

3) *DNS Server Losses*: We also use the dataset provided by QUEEN [23]. These traces measure one-way losses between 146 different DNS servers all over the world. The data were downloaded from <http://cis.poly.edu/~angelawang/projects/lossrate.htm> on July 20, 2009. Overall, we obtained 511 *DNS server* loss traces.

## B. Validating the ON/OFF Model

1) *Stationarity Test*: We first test the traces for stationarity. There is no completely rigorous way to test for stationarity, so we use a simple heuristic test similar to the one used in [26] of checking whether the loss ratio varies significantly in the trace.

We first smooth the loss ratio of the trace using a finite moving average filter with a window size of 1000 packets. We then apply two tests to the smoothed trace to test for stationarity. First, if there is any abrupt increase or decrease of greater than 0.05, the trace is classified as nonstationary. Second, to test the gradual trend in the loss ratio, we fit a straight line to the data using least-squares. If there is a total change in the average loss ratio of 0.15 or greater, over a 1-hour trace segment, the trace then is considered to be nonstationary.

In total, we obtain 5346 stationary traces for network losses (combining both dedicated measurement and PlanetLab datasets). We also remove traces with negligible loss for obvious reasons, leaving 1090 PlanetLab traces and 10 dedicated measurement traces with loss rates between 0.01% and 10%. We obtain 23 stationary Web server traces with nonnegligible loss ratios (between 0.01% and 20%) and 42 stationary DNS traces with loss ratios between 0.01% and 50%.

2) *Correlation Timescale of the Binary Losses, the Success, and Loss Runs*: Testing for ON/OFF renewal properties has been done extensively in previous studies of Internet losses [26], [29]. It requires testing to see if the ON/OFF periods are independent. In this section, we employed the autocorrelation function (ACF) test used in [26] and [29] to test for the renewal properties of our traces. For a sequence of numbers, e.g., ( $\{I_i\}$ ,  $\{Y_i\}$ , or  $\{Z_i\}$ ), the ACF test can be used to find the correlation timescale  $k$ , which is the smallest time lag  $k$  such that there is no significant statistical evidence to conclude that samples of distance  $k$  or more are correlated. When the samples are completely uncorrelated,  $k = 0$ .

We first apply the ACF test to compute the correlation timescale of the binary data. In Fig. 5 (top), we plot the correlation timescale for the 1100 stationary traces in our network loss data (10 from the dedicated measurements and 1090 from PlanetLab). Only 25% of the traces have correlation timescale  $k = 0$ , which means that the samples are uncorrelated. The samples in the other traces exhibit correlation at different timescales. Some are correlated up to a lag of 200 packets.

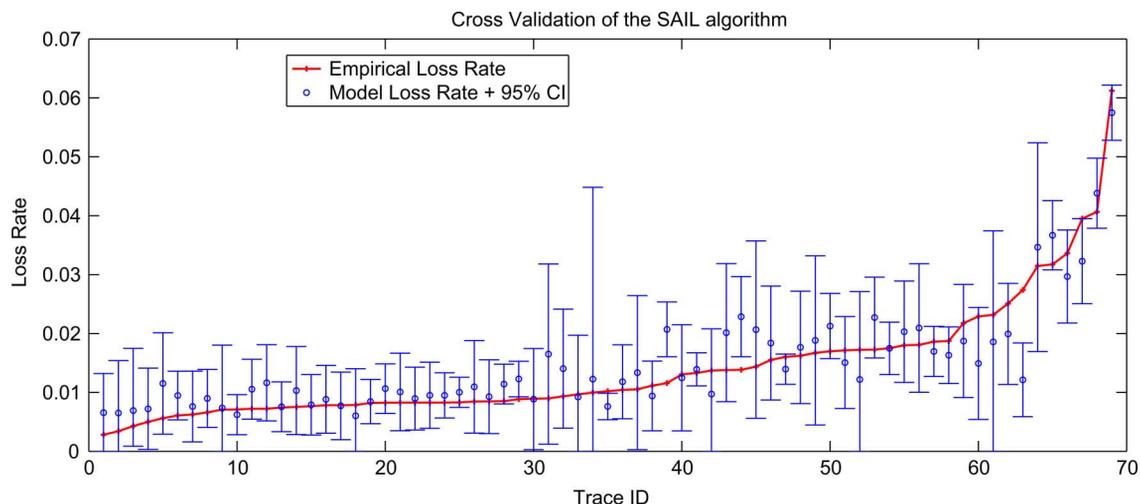


Fig. 6. Cross validation for loss ratio estimated using SAIL for network losses.

These results are consistent with the findings in [26] and [29], but the maximum correlation timescale in our traces is higher than those in [26].

We also plot the correlation timescale  $k$  of the observed loss run lengths  $Y'_i$  and success (or no-loss) run lengths  $Z'_i$  in Fig. 5. We observe that loss run lengths in Fig. 5 (bottom) are uncorrelated in all but a small fraction of our traces. The correlation timescale of the success run lengths are plotted in Fig. 5 (middle). Success run lengths are uncorrelated ( $k = 0$ ) in about 60% of the traces.

Most significantly, in most of the traces where the binary losses have high correlation timescales, the correlation timescales of the loss and success runs remain very small. Therefore, most of the significant correlation in the binary losses is caused by nearby losses in the same loss run rather than interrun correlation.

Finally, we look at the correlation between loss run lengths  $\{Y'_i\}_{i=1}^L$  and success run lengths  $\{Z'_i\}_{i=1}^L$ . Again in about 80% of the traces, the cross correlation of the pair  $\{(Y'_i, Z'_i)\}_{i=1}^L$  is very weak (the correlation coefficient between  $Y'_i$  and  $Z'_i$  is between  $-0.2$  and  $0.2$ ), indicating that the lengths of adjacent loss runs and successful runs bear little relationship to each other. Hence, even though the binary losses are far from independent, an alternating renewal process is reasonable for the majority of our traces.

The Web server and DNS server loss traces have similar correlation structure, but with higher correlation timescale  $k$  for the binary losses. We omit the details of these results.

### C. Cross Validation of the SAIL Algorithm

Unlike simulations where real loss ratios are known, in packet traces we do not know the underlying loss process. In this case, the best method to test our SAIL algorithm is to use cross validation. For each of the traces, we break the trace into two subsamples: one called the *inference trace* and the other the *cross-validation trace*. We divide the trace into two parts randomly so that each sample has an even probability of being put into either subsample. The subsampling property of the Poisson process means that each of these subsamples forms a Poisson

process as well, with rate  $\lambda/2$ , so we can apply all of our techniques to the traces. The PASTA property suggests that the two traces should (on average) report the same loss ratio, and that the loss ratio of one trace should lie in the 95% confidence interval of the other trace roughly 95% of times. We use this fact to test the accuracy of our estimates for the loss ratio variances.

We apply the SAIL algorithm to the inference trace to compute the loss rate and its variance with the length of the discrete time interval  $\Delta t = (2\lambda)^{-1}$ . We then compute the empirical loss ratio of the cross-validation trace. We plot in Fig. 6 the loss ratio and its 95% confidence interval provided by the SAIL algorithm for the inference trace. We also plot on the same figure the empirical loss ratio of the validation trace. In order to make the figure comprehensible, we report on 69 randomly chosen stationary traces (similar results are seen in the other traces).

Two important observations can be made. First, the figure clearly shows that our model gives very accurate estimates of the loss ratio variance. In 65 out of 69 traces (94%), the validation loss ratio falls in the 95% confidence interval derived using SAIL. In the complete dataset, 1012 out of 1100 (92%) fall in the interval. Note that the datasets used for inference and validation are completely separate, so we have satisfied our chief goal of being able to estimate the errors of loss estimates to a reasonable degree of fidelity. Note, however, that the results do not reveal anything about the accuracy of the estimates for the loss ratios.

Second, note that the width of the confidence intervals is large. In many of the traces, the interval is larger than the loss ratio itself. This is a fundamental property of loss ratio measurements, not a failure of SAIL (a fact also supported by results shown in Fig. 11). The intervals must be this wide to encompass the intrinsic variation of loss ratio estimates. The implication is that there is a large uncertainty involved in loss ratio measurements and extreme care needs to be taken in their interpretation. In fact, it is not overstating the case to say that loss ratio measurements are much less useful than typically assumed. Without accurate confidence intervals, it would be easy to come to erroneous conclusions, and this could have dramatic consequences if these measurements were, for instance, used as

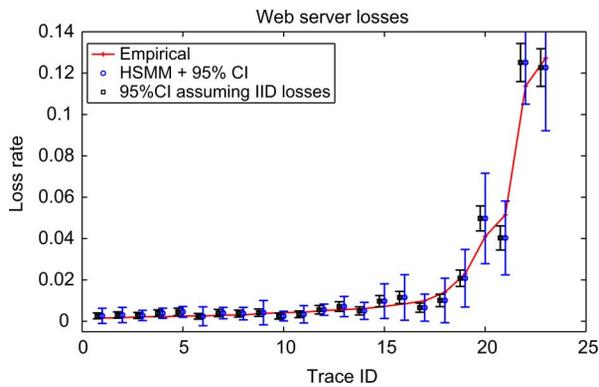


Fig. 7. Cross validation for the Web server losses.

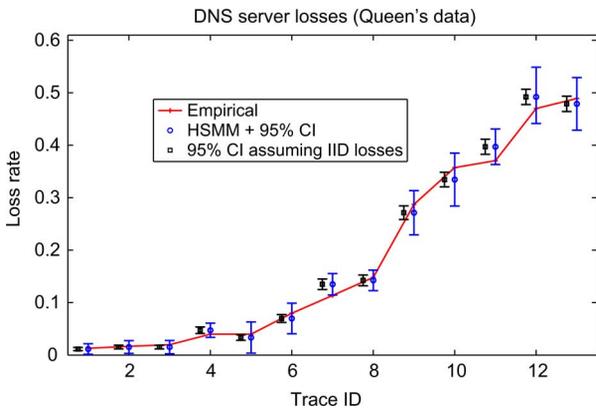


Fig. 8. Cross validation for DNS server losses.

evidence in contractual litigation (say for purported failure to satisfy an SLA).

Similar observations can be made for the Web server and DNS server traces as shown in Figs. 7 and 8. Again, to make the figures comprehensible, we plot the results for all the Web server traces in Fig. 7, but only 1/4 of the DNS server traces (randomly selected) in Fig. 8. In these figures, we also plot the confidence intervals when assuming that losses are IID. It can be observed that the confidence intervals under the IID assumption are too small—they do not overlap the cross-validation loss ratios in 9 out of 23 Web server traces and 5 out of 13 DNS server traces (shown in Fig. 8) (or 15 out of all 42 DNS server traces). We also observe that as the loss ratio increases, the confidence intervals grow larger (where needed) to capture the stronger correlation among the packet losses. Only the SAIL algorithm can capture this effect.

The variance can be computed either directly from the probes (empirical variance) or using the HSMM model. We plot in Fig. 9 the variance computed using these two different methods for the inference and cross-validation traces. In Fig. 9 (left), we compare the empirical variances of the two subtraces (for the 69 1-hour-long traces used in Fig. 6). We observed that the empirical variances are almost completely unrelated to each other. By this measure, the two subtraces (from the same path, during the same time interval) appear completely unrelated. In contrast, the variances estimated by the SAIL algorithm using the HSMM model of the inference and cross-validation traces largely agree

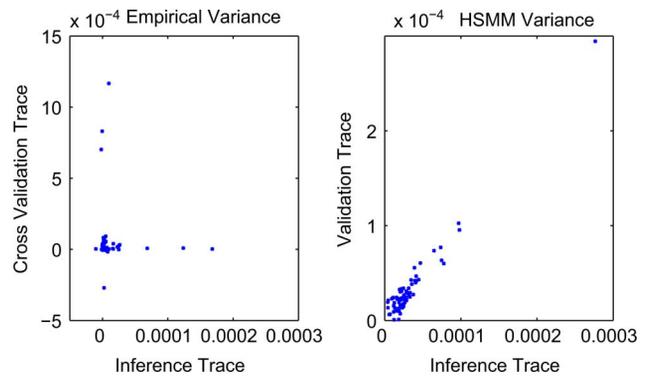


Fig. 9. Cross validation of the variance.

[see Fig. 9 (right)]. The figures illustrate that while it is important to take the correlations between the probes into account when computing the variance of the loss ratio, it is equally important to use an appropriate method, such as SAIL, to compute this variance.

#### D. Loss Ratio Prediction

Loss measurements are often used to predict future behavior. In this section, we test that ability. The ability that SAIL gives us to estimate accurate confidence intervals for our estimates means that our results are the first that can rigorously detect the difference between natural statistical variation in estimates and errors in prediction.

We apply the technique to our 1100 stationary network loss traces (prediction can be performed on nonstationary traces, but requires more sophisticated estimators than we have space to develop here). Note that if these traces were truly stationary, then our predictions would be guaranteed to work, but our method for testing stationarity is crude. It uses arbitrary thresholds and cannot detect changes in higher-order moments. Hence, the traces we examine might be considered “stationary to the naked eye.” The question is, “Can we make valid predictions on such crudely stationary traces, much as an operator might in practice?”

For each of the 1100 stationary network loss traces, we divide the traces into two subtraces. The first subtrace consists of the first 5000 samples of the original trace, and the second subtrace consists of the next 5000 samples of the original traces. We use the first segment to infer the loss ratio and its variance using the SAIL algorithm. We then compute the loss ratio of the second segment and plot the results in Fig. 10 for the 69 traces that are used in Fig. 6.

We observe that the loss ratio falls within the 95% confidence interval of the computed loss ratio in 54 out of 69 traces (860 out of 1100 stationary network loss traces). Clearly, there are some deviations in the loss ratio predictions beyond normal statistical variations, however the results suggest that prediction is not impractical.

#### E. Parameter Values for Measurement Traces

We have shown in the previous sections that SAIL can accurately estimate the parameters of an alternating loss process whose durations follow Gamma distributions. In this section,

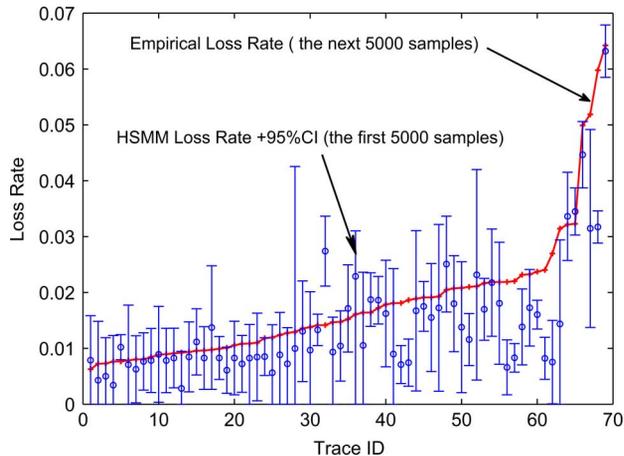


Fig. 10. Cross validation for the predictive power of SAIL. Measurements in the first 5000 samples are used for inference; measurements in the next 5000 samples are used for cross validation.

TABLE I

MEAN (VARIANCE) OF THE PARAMETER VALUES FOR REAL PACKET TRACES. THE LENGTH OF THE DISCRETE TIME-SLOT IS  $\Delta t = 1/(2\lambda)$ , WHERE  $\lambda$  DEPENDS ON THE PACKET TRACES AS DESCRIBED IN SECTION VII-A

Dataset	$k_0$	$\theta_0$	$k_1$	$\theta_1$
Dedicated	2.3 (4.9)	1.6 (5.8)	8.7 (5.2)	21.0 (7.2)
PlanetLab	3.0 (3.7)	0.6 (0.3)	2.8 (2.0)	9.0 (8.4)
Web	2.1 (4.4)	3.8 (2.3)	4.9 (5.8)	52 (13.7)
DNS	1.0 (2.3)	6.1 (3.2)	4.7 (23.7)	21.2 (70.9)

we apply SAIL to the measurement traces in Section VII-A to find the Gamma parameters for their loss and no-loss durations. These values are useful in generating realistic loss processes. We provide in Table I the mean and the variance of the parameters for each of the datasets. Recall that the parameters are scaled by  $\Delta t = 1/(2\lambda)$ . As the traces have different sampling rates, the parameters may be scaled differently between traces.

In all of our datasets, the parameters have large variances showing that the underlying loss process varies significantly between traces in the same dataset. These variations are expected as the datasets span several days and are between hosts in different geographical locations. The mean values of the parameters also vary between different datasets, reflecting the fact that the actual causes of the losses are not the same (in-network losses, hosts induced, etc.). All traces except the DNS traces have large  $k_0$  and  $k_1$ . Thus, the loss and no-loss runs in these traces have Gaussian-like distribution. The length of the loss burst for the DNS traces, however, follows an exponential distribution ( $k_0 = 1$ ). The DNS traces also have large variances for  $\theta_1$ .

#### F. How Many Probes?

The final question we seek to answer is that of ‘‘How many probes?’’ Formally, this is a (statistical) experimental design question. The inputs should be the desired accuracy of the estimates and the model parameters. However, we have observed a wide range of model parameters, and it is not obvious, *a priori*, how we should set these. Hence, we seek practical guidance as to how to choose probes rates.

We explore this question by once again subsampling from our traces with probability  $p_s$  to obtain a new Poisson probe sample

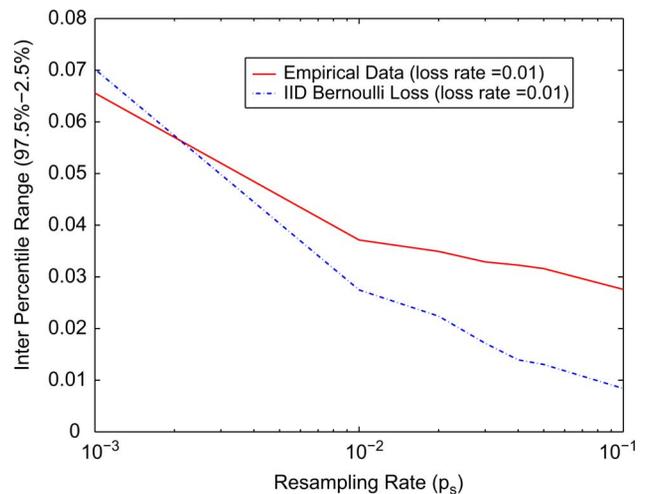


Fig. 11. Variation in loss ratio estimate for different number of samples.

with rate  $\lambda p_s$ . We repeat the resampling procedure 1000 times for each resampling rate  $p_s$  to obtain 1000 (Poisson) subsamples of the same probe sequence. We then look at the empirical distribution of loss estimates for this sequence and measure the width of the distribution by finding the difference between the 2.5th and 97.5th quantile. The width stands in as a proxy for the accuracy of the measurements. Note that it is not dependent on SAIL, but only on the empirical distribution observed for the 1000 subsamples of our original data.

Fig. 11 shows the results for one trace where the loss is moderate (about 1%) from our dedicated measurement dataset. The width is shown as a function of  $p_s$ . We also show the expected width of the distribution for Bernoulli (independent) loss with the same loss rate. We once again see that the width is large, so the natural variability of loss ratio measurements is large. However, the feature of particular interest in this plot is the slow decrease in the width as the sample rate increases (note the log  $x$ -axis). The width decreases significantly more slowly than for the Bernoulli process. The intuitive explanation is that as we probe at higher rates, the probes are closer together, and so they become more correlated. More correlated data give less information, and so the value of these extra probes is smaller than we might hope. It is likely that there is a fundamental bound here, similar to that in [18], below which we cannot go.

The implication is that one should accept that loss measurements are (intrinsically) inaccurate and then one should probe at a low rate. Extra probes should only be done if there is a compelling case for accuracy, and the probes are undertaken with the understanding that their cost/benefit ratio will be large.

## VIII. CONCLUSION AND FUTURE WORK

We have developed SAIL, a statistically accurate Internet loss measurement technique. SAIL employs sophisticated modeling methods from the hidden Markov model literature to overcome the missing intervals problem that commonly occurs when sampling a loss process at a low rate. SAIL first accurately infers the parameters of the underlying loss process and then uses them to compute the loss ratio and along with its confidence interval. We

show that SAIL outperforms alternative methods in both simulation and real packet traces. All code and data are available at <http://bandicoot.maths.adelaide.edu.au/SAIL/>.

SAIL can be improved in several aspects. First, in this paper, we use a parametric model for the ON/OFF durations. The HSMM algorithm used in SAIL can be extended to nonparametric distributions. Exploring the advantages and disadvantages of the nonparametric models is an interesting future direction. Second, the renewal assumption can be relaxed. However, it would require a more complicated model than the HSMM. Keeping the complexity of such a model low will be challenging.

Finally, it would be a great advantage to be able to compute SAIL online, continuously updating our loss ratio estimates as new loss measurements arrive. This is possible, but will require further implementation work.

REFERENCES

[1] G. Almes, S. Kalidindi, and M. Zekauskas, "A one-way packet loss metric for IPPM," IETF, RFC 2680, 1999.  
 [2] J. Andren, M. Hilding, and D. Veitch, "Understanding end-to-end Internet traffic dynamics," in *Proc. IEEE GLOBECOM*, Sydney, Australia, 1998, vol. 2, pp. 1118–1122.  
 [3] F. Baccelli, S. Machiaju, D. Veitch, and J. Bolot, "The role of PASTA in network measurement," in *Proc. ACM SIGCOMM*, Sep. 2006, pp. 231–242.  
 [4] F. Baccelli, S. Machiaju, D. Veitch, and J. Bolot, "On optimal probing for delay and loss measurement," in *Proc. ACM SIGCOMM Internet Meas. Conf.*, Oct. 2007, pp. 291–302.  
 [5] J. C. Bolot, "End-to-end packet delay and loss behavior in the Internet," in *Proc. ACM SIGCOMM*, 1993, pp. 289–298.  
 [6] D. R. Cox and H. D. Miller, *The Uppercase Theory of Stochastic Processes*. London, U.K.: Chapman & Hall, 1965.  
 [7] E. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, pp. 1253–1265, Sep. 1960.  
 [8] W. Jiang and H. Schulzrinne, "Modeling of packet loss and delay and their effect on real-time multimedia service quality," in *Proc. NOSSDAV*, Jun. 2000, Paper no. 37.  
 [9] R. Koodli and R. Ravikanth, "One-way loss pattern sample metrics," IETF, RFC 3357, 2002.  
 [10] B. Melamed and D. Yao, *Advances in Queueing: Theory, Methods and Open Problems. The ASTA Property 1995*. Boca Raton, FL: CRC Press, 1995.  
 [11] R. E. Mortensen, "Alternating renewal process models for electric power system loads," *IEEE Trans. Autom. Control*, vol. 35, no. 11, pp. 1245–1249, Nov. 1990.  
 [12] H. X. Nguyen and M. Roughan, "On the correlation of Internet packet losses," in *Proc. ATNAC*, 2008, pp. 22–27.  
 [13] H. X. Nguyen and M. Roughan, "SAIL: Statistically accurate Internet loss measurement," in *Proc. ACM SIGMETRICS*, Jun. 2010, pp. 361–362.

[14] V. Paxson, "End-to-end Internet packet dynamics," in *Proc. ACM SIGCOMM*, Sep. 1997, pp. 139–152.  
 [15] V. Paxson, A. Adams, and M. Mathis, "Experiences with NIMI," in *Proc. PAM*, 2000, pp. 87–97.  
 [16] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis, "Framework for IP performance metrics," IETF, RFC 2330, 1998.  
 [17] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.  
 [18] M. Roughan, "Fundamental bounds on the accuracy of network performance measurements," in *Proc. ACM SIGMETRICS*, Jun. 2005, pp. 253–264.  
 [19] K. Salamati and S. Vaton, "Hidden Markov model for network communication channels," in *Proc. ACM SIGMETRICS*, 2001, pp. 92–101.  
 [20] H. Sanneck, G. Carle, and R. Koodli, "A framework model for packet loss metrics based on loss runlengths," in *Proc. SPIE ACM Multimedia Comput. Netw. Conf.*, 2000, pp. 177–187.  
 [21] J. Sommers and P. Barford, "An active measurement system for shared environments," in *Proc. ACM SIGCOMM IMC*, Oct. 2007, pp. 303–314.  
 [22] J. Sommers, P. Barford, N. Duffield, and A. Ron, "Improving accuracy in end-to-end packet loss measurement," in *Proc. ACM SIGCOMM*, Aug. 2005, pp. 157–168.  
 [23] A. Wang, C. Huang, J. Li, and K. W. Ross, "QUEEN: Estimating packet loss rate between arbitrary Internet hosts," in *Proc. PAM*, 2009, pp. 57–66.  
 [24] A. Weideman, "Algorithms for parameter selection in the Weeks method for inverting the Laplace transform," *SIAM J. Sci. Comput.*, vol. 21, pp. 111–128, 1999.  
 [25] "PlanetLab," 2012 [Online]. Available: <http://www.planet-lab.org>  
 [26] M. Jainik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modelling of the temporal dependence in packet loss," in *Proc. IEEE INFOCOM*, 1999, vol. 1, pp. 345–352.  
 [27] S.-Z. Yu, "Hidden semi-Markov model," *Artif. Intell.*, vol. 174, pp. 215–243, Nov. 2009.  
 [28] S.-Z. Yu and H. Kobayashi, "A hidden semi-Markov model with missing data and multiple observation sequences for mobility tracking," *Trans. Signal Process.*, vol. 83, pp. 235–250, Dec. 2003.  
 [29] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, "On the constancy of Internet path properties," in *Proc. ACM SIGCOMM IMW*, San Francisco, CA, 2001, pp. 197–211.

**Hung X. Nguyen** (M'11) received the Ph.D. degree in computer science from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, in 2008.

He is a Research Fellow with the School of Mathematical Sciences, University of Adelaide, Adelaide, Australia. His research interests include network measurements, tomography, and privacy-preserving techniques.

**Matthew Roughan** (M'97–SM'09) received the Ph.D. degree in applied probability from the University of Adelaide, Adelaide, Australia, in 1994.

He joined the School of Mathematical Sciences, University of Adelaide, in 2004. Prior to that, he was with AT&T, Florham Park, NJ. His research interests lie in measurement and modeling of the Internet, and his background is in stochastic modeling.