Modeling Time Correlation in Passive Network Loss Tomography

Jin Cao
Bell Labs, Alcatel-Lucent
Murray Hill, New Jersey
NJ 07974, USA
Email: cao@research.bell-labs.com

Aiyou Chen
Google Inc
Mountain View, CA, USA
Email: aiyouchen@google.com

Patrick P. C. Lee
Dept of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, Hong Kong
Email: pclee@cse.cuhk.edu.hk

***Abstract*—We consider the problem of inferring link loss rates using passive measurements. Prior inference approaches are mainly built on the time correlation nature of packet losses. However, passive inference generally has limited control over the measurement process, and it is a challenging issue to adapt loss rate inference to the impact of time correlation. We address this issue and propose a new loss model that expresses an inferred link loss rate as a function of time correlation. Under this loss model with time correlation, we show its identifiability, and propose a novel profile-likelihood-based inference approach that can accurately infer link loss rates for various complex topologies (e.g., trees with many leaf branches). We validate the accuracy of our inference approach with model and network simulations.**

***Index Terms*—network tomography; passive loss rate inference; time correlation; measurement and monitoring techniques; performance evaluation and assessment**

I. Introduction

Network loss tomography is an important technique that computes statistical estimates of internal loss rates of network elements through external data traffic measurements. This enables operators to readily diagnose performance bottlenecks of an operational network during the periods of high traffic loads, and hence devise better strategies for resource provisioning and network planning. Traditional approaches of loss rate inference use *active probing*, which sends probes, either in multicast (e.g., [4], [9], [23]) or unicast (e.g., [11], [12]), to obtain end-to-end measurements for inference. Since most Internet routers do not enable multicast, unicast-based active probing [11], [12] is more appealing in practice. It mainly leverages the *time correlation* nature of packet losses that is typically seen in the Internet (e.g., see analysis in [19]). That is, neighboring packets likely experience the similar loss behavior on the common node/link that they traverse. For example, a router queue may be overflowed during congestion, and this leads to consecutive losses of arriving packets over a time window. Based on the statistical feature of time correlation, loss inference algorithms can then be developed.

However, active probing has its own limitations, such as introducing probing overhead and requiring collaboration of senders and receivers to collect measurement data. Hence, *passive network tomography* is proposed, such that loss inference is achieved by monitoring existing traffic (e.g., using TCP traffic [6], [21], [22]) without generating probing traffic. Like unicast-based active probing, passive network tomography is built on the time correlation nature of packet losses.

While passive network tomography is attractive, it has limited control over the measurement process, and this results in the degradation of the overall inference accuracy. In particular, time correlation of packet losses is imperfect in practice [12]. While active probing techniques can reduce the correlation defect using stripes of probes [12], such a solution cannot be used in passive approaches. It is expected that if back-to-back packets have a larger time difference, then their time correlation will decay more. However, it remains a challenging issue of how to model the decaying property of time correlation in loss rate inference, especially in passive approaches where the measurement process has less control.

In this paper, we propose a passive loss rate inference approach that accounts for different functional forms of time correlation of packet losses. We first model an inferred link loss rate as a function of time correlation. Under this loss model with time correlation, we develop an inference approach based on the *profile likelihood* (PL) method [18], with which we can focus on the parameters of interest (i.e., link loss rates to be inferred) by replacing other nuisance unknowns with appropriate estimates. Our loss rate inference approach is applicable in existing passive network measurement systems and enables them to provide improved inference accuracy.

To further motivate the importance of accurate passive loss rate inference, we consider a commercial network traffic analysis system that we know has implemented simple loss inference techniques. The analysis system passively captures IP traffic of a wireless data network and monitors the performance of the underlying network links and elements. To illustrate, Figure 1 shows a simplified view of a wireless data network and how the analysis system is deployed. The wireless data network embodies a *hierarchical* structure, in which the analysis system observes IP traffic at a Home Agent/Foreign Agent (HA/FA) tap point that is connected with multiple base station controllers (BSCs), and each BSC is connected with multiple base stations (BSes). Typically, a BSC is connected to a large number of BSes, and the traffic workload toward each base station is generally unevenly distributed. In view of this, we include statistically efficient extensions into our baseline

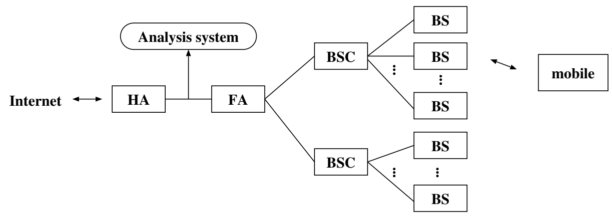


Fig. 1. A simplified wireless data network architecture. A traffic analysis system is deployed to infer the performance of network links and elements. Our passive inference approach is applicable for this type of traffic analysis.

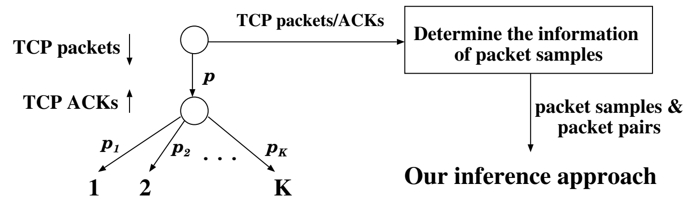


Fig. 2. The TCP-based passive measurement framework for network loss tomography. Our goal is to infer the loss rate of the common link from the root node to the middle node. We let p and p_i ($i = 1, \cdots, K$) be the link success rates of the common link and leaf branch i, respectively, and they will be used in Section III.

PL-based inference approach to account for a "many-branch" topology, in which some nodes are connected to a high degree of branch links that may have uneven traffic loads.

In summary, we make the following contributions:

- We develop a loss model that expresses loss rates as a function of time correlation. This enables us to study the impact of different time correlation functions on the inference results.
- We consider different functional forms of time correlation, and show that our loss model is identifiable as long as the correlation is not a constant other than 1.
- Under the loss model with time correlation, we develop a PL-based inference approach that provides accurate estimates of link loss rates.
- We extend our PL-based inference approach for more general topologies, including topologies with a high degree of branch links and multi-level topologies.
- We verify the accuracy of our inference algorithm through extensive model simulations in R [2] and network simulation in ns2 [1].

The paper proceeds as follows. In Section II, we present a passive TCP monitoring framework on which we develop our inference approach. Section III models time correlation in loss inference and discusses the identifiability issues. Section IV proposes our PL-based inference approach, and Section V discusses the extensions for general topologies. Sections VI and VII validate our inference approach via model and network simulations, respectively. We discuss related work in Section VIII, and conclude in Section IX.

II. Passive Measurements Through TCP Inference

We first introduce a basic TCP-based passive measurement framework on which we can deploy our inference approach. We also state the assumptions of our framework.

Figure 2 depicts the TCP-based passive measurement framework [3], [6], [21], [22]. The framework observes sender-to-receiver TCP data packets (or packets for short) and receiver-to-sender acknowledgments (ACKs) at a measurement point. It seeks to identify whether each packet is delivered or lost from the sender to the receiver. Determining whether a TCP packet is lost is based on the TCP retransmission mechanism: if a sender successfully delivers a packet to a receiver, then the receiver will reply a corresponding ACK; if the sender does not receive the ACK after a timeout, then it deduces that the packet is lost and retransmits the same packet. Thus, intuitively, packet retransmissions can be viewed as indicators of packet losses. Since retransmissions can also be triggered by spurious timeouts or ACK losses in addition to packet losses, there are also more robust approaches of inferring TCP packet losses (e.g., see [3]).

It is important to note that our focus in this paper is *not* on determining TCP packet losses, but instead we aim to leverage the current passive inference frameworks as a platform and enhance the accuracy of network loss tomography (i.e., inferring loss rates of network elements including nodes and links). Thus, we assume that we can correctly determine whether a TCP packet is lost somewhere along a path by observing TCP traffic, and use this end-to-end information as the inputs to our inference approach.

To simplify our discussion, Figure 2 depicts only a two-level tree topology, in which TCP packets traverse along a common link and reach one of the K leaf branches with index i, where $i = 1, 2, \cdots, K$. We collect packet samples at the root node, and infer the loss rate of the common link (on the root-to-leaf direction). While we focus on link losses, the same methodology is also applicable to losses in nodes (e.g., router queues). In Section V, we explain how to extend our inference approach for general topologies. In addition, we expect that this type of loss rate inference is deployed in a network that is administered by a single authority, so the topological information is available.

The framework monitors a stream of TCP packets and ACKs over a measurement epoch. For each packet sample, the framework obtains the following: (i) the arrival timestamp when the measurement point observes the packet, (ii) the index of the leaf branch that the packet will traverse, (iii) the Boolean variable of whether a packet is lost. Such information is used to generate *packet pairs* (see Section III). Both sets of single packet samples and packet pairs will be the inputs to our inference approach, as will be discussed in later sections.

III. Packet Pair Loss Modeling

In this section, we discuss what constitutes a packet pair, and present our loss model for these packet pairs. Our goal is to utilize these packet pair measurements, together with single packet measurements, to estimate link success (or loss) rates. We demonstrate theoretically that our model parameters (success/loss rates) are statistically identifiable. We also formally justify how the perfect correlation assumption can introduce bias to our estimates.

A. Packet Pair

We use the two-level tree topology in Figure 2 for our discussion. We define a *packet pair* as the neighboring packets that have inter-arrival time less than δ, where δ is a tunable parameter, and are destined for different leaf branches. While such a definition is more restricted than that in [11], in which all packet pairs, whether having the same branch or not, are considered, we will show later in Section IV that such restrictions enable us to use a much smaller set of model parameters to greatly simplify our loss rate estimation.

B. Loss Model

Let $(\mathcal{U}, \mathcal{V})$ be a packet pair, where $\mathcal{U}, \mathcal{V}$ represent the first and second packets in the packet pair, respectively. We now model the loss correlation between $\mathcal{U}$ and $\mathcal{V}$.

As the packet pair $(\mathcal{U}, \mathcal{V})$ is destined for different leaf branches, we can always define the sub-path before the divergence of their paths as the *common link*, and define paths after the divergence as new *leaf links*. This forms a tree topology with two leaves. Therefore, to simplify our discussion, we consider a two-level, two-leaf tree topology in Figure 2 (i.e., $K = 2$) to explain our model.

Let packet $\mathcal{U}$ be represented by a tuple (u, Y_u, t_u), where $u = 1, 2$ is the leaf index, and Y_u is the event variable that equals 1 if the packet is received by the leaf branch (i.e., successful) or 0 otherwise, and t_u is the arrival timestamp of packet $\mathcal{U}$ observed at the measurement point. Similarly, we use tuple (v, Y_v, t_v) to represent the packet $\mathcal{V}$. Hence, we can decompose Y_u, Y_v as

$$Y_u = Z_u X_u, \qquad Y_v = Z_v X_v, \tag{1}$$

where Z_u (resp. Z_v) is the event variable that equals 1 if packet $\mathcal{U}$ (resp. $\mathcal{V}$) is successful on the common link and 0 otherwise, and X_u (resp. X_v) is the event variable that equals 1 if packet $\mathcal{U}$ (resp. $\mathcal{V}$) is successful at the leaf link and 0 otherwise.

We assume that loss events on different links are independent of each other, i.e., X_u and X_v are independent events and both of them are independent of the events Z_u and Z_v at the common link. We point out that the assumption of link independence does not strictly hold in general, but the correlation of links is weak and has limited impact on our analysis [20], [21]. On the other hand, since Z_u and Z_v are two events at the common link, they are *correlated*. Let

$$\Delta_u \doteq t_v - t_u$$

be the difference of the arrival timestamps of the packet pair. We assume that the success (and loss) events at the common link form a stationary process with some *correlation function* $\rho_a(\cdot)$, where a represent some unknown parameter, i.e.,

$$Correlation(Z_u, Z_v) = \rho_a(t_v - t_u) = \rho_a(\Delta_u), \tag{2}$$

where $0 \leq \rho_a(\cdot) \leq 1$. It is reasonable to expect that $\rho_a(\cdot)$ is a monotonically decreasing function, $\rho_a(0) = 1$ and $\rho_a(\Delta_u) = 0$ as Δ_u goes to infinity.

Let p be the packet success rate at the common link, and p_1 and p_2 be the link success rates of leaf nodes 1 and 2, respectively (see Figure 2 with $K = 2$). Let $sd(.)$ denote the standard deviation function, and let $Cov(.)$ denote the covariance function. Then (2) implies that the covariance between Z_u and Z_v is:

$$Cov(Z_u, Z_v) = sd(Z_u)sd(Z_v)\rho_a(\Delta_u) = p(1-p)\rho_a(\Delta_u).$$

Furthermore, using (1), we can easily show that

$$Cov(Y_u, Y_v) = p(1-p)p_1p_2\rho_a(\Delta_u). \tag{3}$$

Suppose that we define four possible probabilities of success/failure events of the packet as follows:

$$r_{kl}(\mathcal{U}, \mathcal{V}) \doteq P(Y_u = k, Y_v = l), \quad k, l \in \{0, 1\}. \tag{4}$$

It is easy to show that

$$\begin{aligned} r_{11}(\mathcal{U}, \mathcal{V}) &= pp_1p_2(p + (1-p)\rho_a(\Delta_u)), \\ r_{10}(\mathcal{U}, \mathcal{V}) &= pp_1 - r_{11}, \\ r_{01}(\mathcal{U}, \mathcal{V}) &= pp_2 - r_{11}, \\ r_{00}(\mathcal{U}, \mathcal{V}) &= 1 + r_{11} - pp_1 - pp_2. \end{aligned} \tag{5}$$

C. Modeling Time Correlation

Since the exact form of the time correlation function is generally unknown in advance, it is prudent only to consider an approximation of the function when Δ is close to zero. We consider the following two specific approximations of the time correlation function when $0 \leq \Delta < \delta$ for small δ:

$$\begin{aligned} \rho_a(\Delta) &= \exp(-a\Delta), \quad \text{(linear form)}, \\ \rho_a(\Delta) &= \exp(-a\Delta^2), \quad \text{(quadratic form)}, \end{aligned} \tag{6}$$

where $a > 0$ is an unknown parameter that we want to estimate as will be discussed in Section IV. Since the above approximations hold for small Δ only, we expect that they are good enough to capture the time-decaying property of the exact form of $\rho_a(.)$ (which is unknown in general). Note that when $a = 0$, the above approximations include the special case of the perfect correlation model for all $\Delta \geq 0$:

$$\rho_a(\Delta) = 1 \quad \text{(perfect correlation)}, \tag{7}$$

which implies that both packets in the packet pair always have the same success/loss events. However, if we assume that the packet pair has a perfect correlation but indeed does not, then we show that there is a bias in the estimates (see Section III-E).

It is important to mention that the approach by [11] equivalently adopts the following time correlation function for $\Delta > 0$ (while $\rho_a(0) = 1$):

$$\rho_a(\Delta) = a < 1 \quad \text{(constant form)}. \tag{8}$$

However, we will next show that such a loss model is not statistically identifiable.

D. Model Identifiability

When a packet pair is perfectly correlated, [9] shows that the unknown success rates on links are identifiable. However, if there is no perfect correlation, the identifiability remains an open question. Although the imperfect correlation model has been discussed in [11], the identifiability of model parameters is not rigorously discussed. Here we show that if we take a functional form of the loss correlation model (i.e., the correlation depends on the time difference such as the linear and quadratic models in (6)), then the success rates p, p_1, p_2 are statistically identifiable. However, if we assume that the correlation is constant as in (8), then the success rates may not be identifiable without further assumptions. We will demonstrate this using the simple two-leaf tree.

Theorem 1: Under the loss correlation model in (2), the link loss rates p, p_1, p_2 in the two-leaf tree as well as the correlation model parameter a are identifiable, given that $\rho_a(0) = 1$. ∎

Proof (Sketch): From (3), we see that covariance between Y_u and Y_v (i.e., the success events of the two packets in the packet pair) can be used to determine $p(1-p)p_1p_2$ by setting $\rho_a(0) = 1$. Note that $E(Y_1) = pp_1$ and $E(Y_2) = pp_2$. Thus, p, p_1, p_2 are now all identifiable from the success events Y_u and Y_v. Furthermore, since $\rho_a(\Delta_u)$ changes as a function of Δ_u, the parameter a is identifiable from (3) (e.g., by fixing some Δ_u and evaluating (3)). ∎

On the other hand, when $\rho_a(\cdot) = a$ and $0 \leq a < 1$ (see (8)), $Cov(Y_u, Y_v) = p(1-p)p_1p_2a$. With two more equations $E(Y_1) = pp_1$ and $E(Y_2) = pp_2$, it is not enough to identify the four unknowns p, p_1, p_2 and a. Note that [11] uses the constant correlation function and all packet pairs (not just packet pairs destined for distinct leaves). However, it can be shown that even in such scenarios, the link loss rates are not identifiable without further assumptions.

E. Bias When Assuming Perfect Correlation

Here, we study the bias of the estimate of p (i.e., the common link success rate) when it is derived based on an *incorrect* assumption of perfect correlation. This motivates us to devise a more accurate modeling scheme to capture the effect of time correlation. We use a simple two-level, two-leaf tree topology for this study as a closed-form estimate of p can be obtained. In Section VI, we shall use model simulation to show that our theoretical results also hold empirically for a two-level tree with a general number of leaves.

Assuming there are no additional single packet measurements other than those obtained from packet pairs. Let $M_{kl}, k = 0, 1$ be the total number of packet pairs to the two leaves with success state (k, l) (1 implies a success and 0 implies failures). Suppose that $M_{01}M_{10} \leq M_{00}M_{11}$, which is generally true since if the first packet is received (or lost), then it is more likely for the second packet to be received (or lost) due to correlation. Thus, for the simple two-leaf tree, the MLE of p is given by:

$$\hat{p} = \frac{(M_{10} + M_{11})(M_{01} + M_{11})}{M_{11}N}. \tag{9}$$

Suppose that all packet pairs $\{(\mathcal{U}, \mathcal{V})\}$ have a constant inter-arrival time $\Delta_u = \Delta$. The following theorem gives the asymptotic bias of $\hat{p}$ in (9) as the number of packet pairs goes to infinity. We leave the detailed proof to Appendix.

Theorem 2: Let the packet pair measurements on the two-leaf tree have a fixed inter-arrival time $\Delta > 0$. Let $\rho = Cor(Z_u, Z_v) \leq 1$, where (Z_u, Z_v) is the unobserved successful events on the common link in the two-leaf tree. As $N \to \infty$, the estimate of p in (9) has an asymptotic bias of

$$\text{Bias}(\hat{p}) \doteq \hat{p} - p \longrightarrow \frac{p(1-p)(1-\rho)}{p + (1-p)\rho}. \tag{10}$$

Hence it is easy to conclude that the relative bias for loss rate $(1-p)$ is

$$\text{RelativeBias}(1-\hat{p}) \doteq \frac{\text{Bias}(1-\hat{p})}{1-p} \longrightarrow -\frac{p(1-\rho)}{p + (1-p)\rho}. \tag{11}$$

Thus, when p is close to 1, $\text{RelativeBias}(1-\hat{p}) \approx \rho - 1$. ∎

The above theorem states that for a two-leaf tree, the estimate under the perfect correlation model results in an underestimate of the true loss rate, with an approximate relative bias of $\rho - 1$ for small loss rates. We immediately have the following corollary.

Corollary 1: Let $f(t)$ be the density function of the time difference Δ_u between packet pairs. As p is close to 1, the asymptotic relative bias of the perfect correlation estimate is:

$$\text{RelativeBias}(1-\hat{p}) \longrightarrow -\int_t (1 - \rho_a(t))f(t)dt. \tag{12}$$

∎

IV. Profile Likelihood Based Inference

In this section, we propose our loss rate inference approach that utilizes both single packet and packet pair measurements. Our approach is based on the *profile likelihood* (PL) method [18]. We first present our methodology for a two-level tree (see Figure 2), and we later our solution for more complex networks (see Section V).

A. Single Packet and Packet Pair Measurements

As mentioned before, our criteria for a valid packet pair are neighboring packets that are (i) destined for different leaf branches, and (ii) separated by a relatively small time. In addition, we also consider all observed packets as valid single packet measurements. Clearly, a packet pair is considered both a packet pair and two single packet measurements. Both single packet and packet pair measurements will be used as inputs to our loss rate inference approach.

B. The Profile Likelihood Method

We consider a two-level tree with K leaves shown in Figure 2. Let $\mathcal{P}_1$ and $\mathcal{P}_2$ represent the sets of single packet measurements and packet pair measurements, respectively. Note that measurements in $\mathcal{P}_1$ and $\mathcal{P}_2$ may be correlated. However, we shall ignore these correlations and adopt a pseudo-likelihood approach [11], [14] by assuming that these measurements are independent. The implication of this assumption is that we may have an estimate that has a larger variance (i.e., not as statistically efficient as if we model the complete dependency). However, we observe that this approach still gives an accurate estimate in general (according to our evaluations in Sections VI and VII).

We adopt the similar notation as in Section III. Recall that $r_{kl}(\mathcal{U},\mathcal{V}) \doteq P(Y_u = k, Y_v = l)$ represents the success/failure probabilities of the packet pair $(\mathcal{U},\mathcal{V})$ (1 implies a success and 0 implies a failure). Based on our correlation function at the common link (see (2)), we can show that the log-likelihood of pair measurements in $\mathcal{P}_2$ is

$$L = \sum_{(\mathcal{U},\mathcal{V})\in\mathcal{P}_2} \sum_{k,l=0}^{1} I(Y_u = k, Y_v = l) \log r_{kl}(\mathcal{U},\mathcal{V}), \qquad (13)$$

where $\log r_{kl}(\mathcal{U},\mathcal{V})$ can be expressed in a similar manner as in (4) by replacing p_1 with p_u, and p_2 with p_v. The unknown parameters in the above are $\{p,\, p_1, p_2, \cdots, p_K,\, a\}$, where a is the parameter that specifies the decay in time correlation.

In the following, we describe the steps of estimating p, the packet success rate at the common link.

Step 1: apply end-to-end success rates into the model. For $i = 1, \ldots, K$, let P_i be the end-to-end success rate to the K leaf links. We have

$$P_i = pp_i, \qquad (14)$$

For statistical inference, we first re-parameterize the likelihood in (13) using the new parameter set $\{p, P_1, P_2, \cdots, P_K, a\}$. Therefore, we can rewrite $r_{k,l}(\mathcal{U},\mathcal{V})$ for $k, l = 0, 1$ as

$$\begin{aligned} r_{11}(\mathcal{U},\mathcal{V}) &= P_u P_v p^{-1}(p + (1-p)\rho_a(\Delta_u)), \\ r_{10}(\mathcal{U},\mathcal{V}) &= P_u - r_{11}(\mathcal{U},\mathcal{V}), \\ r_{01}(\mathcal{U},\mathcal{V}) &= P_v - r_{11}(\mathcal{U},\mathcal{V}), \\ r_{00}(\mathcal{U},\mathcal{V}) &= 1 + r_{11}(\mathcal{U},\mathcal{V}) - P_u - P_v. \end{aligned} \qquad (15)$$

Step 2: remove nuisance parameters via the profile likelihood approach. We now propose an approach based on *profile likelihood* (PL) [18] for parameter estimation. The core idea of the PL approach is to replace some of the unknown parameters by their appropriate estimates (or based on other unknown parameters), in order to reduce the number of dimensions of the optimizing problem substantially. It has been shown that the PL approach works very well in the presence of many nuisance parameters. Here, we treat the common link success rate p as the main parameter of interest, and P_i $(i = 1, \ldots, K)$ as the nuisance parameters.

Let N_i be the number of single packet measurements to leaf link i, and M_i be the number of total successes among these packets. We now replace P_i in (13) by the maximum likelihood estimate (MLE) based on end-to-end single packet measurements, i.e.,

$$\hat{P}_i = M_i/N_i. \qquad (16)$$

Thus, we now optimize the following likelihood with respect to p and a:

$$L_{prof}(p,a) = \sum_{(\mathcal{U},\mathcal{V})\in\mathcal{P}_2} \sum_{k,l=0}^{1} I(Y_u = k, Y_v = l) \log \widetilde{r}_{kl}(\mathcal{U},\mathcal{V}), \qquad (17)$$

where $\widetilde{r}_{kl}(\cdot,\cdot), k, l = 0, 1$ are obtained from (15) with P_i replaced by its estimate $\hat{P}_i$. Given the constraints that $0 \le r_{kl}(\cdot,\cdot) \le 1$, we can derive that the search space for p in optimizing $L_{prof}(p,a)$ in (17) is

$$\max_{i=1}^{K} \hat{P}_i \le p \le 1. \qquad (18)$$

To derive the confidence intervals for the unknown parameter p, we can adopt one of the following two methods. The first is to apply the PL method by treating a as the nuisance parameter [18]. The second is based on a bootstrap method [13]. We do not address the details here in the interest of space.

We remark here that in passive monitoring the volume of single packet measurements is much larger than the packet pair measurements. Hence, for a small two-level tree, we expect to have a very accurate estimate of $\mathcal{P}_1$. However, this no longer holds when the number of leaves K increases. In Section V, we shall discuss how we adapt our approach to large K.

Step 3: estimate p when $\rho_a(\cdot)$ is unknown. In reality, we do not know the exact form of $\rho_a(\cdot)$. The approach we take in this paper is to choose a small value of δ and use statistical model selection to choose $\rho_a(\cdot)$ from a set of simple functional forms. For example, when we consider the linear or quadratic forms of $\rho_a(\cdot)$ as in (6), we first find the optimum value of (p,a) for each model, and use the estimate from the model that returns the largest likelihood value.

Since the validity of the correlation function may depend on the choice of δ, it is important to choose δ appropriately in estimation. A larger δ allows us to use more packet pairs for inference, but may introduce a bias given the simple models of $\rho_a(\cdot)$. In our network simulation (Section VII), we note that if δ is in the range of 10 to 100 milliseconds, then our inference can return quite accurate and consistent results, even though we use the approximation of the time correlation function in Equation 6 (Section III).

When we estimate under the linear and quadratic forms of $\rho_a(\cdot)$, there is no close-form solution for the optimization of the PL function L_{prof} in (17). Fortunately, since it is a two-dimensional constrained optimization problem, the optimum value of (p,a) is not difficult to obtain. We conduct constrained optimization using the BFGS quasi-Newton method [8], in which the estimates obtained from the perfect correlation model (i.e., $\rho(\Delta_u) = 1$) are used as starting values.

C. Comparison with the Earlier Approach

We remark here the difference of our inference approach from that in [11], both of which use packet pair and single packet measurements for statistical inference. First, as stated earlier, [11] uses all packet pairs, while we restrict the packet pairs to those that are destined for different leaf branches.

As a result, for the simple two-level tree with K leaves, the model in [11] introduces K additional parameters to represent the conditional success probabilities of the leaf links. Also, as addressed in [22], these conditional success probabilities satisfy the natural constraints that they are larger than the unconditional success probabilities p_i. This and the increased size of parameter space will significantly complicate the loss rate estimation.

Another difference is that we explicitly model the decay of success correlation of packet pairs as a function of the time difference, while in [11], it is treated as a fixed unknown constant. As shown in Theorem 1, the model in [11] is not statistically identifiable without any additional constraints. In addition, by modeling the correlation decay, we can obtain better parameter accuracies by choosing an appropriate minimum time difference δ allowable for the analysis.

V. Extensions of Basic Methodology

In this section, we extend our basic PL-based inference approach developed for two-level, two-leaf trees for more general topologies.

A. Two-level Tree with Many Leaves

Let K be the number of leaves of a two-level tree, and N be the total number of pair measurements. When K is large while N is fixed, it is likely that the traffic loads on the leaf links are uneven, and some leaf links have an insufficient amount of traffic to accurately estimate the end-to-end success rate P_i for $i = 1, \ldots, K$ (see (16)). As these P_i values are treated as nuisance parameters in our PL-based inference, the resulting estimate of p (i.e., the common link success rate) may have poor accuracy, as will be shown via model simulation in Section VI. In the following, we propose a statistically efficient approach to address this issue.

Our goal is to remove the deviations among the nuisance parameters. We consider a simple heuristic as follows. Instead of treating the end-to-end success (loss) rates of individual leaf branches as separate parameters, we shall treat them as having the same value of P, i.e., $P_i = P$ for $i = 1, \ldots, K$. Let M be the total number of successful single packets and N be the total number of single packets. Then the MLE for P from single packet measurements is $\hat{P} = M/N$. Thus, (15) reduces to

$$r_{11}(\mathcal{U}, \mathcal{V}) = P^2 p^{-1}(p + (1-p)\rho_a(\Delta_u)),$$

and we can carry out the same optimization procedure to obtain the estimate p as before. In fact, we can readily show that if we use the same $\hat{P}$ as estimates of P_i to optimize (17) with respect to (p, a), then it is actually equivalent to optimizing the function on a two-leaf tree in which packet $\mathcal{U}$ and packet $\mathcal{V}$ always go to the first leaf and the second leaf, respectively, and both leaf links have the same success rate. We shall demonstrate in Section VI that when N is reasonably large, such a procedure for estimating p performs almost as well as knowing the true values of the end-to-end success rate P_i when K is large.

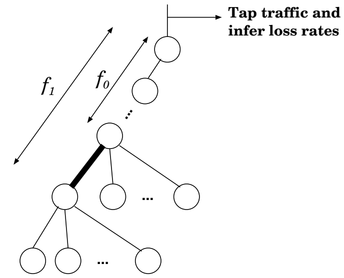


Fig. 3. Inferring the loss rate of the link of interest (bold link) in a general topology.

B. General Topologies

For simplicity and scalability, we adopt the following approach to generalize our method for general topologies. Specifically, we decompose the problem into many two-level tree estimation problems to which we can apply our developed procedure.

Figure 3 depicts a general topology where we want to infer the loss rate of the link of interest. Denote the link of interest by l and its loss rate by f. Let $Path_0(l)$, $Path_1(l)$, $Path_0(l) \subset Path_1(l)$, denote the parent path and the self path from the root to the two end nodes of link l. Let f_0, f_1 be the loss rates for paths $Path_0(l)$ and $Path_1(l)$, respectively, and let $\hat{f}_0, \hat{f}_1$ denote their respective estimates. Note that $(1 - f_1) = (1 - f)(1 - f_0)$, therefore f can be estimated by

$$\hat{f} = \max(0, (\hat{f}_1 - \hat{f}_0)/(1 - \hat{f}_0)). \tag{19}$$

By treating a path as a composite link and its children as leaf links, estimates $\hat{f}_0, \hat{f}_1$ can be derived from the PL-based inference approach developed for two-level trees.

VI. Model Simulation

We start with using model simulation to evaluate the effectiveness of our PL-based inference approach under different forms of time correlation functions that we explicitly specify, so as to motivate the use of PL-based inference when there are imperfect time correlations of packet losses. Our evaluation is built on the statistical tool R [2].

Here, we focus our simulation on two-level trees. We generate a set of packet pair samples whose loss behavior follow the time correlation functions that we specify. We assume that all packets pairs that we generate are statistically independent of each other. Making this assumption enables us to ignore other side impacts and focus only on the imperfect time correlations that are shown in each individual packet pair. In Section VII we use network simulation to evaluate more realistic scenarios where packet pairs are generally dependent, and the tree topologies that have more than two levels.

In each simulation run, we generate 5000 samples of packet pairs. For each leaf branch i, we assign a weight $0 < w_i < 1$ such that the frequency of packet pairs to the two distinct leaves (i, j) is proportional to $w_i w_j$. The leaf loss probabilities are generated uniformly at random between 1% and 15%, and

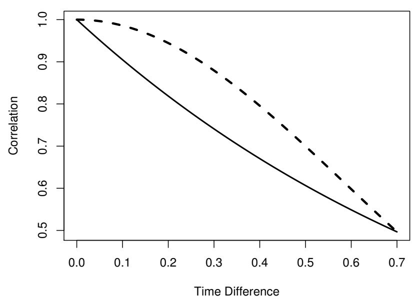


Fig. 4. Loss time correlation functions in model simulation: linear (solid) and quadratic (dotted).

we vary the common link loss probabilities from 2% to 10% in increment of 0.5%. To model time correlation, we consider both the linear and quadratic forms that are defined as follows (see Figure 4 for illustration):

$$\rho(\Delta) = \exp(-\Delta), \quad \text{and} \quad \rho(\Delta) = \exp(-1.45\Delta^2). \tag{20}$$

We then generate the time difference of each packet pair uniformly between 0 and 0.7. The range of parameters is chosen in such a way so that both functions in (20) decay from 1 to 0.5 when Δ increases from 0 to 0.7. For each set of simulation, we carry out 100 simulation runs and obtain the average results.

Experiment A.1 (PL-based Loss Estimates for Small Trees). We first evaluate the PL-based estimates on small trees. In this experiment, we aim to show that there is a significant bias in loss estimates when we wrongly assume a perfect loss correlation model. In addition, we compare the empirical bias obtained from simulation and the theoretical bias in Theorem 2 that is developed for two-leaf trees, so as to demonstrate that the derived theoretical bias formula also works well empirically for a general two-level tree with more than two leaves.

Here, we consider a two-leaf tree and a five-leaf tree. For the two-leaf tree, we set the weights $w_1 = w_2 = 1$, and for the five-leaf tree, we use $w_i = i, i = 1, \ldots, 5$ such that the packet pair distribution among leaf pairs are quite skewed.

We use the relative bias and standard deviation to assess the performance accuracy of our loss estimates on the common link. Let f be the actual common link loss rate, $\hat{f}_r$ be the inferred estimate obtained in the rth run, and $\hat{\mu}(\hat{f})$ be the empirical mean of 100 runs. Then we compute the relative bias and standard deviation as follows:

$$\mathrm{RelBias}(\hat{f}) = f^{-1}\left(\hat{\mu}(\hat{f}) - f\right), \tag{21}$$

$$\mathrm{RelSD}(\hat{f}) = f^{-1}\left(\frac{1}{100}\sum_{r=1}^{100}\left(\hat{f}_r - \hat{\mu}(\hat{f})\right)^2\right)^{-\frac{1}{2}}. \tag{22}$$

Figure 5 depicts the average relative bias of the link loss estimate versus the actual link loss rate f under two loss models in (20). We see from the figure that our PL-based inference (solid curves) does not have a visible bias. However, if we ignore the correlation decay and assuming a perfect correlation model, then we see from the figure that there is significant bias in loss estimation (dashed curves). Furthermore, the theoretical

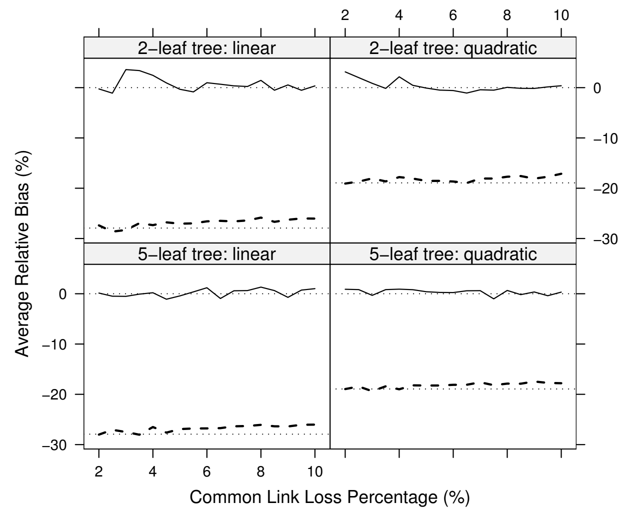


Fig. 5. Experiment A.1: Average relative bias of loss estimates for the common link using PL-based inference for two-leaf and five-leaf trees: Estimates under the correct loss correlation model (solid), estimates under the perfect correlation model (dashed) and their approximate theoretical bias (dotted).

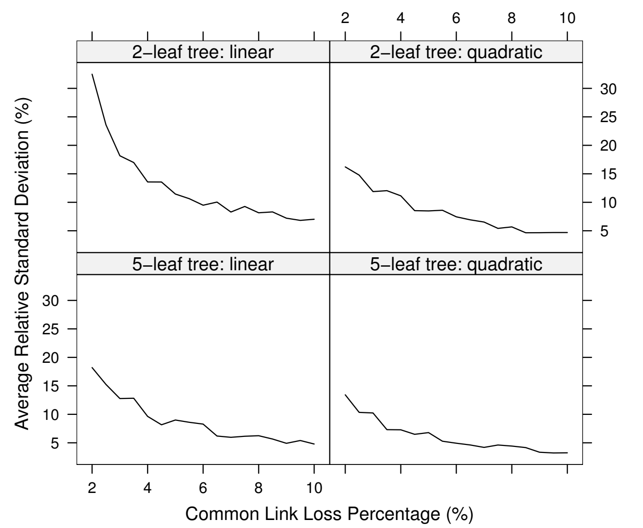
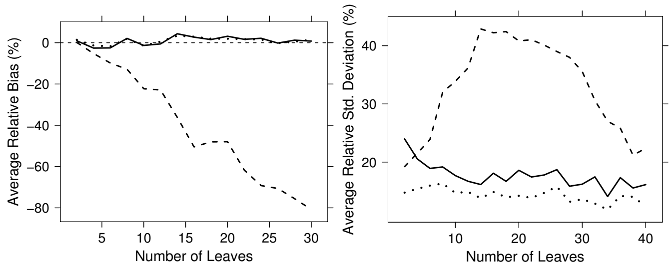


Fig. 6. Experiment A.1: Average relative standard deviation of the same loss estimates for the common link as in Figure 5, for 2-leaf and 5-leaf trees under two loss models (20) respectively.

bias (dotted curves) obtained from (12) also works well for both two-leaf and five-leaf trees, although it is only derived for the two-leaf tree case.

Figure 6 shows the average relative standard deviation versus the true loss rate f based on our PL-based inference approach. We can see that the relative standard deviation of estimates decays as f increases. This is expected, since when f increases, the end-to-end loss is more likely due to the common link loss. The other nuisance parameters in our PL-based inference has less influence, and the deviation also decreases.

Experiment A.2 (PL-based Loss Estimates for Large Trees). In this experiment, we study how our PL-based estimation performs as the number of leaves K increases. Here



(a) Avg. rel. bias (b) Avg. rel. standard deviation

Fig. 7. Experiment A.2: Average relative bias and standard deviations versus the number of leaves of the two-level tree using different methods for computing the end-to-end success probabilities - *est.equal* (solid), *est.self* (dashed), and *est.true* (dotted).

we fix the common link loss rate at 5%, and vary the number of leaves from 2 to 40, while the number of pair measurements remains to be 5000. The leaf weights $w_i, i = 1, \ldots, K$ are sampled independently from an exponential distribution. Thus, the distribution of packet pairs among leaf pairs is quite skewed. Thus, some leaf branches have far more packet pair samples than the others, especially when K is large.

We consider three kinds of loss estimates for our PL-based estimation, given the increasing number of leaves K. The first estimate, denoted by *est.equal*, is the our proposed estimate in Section V-A assuming the same end-to-end success rate. The second estimate, denoted by *est.self*, is the estimate obtained using the empirical end-to-end success rates (which may be skewed) As a reference, we also obtained estimates using the true end-to-end success rates, denoted by *est.true*.

Figures 7(a) and 7(b) respectively depict the average relative bias and average relative standard deviations of the three link loss estimates versus the number of leaves. From both figures, we observe that when the number of leaves K is small, *est.equal* (solid curves) is slightly worse than *est.self* (dashed curves), since there is a bias introduced when we assume the same end-to-end loss rates while all leaf branches receive sufficient packet samples. However, we emphasize that both are almost as good as *est.true* (dotted curves). However, as K increases (around $K = 10$), the performance of *est.equal* is almost in par with *est.true*, while the performance of *est.self* degrades significantly. This is due to the fact that when K increases, some leaf links do not receive enough samples (as we generate samples based on a skewed distribution), one can no longer estimate the end-to-end loss rates well, and this significantly degrades the estimation performance. On the other hand, using *est.equal*, we can estimate the common link loss rate quite well, and we see from the empirical results that *est.equal* is robust to the differences of the numbers of samples among the leaf links.

Experiment A.3 (Bias When the Loss Correlation Model is Unknown). In practice, we do not know the exact form of the correlation model. As mentioned in Section IV, the approach we take here is to run several loss correlation models and pick the one with the highest value of the PL function. We investigate here how this practice would affect the performance

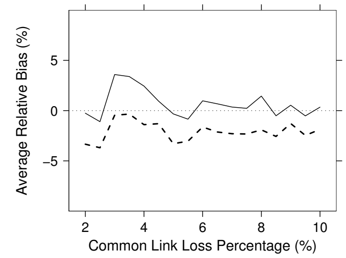


Fig. 8. Experiment A.3: Average relative bias for 5-leaf trees - *est.true* (solid) and *est.bestmodel* (dotted).

of our loss rate estimates.

In this experiment, we generate packet pairs for the same five-leaf tree in Experiment A.1 using only a linear time correlation model in (20). However, we obtain loss estimates under both linear and quadratic models, and select the estimates from the model that returns the higher likelihood value, which we refer to this as *est.bestmodel*. We compare this estimate with that obtained under the true (i.e., linear) correlation model referred as *est.true*.

Figure 8 depicts the average relative bias of *est.true* (solid) and *est.bestmodel* (dashed) versus the true loss rate f. We see that the bias of *est.bestmodel* does not deviate much from *est.true*. There is an acceptable small negative bias of *est.bestmodel* (around 5%) compared to *est.true*. This is much less than just fitting under the wrong model, and much better than 20% bias for the perfect correlation model (see Figure 5).

VII. Network Simulation

We now evaluate our PL-based inference approach using network simulation in ns2 [1]. Our goal is to show that our approach generates tolerable errors of link loss estimates under realistic traffic conditions as well as large network topologies. In particular, we seek to evaluate the cases where the time correlation function is difficult to model in practice.

We consider TCP-based loss inference in our simulation. To verify the correctness of different loss inference algorithms, we consider the scenarios where traffic loads are fairly high and hence packet losses are prominent. Specifically, we create short-lived TCP flows that follow a Poisson arrival process with mean 10ms, and each TCP session has an exponentially distributed duration with mean 1s. We use short-lived TCP flows as they are prevalent in the Internet [17]. Also, we create a number of background UDP on-off flows to add some variance to the network traffic. Unless stated otherwise, all TCP and UDP flows will send packets from the root to one of the leaf branches that is uniformly chosen. Thus, we expect that the traffic load in each leaf branch is fairly balanced, provided that the number of leaf branches is small (we consider the case of skewed loads in Experiment B.3). Since our inference is based on TCP, we only use TCP packets to compute the actual loss rate for our ground truth. Our metric of interest is the *absolute relative error* of the inferred loss rate relative to the actual loss rate.

We obtain single packet measurements and packet pair measurements from each simulation run. We then fit the measurement data into different loss estimation approaches that we develop in R (see Section VI). Here, we consider three loss rate estimation approaches:

- *est.equal*: the PL-based estimation that uses the same end-to-end success rate to remove the skewness of packet distribution in leaf branches (see Section V-A);
- *est.self*: the PL-based estimation that directly uses the end-to-end success (or loss) rates of leaf branches (which may be skewed); and
- *est.perfect*: the baseline estimation that assumes perfect loss correlation and directly uses the end-to-end success (or loss) rates of leaf branches.

We let both *est.equal* and *est.self* use the linear form of the time correlation function (see Equation 6 in Section III). Also, we have all approaches treat two neighboring packets as a packet pair if they arrive within 100 milliseconds and target different leaf branches.

A. The Two-level Tree Topology

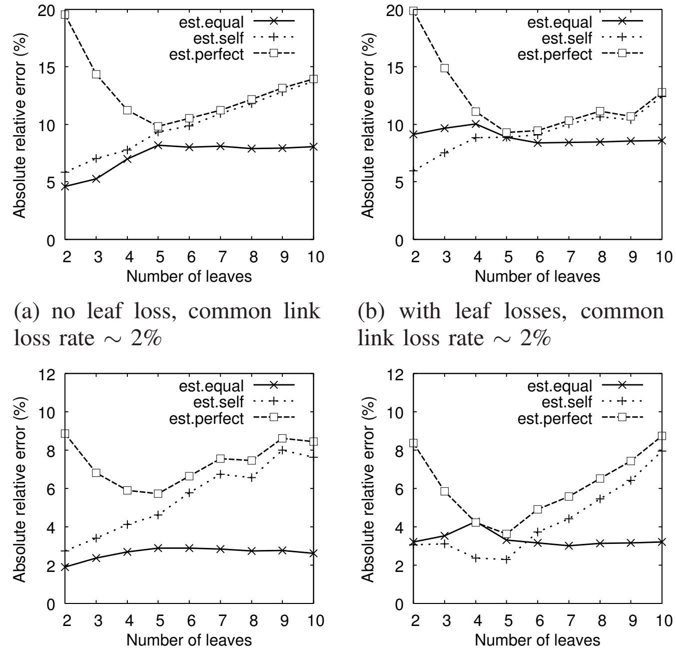
We first consider the two-level tree topology shown in Figure 2 (see Section II). The topology has a common link at the root node, and a varying number of leaf branches. We set the common link bandwidth to be 5Mb/s, while the leaf branches have the same bandwidth 1Mb/s. All links have propagation delay 5ms. Based on the given loss models, we aim to infer the loss rate of the common link at the root node.

In the following results, each measurement is averaged over 30 simulation runs with different random seeds, and each simulation run lasts for 60 seconds.

Experiment B.1 (The ON-OFF loss model). In this experiment, we consider the *ON-OFF* loss model, in which we view each link as a network path, and associate it with a continuous-time Markovian on-off model (i.e., Gilbert model) to approximate path-level correlated packet losses. Such an on-off model is also used in previous work (e.g., [5]).

We consider two cases of packet losses: either losses only occur in the common link (i.e., no leaf loss), or losses occur in both the common link at the root node and the leaf links (i.e., with leaf losses). In the latter case, we further add some variations by uniformly letting each leaf link have one of the two possibilities: either 0% of loss or approximately 2% of losses (note that in the latter case, losses are generated from the ON-OFF model and the average off period is set to 2% of time). Here, we consider the actual loss rates at the common link at approximately 2% and 5%, respectively, and we obtain these rates by adjusting the parameters of the ON-OFF model at the common link.

Figure 9 shows the inference results. Let us first consider Figure 9(a) (i.e., the no-leaf-loss case). Both the PL-based inference approaches *est.equal* and *est.self* have smaller inference error rates than *est.perfect*. The degraded performance of *est.perfect* is due to the fact that it uses perfect correlation, so it assumes that the packets in each packet pair must be either both lost or both delivered (as there is no leaf loss). However, it is possible to have only one lost packet in a packet pair



(a) no leaf loss, common link loss rate $\sim 2\%$

(b) with leaf losses, common link loss rate $\sim 2\%$

(c) no leaf loss, common link loss rate $\sim 5\%$

(d) with leaf losses, common link loss rate $\sim 5\%$

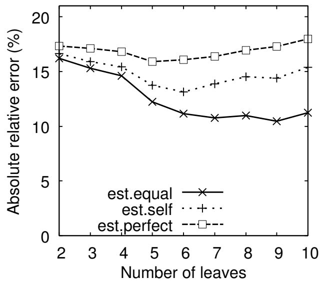
Fig. 9. Experiment B.1: Performance of different inference approaches in the ON-OFF loss model.

if there is an on-off transition in the loss model between the two packets. On the other hand, this possibility is inherently considered in PL-based inference approaches *est.equal* and *est.self*, so they have smaller error rates.

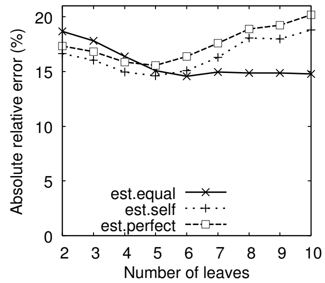
In Figure 9(b) (i.e., the with-leaf-losses case), we still see that both *est.equal* and *est.self* have smaller error rates than *est.perfect*. We note that *est.equal* now has a higher error rate than *est.self* when the number of leaf branches is small. However, *est.equal* outperforms *est.self* when the number of leaf branches is large. This observation is consistent with that in our model simulation (see the discussion in Experiment A.2 in Section VI). In both Figures 9(a) and 9(b), we observe that *est.equal* remains robust toward the number of leaf branches and its error rate is within 10%.

In Figures 9(c) and 9(d), the common link has a higher loss rate (i.e., 5%). We observe that all approaches incur smaller error rates than Figures 9(a) and 9(b), where the common link loss rate is 2%. This is expected, as right now the end-to-end losses more likely occur at the common link. In this case, we still observe that both *est.equal* and *est.self* outperform *est.perfect*, while *est.equal* maintains low error rates (within 4.2%) with regard to the number of leaf branches.

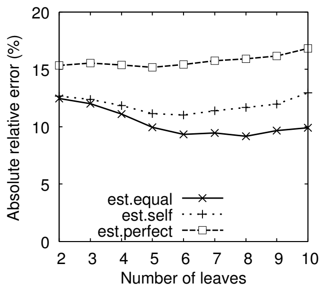
Experiment B.2 (The QUEUE loss model). We consider another way of generating packet losses, which we call the QUEUE loss model. In this model, we associate each node with a finite-buffer drop-tail queue, given that the drop-tail policy is configured in most Internet routers today. We inject large UDP bursts to cause queue overflows and hence packet losses. The UDP bursts follow an exponential on-off distribution, in which we adjust the on-off periods to obtain



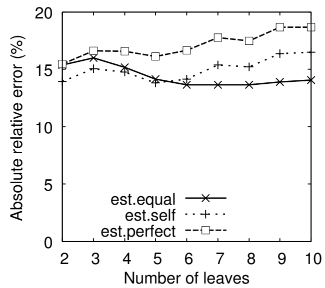
(a) no leaf loss, common link loss rate $\sim 2\%$



(b) with leaf losses, common link loss rate $\sim 2\%$

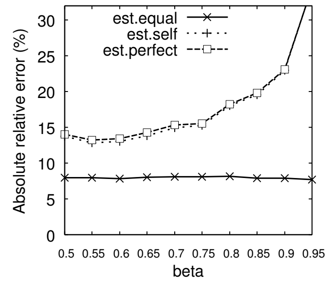


(c) no leaf loss, common link loss rate $\sim 5\%$

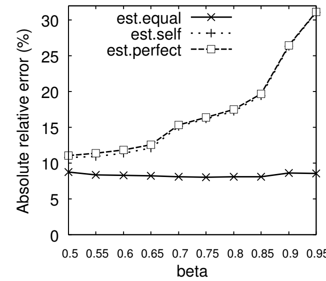


(d) with leaf losses, common link loss rate $\sim 5\%$

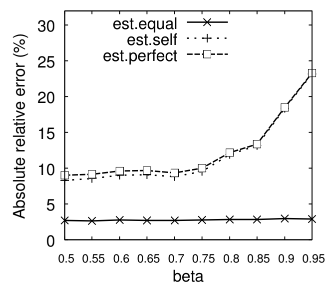
Fig. 10. Experiment B.2: Performance of different inference approaches in the QUEUE loss model.



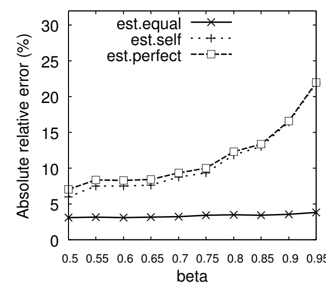
(a) no leaf loss, common link loss rate $\sim 2\%$



(b) with leaf losses, common link loss rate $\sim 2\%$



(c) no leaf loss, common link loss rate $\sim 5\%$



(d) with leaf losses, common link loss rate $\sim 5\%$

Fig. 11. Experiment B.3: Performance of different inference approaches under skewed traffic distribution, assuming that there are 10 leaf branches and the ON-OFF model is used.

the actual loss rates that we consider, i.e., $\sim$2% and $\sim$5%. We again consider the no-leaf-loss and with-leaf-losses cases as in Experiment B.1.

Figure 10 shows the inference results for the QUEUE loss model. In general, the error rates of all inference approaches in the QUEUE model are in the range 10-20%, and are higher than those in the ON-OFF model. Nevertheless, similar to Experiment B.1, in the with-leaf-losses cases (i.e., Figures 10(b) and 10(d))), *est.equal* introduces higher error rates when the number of leaf branches is small, but outperforms the other two approaches when the number of leaf branches increases. In short, we observe similar improvements in *est.equal* and *est.self* over *est.perfect* as in Experiment B.1.

Experiment B.3 (Skewed traffic loads). In this experiment, we consider the scenarios when the traffic loads in the leaf branches are uneven. We first divide the leaf branches into two groups (denoted by Group 1 and Group 2). For each TCP flow we generate, it first selects either Group 1 or Group 2 with probabilities β and $1-\beta$, respectively, where $0.5 \leq \beta < 1$. It then sends traffic to one of the uniformly chosen leaf branches within the selected group. Thus, if $\beta > 0.5$, then the leaf branches in Group 1 will see more traffic than in Group 2.

Here, we fix the number of leaf branches to be 10, and focus on the ON-OFF loss model. Figure 11 shows the performance of different inference approaches versus β. We observe that as β increases, the performance of both *est.self* and *est.perfect* degrade significantly, even though *est.self* is slightly better. On the other hand, *est.equal* remains robust toward the skewed traffic loads by assuming the same end-to-end loss rates in all leaf branches. Similar observations are made for other numbers of leaf branches and for the QUEUE loss model.

B. The Large Tree Topology

We now evaluate the inference algorithms based on the extensions for general topologies (Section V). Here, we consider a large four-level tree topology as shown in Figure 12, which mimics the four-level hierarchy as in a wireless data architecture (see Figure 1 in Section I). Here, we assume that the link bandwidth at the root node is 5Mb/s, and other links have the same bandwidth 1Mb/s. All links have propagation delay 5ms.

In Figure 12, the topology has four levels labeled as levels 1 to 4. The degrees of levels 1-4 are 1, 4, 4, and 10, respectively. We focus on the ON-OFF loss model. Each TCP flow sends traffic from the root to one of the leaf branches in level 4. In each non-leaf level (i.e., levels 1 to 3), we select the leftmost link (see the bold links in the figure) as our link of interest. Our goal is to infer their loss rates using different inference approaches. We obtain estimates over 30 simulation runs. For each run, we increase the simulation time to 300s to obtain more samples.

Experiment B.4 (Large topology). We first focus on the case where packet losses only occur in the links of interest (i.e., the bold links in Figure 12), while other links do not have any loss. For each link of interest, we generate an actual loss rate of approximately 2-3%.

Figure 13 shows the boxplots of different inference approaches. In level 1, the error rates are consistent with the two-level tree settings in Experiment B.1 (with an average of around 5%). In levels 2 and 3, we see fewer packet

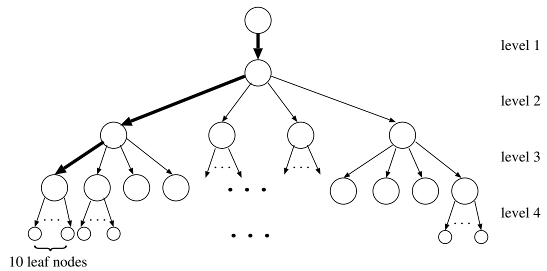


Fig. 12. The four-level tree topology in network simulation (Section VII-B). Our goal is to infer the loss rates of the bold links.

samples, so the error rates are higher than in level 1. In general, both *est.equal* and *est.self* outperform *est.perfect*. Such improvements are more obvious in level 2 and level 3. In particular, we note that in level 3, the error of *est.perfect* almost reaches 100%, and we find that it reports 0% loss estimates in most cases.

We note that *est.equal* has a slightly higher error rate than *est.self* in levels 1 and 2. However, it is interesting to note that *est.self* has a very high error rate in level 3. One reason is that we here only generate packet losses at the links of interest, so different root-to-leaf paths will have different end-to-end loss rates. Thus, the TCP flows along more lossy paths will experience more timeouts in response to packet losses, and different paths will have uneven traffic loads. This makes *est.self* less robust, as shown in the two-level tree settings. On the other hand, the error rates of *est.equal* are the smallest (within 20% in most cases) among all approaches in level 3.

Figure 14 shows the results of another scenario in which we generate losses at *all* links in the topology at approximately 2%. We note that both *est.equal* and *est.self* have similar error rates, given that every link has a similar loss rate and all paths have fairly balanced traffic loads. Both of them significantly outperform *est.perfect* in levels 2 and 3.

Summary. We compare different inference approaches using network simulation. Similar to our model simulations (Section VI), our PL-based inference outperforms the inference that assumes perfect correlation. Also, by using the same end-to-end loss rates, our PL-based inference can be robust to uneven traffic loads in different end-to-end paths. We also show that our PL-based inference remains feasible in large-scale topologies.

VIII. Related Work

Loss rate inference is first addressed under multicast settings [4], [9], [23], in which a source dispatches multicast probes, and then multiple receivers measure the end-to-end loss rates of such probes. Statistical techniques, such as maximum likelihood estimation [9], [23] and temporal estimation [4], are then applied to infer internal losses. The above methods focus on a single tree, and [7] considers a multi-tree setting.

Given that IP multicast is not widely deployed, unicast-based loss rate inference is proposed in [11], [12], [16], whose idea is to send unicast probing packets to different receivers to infer losses. The main assumption is that unicast probes are loss correlated temporally on the common link that they traverse. A more robust unicast-based inference approach is later considered in [20], which exploits the variance of loss rates of unicast probes.

The above methods are based on active probing, as they generate dedicated probing packets purely for loss measurements. However, active probing introduces additional traffic overhead. Passive measurement techniques are thus proposed [6], [21], [22] to infer losses through existing network traffic. While [6], [21] focus on identifying lossy links, [22] is closely related to us as they also use time correlation of existing TCP traffic for loss rate inference. However, a key distinction is that our work explicitly models time correlation in loss estimates, while this is not considered in [22].

There are recent directions of research on loss inference, such as practical implementation of inference tools (e.g., [15]). Various studies also consider delay and topology inferences, and the survey of these studies are found in [10].

IX. Conclusions

In network tomography, passive measurements introduce no probing overhead and hence are more attractive than active measurements. This paper presents a new approach of inferring link loss rates that is adaptive to different forms of time correlation when passive measurements are assumed. We develop a loss model as a function of time correlation and address its identifiability. We then apply the concept of profile likelihood in a novel way for passive network tomography to enable us to accurately infer link loss rates in complex topologies. Using extensive model and network simulations, we show that our approach provides high inference accuracy under different network settings.

Appendix

A. *Proof of Theorem 2*

From (9), we have

$$\hat{p} - p = \frac{(M_{10} + M_{11})(M_{01} + M_{11}) - pM_{11}N}{M_{11}N},$$

where the numerator can be reduced to

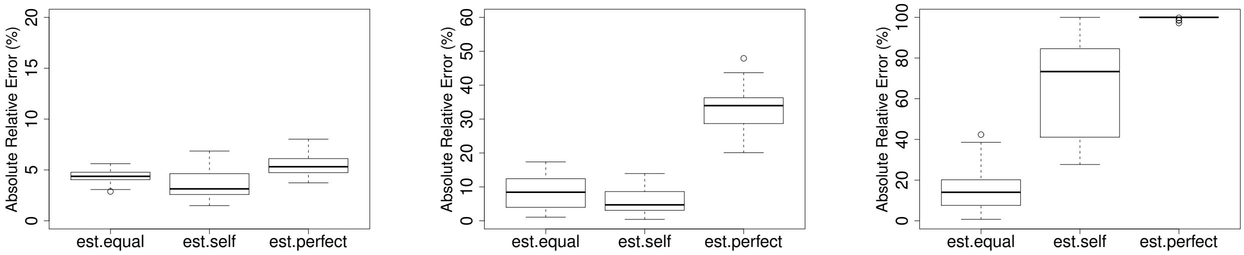
$$M_{10}M_{01} + (N - M_{00})M_{11} - pM_{11}N. \tag{23}$$

Our proof is based on the law of large numbers. Recall that the time difference between packet pairs here are assumed constant. In the following, we use the same notation as in (4) and (5), and ignore the time dependence since it is constant. It is easy to show that

$$E(M_{10}M_{01}) = N(N-1)r_{10}r_{01}, \tag{24}$$

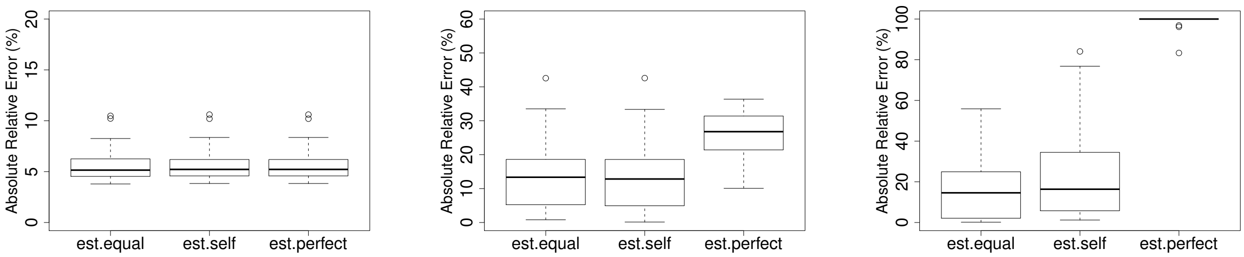
since the same packet pair cannot have different success states (i.e., the events $(Y_u = 1, Y_v = 0)$ and $(Y_u = 0, Y_v = 1)$ are contradictory) and hence only different pairs have a contribution to the above. Similarly we can show

$$E(M_{00}M_{11}) = N(N-1)r_{00}r_{11}, \tag{25}$$



(a) Level 1 (b) Level 2 (c) Level 3

Fig. 13. Experiment B.4: Performance of different inference approaches in a large tree topology, where losses occur only in the links of interest.



(a) Level 1 (b) Level 2 (c) Level 3

Fig. 14. Experiment B.4: Performance of different inference approaches in a large tree topology, where losses occur in *all* links.

and it is obvious that $E(M_{11}N) = N^2 r_{11}$. Substitute the right side of the above three equations, and by ignoring the difference between N^2 and $N(N-1)$ asymptotically, we can express (23) (for large N) as:

$$N^2(r_{10}r_{01} - r_{00}r_{11} + (1-p)r_{11}).$$

Therefore the asymptotic bias is

$$\hat{p} - p = \frac{(r_{10}r_{01} - r_{00}r_{11} + (1-p)r_{11})}{r_{11}} = \frac{p^2 p_1 p_2 - p r_{11}}{r_{11}},$$

where the second relation in the above can be easily derived by (5). Hence the result of (10).

ACKNOWLEDGMENTS

The work of Patrick P. C. Lee was supported in part by the CUHK CSE startup fund and the CUHK faculty direct grant number 2050447.

REFERENCES

[1] The network simulator - ns-2. http://www.isi.edu/nsnam/ns/.
[2] The R Project for Statistical Computing. http://www.r-project.org/.
[3] M. Allman, W. M. Eddy, and S. Ostermann. Estimating loss rates with tcp. *ACM SIGMETRICS Performance Evaluation Review*, 31(3):12–24, Dec 2003.
[4] V. Arya, N. Duffield, and D. Veitch. Multicast inference of temporal loss characteristics. *Performance Evaluation*, 64:1169–1180, 2007.
[5] J.-C. Bolot, S. Fosse-Parisis, and D. Towsley. Adaptive fec-based error control for internet telephony. In *Proc. of IEEE INFOCOM*, 1999.
[6] E. Brosh, G. Lubetzky-Sharon, and Y. Shavitt. Spatial-temporal analysis of passive tcp measurements. In *Proc. of IEEE INFOCOM*, 2005.
[7] T. Bu, N. Duffield, F. Lo Presti, and D. Towsley. Network tomography on general topologies. In *Proc. ACM SIGMETRICS*, 2002.
[8] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Scientific Computing*, 16(16):1190–1208, 1995.
[9] R. Cáceres, N. Duffield, J. Horowitz, and D. Towsley. Multicast-based inference of network-internal loss characteristics. *IEEE Trans. on Information Theory*, 45:2462–2480, 1999.
[10] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu. Network tomography: Recent developments. *Statistical Science*, 19(3):499–517, 2004.
[11] M. Coates and R. Nowak. Network loss inference using unicast end-to-end measurements. In *Proc. ITC Seminar on IP Traffic, Measurement, and Modelling*, 2000.
[12] N. Duffield, F. Lo Presti, V. Paxson, and Tow. Network loss tomography using striped unicast probes. *IEEE/ACM Trans. on Networking*, 14:697–710, 2006.
[13] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1994.
[14] L. Gang and B. Yu. Pseudo likelihood estimation in network tomography. In *Proc. IEEE INFOCOM*, 2003.
[15] D. Ghita, H. Nguyen, M. Kurant, K. Argyraki, and P. Thiran. Netscope: Practical network loss tomography. In *Proc. IEEE INFOCOM*, 2010.
[16] K. Harfoush, A. Bestavros, and J. Byers. Robust identification of shared losses using end-to-end unicast probes. In *Proc. IEEE ICNP*, 2000.
[17] M. Mellia, I. Stoica, and H. Zhang. Tcp model for short lived flows. *IEEE Communication Letters*, 6:85–87, 2002.
[18] S. A. Murphy and A. W. V. D. Vaart. On profile likelihood. *Journal of the American Statistical Association*, 95(450):449–465, 2000.
[19] H. X. Nguyen and M. Roughan. On the correlation of internet packet losses. In *ATNAC*, 2008.
[20] H. X. Nguyen and P. Thiran. Network loss inference with second order statistics of end-to-end flows. In *Proc. of IMC*, 2007.
[21] V. N. Padmanabhan, L. Qiu, and H. J. Wang. Server-based inference of internet link lossiness. In *Proc. of IEEE INFOCOM*, 2003.
[22] Y. Tsang, M. Coates, and R. Nowak. Passive network tomography using EM algorithms. In *Proc. IEEE ICASSP*, 2001.
[23] A. Ziotopolous, A. Hero, and K. Wasserman. Estimation of network link loss rates via chaining in multicast trees. In *Proc. IEEE ICASSP*, 2001.