Maximum Likelihood Network Topology Identification from Edge-based Unicast Measurements

Mark Coates
Dept. Elect. Comp. Eng.
McGill University
3480 University St
Montreal, QC, H3A 2A7
coates@ece.mcgill.ca

Rui Castro
Dept. Elect. Comp. Eng.
Rice University
6100 Main St
Houston, TX 77005-1892
rcastro@rice.edu

Robert Nowak
Dept. Elect. Comp. Eng.
Rice University
6100 Main St
Houston, TX 77005-1892
nowak@rice.edu

ABSTRACT

Network tomography is a process for inferring "internal" link-level delay and loss performance information based on end-to-end (edge) network measurements. These methods require knowledge of the network topology; therefore a first crucial step in the tomography process is topology identification. This paper considers the problem of discovering network topology solely from host-based, unicast measurements, without internal network cooperation. First, we introduce a novel delay-based measurement scheme that does not require clock synchronization, making it more practical than other previous proposals. In contrast to methods that rely on network cooperation , our methodology has the potential to identify layer two elements (provided they are logical topology branching points and induce some measurable delay). Second, we propose a maximum penalized likelihood criterion for topology identification. This is a global optimality criterion, in contrast to other recent proposals for topology identification that employ suboptimal, pair-merging strategies. We develop a novel Markov Chain Monte Carlo (MCMC) procedure for rapid determination of the most likely topologies. The performance of our new probing scheme and identification algorithm is explored through simulation and Internet experiments.

1. INTRODUCTION

The explosive growth of the Internet, combined with rapid and unpredictable developments in applications and workloads, has rendered network modeling, control, and performance prediction increasingly demanding tasks. Optimizing the performance of high-end applications requires that end-systems have knowledge of the internal network traffic conditions and services. Spatially localized information about network performance plays an important role in isolation of network congestion and detection of performance degradation. One approach to gathering local performance information is to augment the Internet infrastructure with special-purpose hardware and software, but this is impractical for many reasons [3, 21]. The alternative is to indirectly infer dynamic network characteristics from edge-based network measurements. The edge-based approach is more challenging from a measurement and inference perspective, but is much more practical and scalable.

Several groups have begun investigating methods for inferring internal network behavior based on "external" end-to-end network measurements [3, 5, 6, 7, 11, 14, 21]. This problem is often referred to as *network tomography*. All of these methods require knowledge of the network topology. Thus, a first key step in any approach to monitoring network conditions from the edge is the determination of network topology. This paper considers the problem of discovering network topology solely from host-based, unicast measurements, without internal network cooperation.

Most existing tools for network topology mapping, such as `traceroute`, rely on the cooperation of routers and thus can only reveal those portions of the network that are functioning properly and wish to be known. These cooperative conditions are often not met in practice, and may be increasingly uncommon as the network grows and privacy and proprietary concerns increase.

In this paper, we concentrate on the case of estimating the *logical* topology which arises from consideration of a single source communicating with multiple receivers. The logical topology is generated from the physical topology by using a single logical link to represent the set of physical links that connects two branching nodes. We assume the routes from the sender to the receiver are fixed during the measurement period, implying that the unknown topology connecting the sender to the receivers is a tree-structured graph.

Contribution: The contribution of this paper is two-fold. First, we introduce a novel measurement scheme based on special-purpose unicast probes that we entitle "sandwich" probes. The sandwich probing scheme is delay-based, but it measures only *delay differences*, so that no clock synchronization is required. Each measurement is generated by the difference in arrival times of two probe packets at a single receiver, as measured by that receiver. Each sandwich probe consists of three packets, two small packets destined for one receiver separated by a larger packet destined for another receiver. The idea behind the sandwich probe is that the second small packet queues behind the large, inducing extra separation between the small packets on the shared links. The difference between the arrival times of the first and second small packet at their receiver is related to the bandwidths on the portion of the path shared with the other receiver.

The goal of the probing scheme is to generate a metric that can be used for topology identification. There are two key advantages of the sandwich probing scheme as opposed to previously proposed alternatives based on loss and delay measurements. Firstly, it is our experience that probe loss on much of the Internet is very rare. Therefore, an extremely large number of probes must be sent before a loss-based metric becomes reliable for use within logical topology identification. Secondly, generating most delay-based metrics requires accurate clock synchronization between the sender and receivers. In most cases, GPS access is assumed, which severely restricts the application of delay-based schemes. Since the sandwich probing scheme is based on delay differences, it avoids the need for clock synchronization.

The second contribution of the paper is a new, penalized likelihood-based framework for topology identification. The basic idea is that, in principle, one could evaluate the statistical likelihood of every possible topology configuration given the measurements and then select the topology that maximizes the likelihood (i.e., the Maximum Likelihood Estimate (MLE)). However, the number of possible topologies grows exponentially as the number of receivers increases, and exhaustively searching over them all is infeasible in large-scale networks [10]. A crucial issue is therefore how to efficiently search the space of topologies. We propose a special Markov Chain Monte Carlo (MCMC) procedure for this task. The MCMC procedure quickly searches through the "topology space," concentrating on regions with the highest likelihood. The most advantageous attribute of the MCMC procedure is that it attempts to identify the topology *globally*, rather than incrementally (and suboptimally) a small piece at a time. The procedure is scalable, as its complexity grows linearly with the number of receivers in the network under study. The performances of the sandwich probing scheme and the MCMC topology identification algorithm are explored through `ns` simulations [19] and Internet experiments.

Related Work: Several methods have been proposed for identifying multicast topologies. The first of these, [22], was based on the observation that multicast receivers sharing a longer portion of the path have higher shared loss rates. By measuring the shared loss-rates, and recursively grouping nodes, an estimate of the logical tree can be formed. The approach was extended in [9, 10]. These papers made the approach applicable to general topologies and demonstrated that metrics based on other measurements can be used, provided the metrics satisfy certain properties as discussed in Section 2. Specific metrics that have been proposed include loss rates, mean delay, delay covariance, and link utilization (frequency of zero delay). Experiments in [9] suggested that topology identification was most reliable when either the link utilization metric or the loss metric were used. However, when network load is low (low link loss and utilization), topology identification based on the loss metric performs poorly; when network load is high (high link loss and utilization), topology identification based on the utilization metric performs poorly. To address these issues, an adaptive scheme was introduced in [8] that incorporated both utilization and loss information. The loss-based multicast methodology of [9] was adapted to the case of unicast measurements in [2].

There are two major differences between these methods and our new approach. Firstly, we use special unicast probe measurements that are neither based on losses nor require clock synchronization. Secondly, the method we use for unicast topology identification is based on maximum (penalized) likelihood, in contrast to the receiver-grouping rules that comprise the algorithms in the aforementioned papers. Although the multicast loss-based topology identification problem was posed as a maximum likelihood problem in [10], the authors proposed exhaustive search algorithms that are only feasible for very small networks, and chose to focus on deterministic grouping algorithms that do not address the maximum likelihood problem. It is proven in [2, 10] that the proposed grouping algorithms are guaranteed to identify the correct tree if the measurements provide perfect end-to-end metrics (and that the identification algorithms are consistent). However, when few measurements are available, the variation of an estimated end-to-end metric around the true metric (the noise) can be very high. Moreover, the extents of variation along different paths can differ substantially. The algorithms of [2, 10] make deterministic, local decisions based on *estimated* end-to-end metrics; they do not take into account the effects of variation or noise. Our scheme directly models the noise, and avoids making local decisions. It identifies the best (maximum likelihood) global topology. This global optimization results in substantial improvement in performance over the algorithm of [2] when either few probe measurements are made or there is substantial noise in the measurements. In addition, we incorporate complexity penalization in our framework, which enables identification of non-binary trees without the need for a supplemental pruning procedure.

The sandwich probing scheme we propose was inspired by the cartouche probing method of [15]. Cartouche probing was designed for the purpose of identifying the bottleneck bandwidth of path segments. Sandwich probing has a very different aim, so the probe structure is altered in a critical manner, as outlined in the following section. Both sandwich probing and cartouche probing have strong connections with earlier probing schemes used for bandwidth determination [17, 20, 22] and shared path detection [26].

2. SANDWICH PROBE MEASUREMENTS

Schemes for topology identification that utilize solely end-to-end measurement involve three main steps. Firstly, end-to-end measurements are made (in previous work [2, 8] these have been end-to-end loss and end-to-end delay). Secondly, a set of "end-to-end" metrics are *estimated* based on the measurements. A metric estimate is generated for each pair of receivers. This is the estimate of the true metric of the shared portion of the paths from the source to the two receivers. For successful topology identification it must be possible to decompose the end-to-end or path-level metrics into sets of "link" metrics. Examples of previously used metrics include counts of joint zero delay events (the utilization metric), counts of joint loss events, and delay covariance. In the third step, an inference algorithm uses these pairwise metrics to estimate the topology.

Our algorithm and the algorithms in [2, 9] hinge on the true end-to-end metrics being *monotone* and *separable* (using the terminology of [2]). Monotonicity requires that the true metric of a shared path must not decrease (increase) if another link is added to that shared path. Separability simply means that a path metric can be decomposed into the metrics associated with the links comprising the path. For the metric we propose, the path metric is simply the sum of the metrics of its constituent links. We discuss the monotonicity of the metric below.

The reliability of logical topology identification grows as the number of measurements increases. If an infinite number of measurements could be made, the path metrics would be known exactly, and the correct tree would always be identified [2, 10]. When a finite amount of data is available, mistakes are made. The choice of metric has a major effect on how fast the percentage of successful identification improves as the number of measurements increases. The improvement is slow if it takes many measurements to provide a sufficiently accurate indication of the true end-to-end metrics to enable successful differentiation of receiver pairs. This is the reason why loss-metrics perform poorly in lightly-loaded networks. Thousands of measurements have to be made before the one-percent loss rate of a link substantially affects the estimated end-to-end metrics. A link with a ten-percent loss rate makes its presence felt much sooner. In effect, the measurements in which a loss occurs are highly informative, whereas those where no loss occurs are relatively uninformative. Delay-based measurements tend to be more informative, but there one encounters clock synchronization issues that are difficult to overcome.

In developing a metric and associated probing scheme, we strive to make every measurement informative regardless of network conditions. This has the effect of increasing the rate at which the estimated end-to-end metrics converge to the true end-to-end metrics. A metric will result in more successful identification if the distance between path metrics is large (relative to measurement error), because it makes it easier to differentiate between them. For monotonically increasing metrics, it is therefore desirable that all link metrics be non-zero and large (in a relative sense). In the case of the utilization and loss metrics, each link metric is only non-zero because competing traffic causes delay or loss on that link. We aim to develop a metric whose constituent link metrics are intrinsically non-zero because of the measurement technique; cross-traffic effects can then be treated as noise. We perform measurement using "sandwich" probes, as described below. Almost every probe provides an informative measurement of delay difference (the metric); the only exceptions are when one or more of the probe packets are dropped.

The metrics we use are mean delay differences. Because we only ever need to measure delay differences locally (at a single receiver), there is no need for clock synchronization between the sender and receivers. The delay differentials are generated in the following manner. We perform measurement using a "sandwich" probe, which is comprised of two small packets separated by a larger packet.[1] See Figure 1 for a depiction of the probe.

Let us first consider the situation where there is no cross-traffic. In this case, the first small packet p_1 in a probe experiences no queueing delay. Suppose the shared path consists of n physical links, labelled $L_1, \ldots, L_m$ with bandwidths $b_1, \ldots, b_m$. As shown in [2], the second small packet queues behind the large packet preceding it at every queue on the shared path, provided:

$$\frac{s(q)}{s(p)} \geq \frac{b_{i+1}}{b_i} \qquad i = 1, \ldots, m \tag{1}$$

where $s(q)$ is the size of the large packet and $s(p)$ is the size of the small packet. In practice, we set $s(q)$ to be the maximum IP packet size not subject to IP fragmentation. For example, in our Internet experiments we use $s(q) = 1500$ bytes and set $s(p) = 56$ bytes. With these settings, physical links will impart extra spacing between the two small packets until a link is encountered whose bandwidth is more than 25 times that of the link preceding it. If cross-traffic is present, then the condition (1) is sufficient but not always necessary, since the large packet may be held up by the additional queuing delay.

When the large packets and small packets diverge, the spacing between the small packets is $d + \Delta d$ (see Figure 1). Suppose the path from the branching node to the destination of the small packets consists of a set of physical links $L_{m+1}, \ldots, L_n$ with bandwidths $b_{m+1}, \ldots, b_n$. As shown in [15], in the absence of cross-traffic the spacing between the small packets is preserved provided that

$$\frac{s(p)}{d + \Delta d} \leq \min_{m+1 \leq i \leq n} b_i. \tag{2}$$

In practice, this relationship is not really the governing factor in making a decision about the initial spacing d; it is determined more by the anticipated effects of cross-traffic, as discussed below.

The cross-traffic induces substantial variation in the measurements, with queuing effects before and after the branching node disrupting the measured spacing from its theoretical value. Our basic assumption is that cross-traffic has a zero-mean effect on the measurements. Our experiments suggest that this assumption is reasonable provided d is sufficiently large to eliminate (or greatly reduce) the number of occurrences of the more systematic measurement errors that occur when p_2 reaches a queue on the unshared path before p_1 has exited the queue. In these cases, the eventual spacing is directly related to the bandwidth of the (last) shared queue and the number of intervening packets. If d

[1]This probe is similar to the cartouche probe of [2]. The most important difference is that in a cartouche probe there is another large packet in front of the first small packet. When the constraints in (1) are met, the extra spacing induced in a cartouche probe is due solely to back-to-back queueing in the bottleneck queue; for the sandwich probe, extra spacing is added at every queue in the shared path.

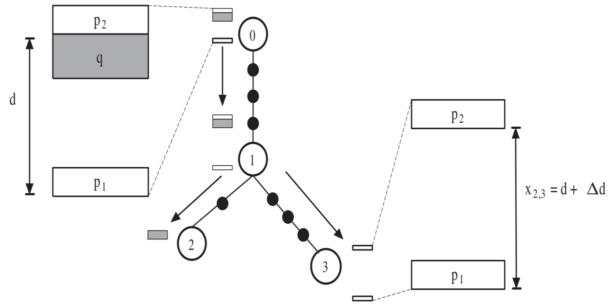

Figure 1: An example of sandwich probe measurement. The large packet is destined for node 2, the small packets for node 3. The black circles on the links represent physical queues where no branching occurs. In the absence of cross-traffic, the initial spacing between the small probes d is increased along the shared path from nodes 0 to 1 because the second small probe p_2 queues behind the large packet. The measurement $x_{2,3}$ for this receiver pair is equal to $d + \Delta d$. A larger initial spacing d reduces the chance of p_2 catching p_1 because of a bottleneck or cross-traffic on the path from node 1 to 3.

is too small, and there is relatively heavy cross-traffic on a bottleneck link, then these errors can occur relatively frequently and induce a substantial negative bias in the estimated mean. If d is set reasonably large (there is a trade-off between d and the number of probe measurements that can be made), then the number of such erroneous measurements is greatly reduced. Moreover, it is easy to differentiate between the measurements that have been affected in this way and the other measurements because the spacing between the affected measurements is much less than that of the others. These affected measurements are "outliers" that can be discarded.

We now formalize our end-to-end metric construction. Let us consider a single sender transmitting sandwich probes to a set $\mathcal{R}$ of N receivers. For every pair of receivers $i, j \in \mathcal{R}$ there are two types of measurement: one in which the two small packets in the sandwich probe are sent to receiver i and the large packet is sent to j and one in which the destinations are reversed. In total there are $N(N-1)$ different types of measurements (since we do not consider sending all packets to a single receiver). Suppose that K measurements are collected in total. For each measurement $k = 1, \ldots, K$ let $(r_1(k), r_2(k))$ denote the pair of receivers in the k-th measurement, with $r_1(k)$ referring to the receiver of the two small packets and $r_2(k)$ referring to the receiver of the larger packet(s), and let $\delta(k)$ denote the difference between the arrival times of the two small packets at receiver $r_1(k)$. Consider the subset of measurements $\{\delta(k) : r_1(k) = i, r_2(k) = j\}$. We assume that these measurements are independent and identically distributed; this assumption is reasonable if the probes are sufficiently separated in time. Let $x_{i,j}$ denote the sample mean of these measurements and $\sigma^2_{i,j}$ denote the sample variance. The sample variance provides a measure of confidence in the sample mean $x_{i,j}$, which will be used in the definition of our topology identification criterion in the next section. Computing these quantities for all pairs of measurements produces the set of metrics $\boldsymbol{x} = \{x_{i,j}\}_{i,j\in\mathcal{R}}$ and a corresponding set of variances $\{\sigma^2_{i,j}\}_{i,j\in\mathcal{R}}$.

3. MAXIMUM LIKELIHOOD TOPOLOGY IDENTIFICATION

In this section we describe and motivate the criterion on which we base our topology identification procedure. The fundamental reason for adopting a maximum (penalized) likelihood criterion is that it accounts for variation in the measurements, and results in an efficient estimator that performs well even when a small number of measurements are available. Efficiency is critical in unicast topology identification, because each probe measurement is much less informative than an equivalent multicast measurement.

We assume that the routes from the sender to the receiver are fixed during the measurement period. This implies that the (unknown) topology connecting the sender to the receivers is a tree-structured graph. The vertices of the graph represent the sender, receivers, and internal routers. Not all routers are apparent in the graph; only those routers associated with points where two or more paths to the receivers diverge are explicit. Thus, the edges (connections between vertices) in the graph correspond to sequences of one or more physical links in the network. Throughout the paper we will refer to the edges of the graph as *links*, with the understanding that each of these "logical links" corresponds to one or more physical links, and vertices will be called *nodes*. The *true* tree associated with the network under consideration is denoted by $\mathcal{T}^*$. The set of all possible trees connecting the sender to the receivers is denoted by $\mathcal{F}$. The set $\mathcal{F}$ is called a *forest*. The goal of topology identification is to find the tree $\mathcal{T}^*$ in the forest $\mathcal{F}$ based on the data $\boldsymbol{x}$ and no other information[2]. Of course, the data are imperfect and finding $\mathcal{T}^*$ cannot be guaranteed by any method. Thus, we need to define a criterion for assessing the fitness of trees.

The basis for our approach is a maximum likelihood criterion. To derive the criterion, we must first adopt a probability model for the mean delay differential metrics. Suppose that data arise from a tree $\mathcal{T}$. Associated with each link e_ℓ in the tree is a theoretical delay difference value μ_ℓ. Let $\mathcal{S}_{i,j}$ denote the set of shared links in the paths to receivers

[2]It is also possible to include side information in the problem, such as partial knowledge of the topology, but this is not considered here.

i and j. The theoretical (expected) value of $x_{i,j}$ is given by $\gamma_\ell \equiv \sum_{\ell \in \mathcal{S}_{i,j}} \mu_\ell$. The more shared links, the greater the total distance. We probabilistically model the measured difference as

$$x_{i,j} \ \sim \ \mathcal{N}(\gamma_{i,j}, \sigma^2_{i,j}), \tag{3}$$

where $\sigma^2_{i,j}$ is measured variability of the $x_{i,j}$ and $\mathcal{N}(\gamma, \sigma^2)$ denotes the Gaussian density with mean γ and variance σ^2. This model reflects the fact that the theoretical value for $x_{i,j}$ is $\gamma_{i,j}$, but there is some randomness in our measurement. We point out that we are not assuming that the delays themselves are Gaussian distributed. The motivation for the model above is that the average of several independent measurements of delay differences (recall $x_{i,j}$ is a sample mean) tends to a Gaussian distribution according to the Central Limit Theorem. Let $\boldsymbol{\mu}(\mathcal{T}) = \{\mu_\ell\}$, where ℓ runs over all internal links (not including final links to receivers). We can write the probability density for the measured data $\boldsymbol{x}$ as $p(\boldsymbol{x}|\mathcal{T}, \boldsymbol{\mu}(\mathcal{T}))$.

This probability model induces a likelihood function on the forest of trees $\mathcal{F}$ as follows. If we consider a particular tree $\mathcal{T}$, then the means $\boldsymbol{\mu}(\mathcal{T})$ of the probability model $p(\boldsymbol{x}|\mathcal{T}, \boldsymbol{\mu}(\mathcal{T}))$ can be estimated from the measurements $\boldsymbol{x}$. The natural choice for the estimator is the Maximum Likelihood Estimator (MLE). We compute the values of $\{\mu_\ell\}$ that maximize $p(\boldsymbol{x}|\mathcal{T}, \boldsymbol{\mu}(\mathcal{T}))$, subject to the constraint that each mean is non-negative. That is, the parameter space is restricted to the set $\{\mu_\ell \geq 0\}$ to enforce the known positivity conditions. Let $\widehat{\boldsymbol{\mu}}(\mathcal{T}) \equiv \{\widehat{\mu}_\ell\}$, denote the maximum likelihood estimate of $\boldsymbol{\mu}(\mathcal{T})$. We define the log *likelihood* of $\mathcal{T}$ by

$$L(\boldsymbol{x}|\mathcal{T}) \equiv \log p(\boldsymbol{x}|\mathcal{T}, \widehat{\boldsymbol{\mu}}(\mathcal{T})). \tag{4}$$

In words, we consider the likelihood of the tree $\mathcal{T}$ with the parameters $\boldsymbol{\mu}(\mathcal{T})$ chosen to maximize that value (i.e., we consider the very best fit $\mathcal{T}$ can provide to the data). The log is taken for mathematical convenience (it is a monotonic transformation that preserves the ordering of the likelihood values). The maximum likelihood tree (MLT) is the one in the forest that has the largest likelihood value.

One drawback to the likelihood criterion is that it places no penalty on the number of links in the tree. As a consequence, trees with more links can have higher likelihood values (since the extra degrees of freedom they possess can allow them to fit the data more closely). In general, the true tree $\mathcal{T}^*$ will have a smaller likelihood value than another tree $\mathcal{T}$ that is identical to $\mathcal{T}^*$ except that one or more of the nodes in $\mathcal{T}^*$ are replaced with extra branching nodes that allow $\mathcal{T}$ to fit the data more closely. This is an instance of the classic "overfitting" problem associated with model estimation; the more degrees of freedom in a model, the more closely the model can fit the data. Of course, we are not interested in simply fitting the data, but rather in determining a reasonable estimate of the underlying topology.

The overfitting problem can be remedied by replacing the simple likelihood criterion with a *penalized* likelihood criterion.

$$L_\lambda(\boldsymbol{x}|\mathcal{T}) \ = \ \log p(\boldsymbol{x}|\mathcal{T}, \widehat{\boldsymbol{\mu}}(\mathcal{T})) \ - \ \lambda\, n(\mathcal{T}) \ , \tag{5}$$

where $n(\mathcal{T})$ is the number of links in the tree $\mathcal{T}$ and $\lambda \geq 0$ is a parameter, chosen by the user, to balance the trade-off between fitting to the data and controlling the number of links in the tree. The maximum penalized likelihood tree (MPLT) is defined by

$$\widehat{\mathcal{T}}_\lambda \ \equiv \ \max_{\mathcal{T} \in \mathcal{F}} L_\lambda(\boldsymbol{x}|\mathcal{T}). \tag{6}$$

This is the approach we will follow for topology identification. The MPLT determines a reasonably simple tree that accurately fits the measured data. We explicitly indicate the dependence of the MPLT on the penalty parameter λ. Setting $\lambda = 0$ will produce the MLT; it is easy to check that the MLT will always be a binary tree (with $2N - 1$ links), since binary trees provide the largest number of degrees of freedom and thus fit the data most closely. The larger the value of λ the more the penalized likelihood criterion favors simpler trees with fewer links. A discussion concerning the choice of λ is deferred to Section 4.5.

With our penalized likelihood criterion in place, the only issue remaining is the method for computing the required maximization. If the number of receivers N is small, then we can exhaustively compute the penalized likelihood value of each and every tree in $\mathcal{F}$. For large N, we have devised a more efficient approach to this maximization, which we outline next.

4. FINDING THE TALLEST TREE IN THE FOREST

Although the maximum penalized likelihood is a desirable criterion for selecting a tree, it is extremely difficult to formulate a deterministic optimization strategy that will identify $\widehat{\mathcal{T}}_\lambda$ of (6). The space of possible trees (the forest) is vast even for a relatively small number of receivers, so calculating likelihoods for all trees is infeasible. However, we contend that the likelihood surface of the forest is very peaky (only a relatively small number of trees have significant likelihood). Our goal is to design a procedure that explores the forest in an efficient manner, concentrating almost exclusively on the small set of likely trees; we also want the procedure to avoid becoming trapped in suboptimal regions of the forest. We adopt a stochastic search methodology that avoids the identification of a locally, but not globally, optimal tree.

Preferably, the search should be guided along paths where trees have a substantial likelihood. Developing such a guide is achieved by reconsidering the MPL criterion. We see from (5) that the exponentiated MPL criterion is proportional to the posterior probability:

$$\exp(L_\lambda(\boldsymbol{x}|\mathcal{T})) \ = \ \mathrm{e}^{-\lambda n(\mathcal{T})} p(\boldsymbol{x} \,|\, \mathcal{T}, \boldsymbol{\mu}) \ \propto \ p(\mathcal{T}, \boldsymbol{\mu} \,|\, \boldsymbol{x}) \tag{7}$$

provided that we adopt the prior $p(\mathcal{T}) \propto \exp(-\lambda n(\mathcal{T}))$, $\lambda > 0$ as defined above; and provided that the prior for each link $p(\mu_\ell)$ is a uniform density (constant) over $[0, \mu_{max}]$, where μ_{max} is the maximum possible μ value. The MLE of each μ_ℓ cannot be larger than the largest of the $\{x_{i,j}\}$, and therefore we set $\mu_{max} = \max_{i,j} x_{i,j}$. The priors $p(\mu_\ell)$ are *non-informative*; they have no effect other than to enforce the non-negativity constraint.

With this alternative interpretation of the penalized likelihood criterion, it is clear that we can use the posterior distribution as a guide for searching the forest. The scheme involves generating and comparing random samples $(\mathcal{T}, \boldsymbol{\mu})$ from the posterior distribution. The random samples will naturally be concentrated in the regions of high likelihood. The major task to be addressed is the generation of random samples from the posterior distribution.

4.1 Search Methodology

The basic idea is that the posterior distribution can guide a stochastic search of the forest. The stochastic search involves a set of random moves between trees, both within the parameter space of a given tree and from one tree to another. These moves are based on random samples from the posterior distribution. Because the posterior density is peaked near highly likely trees, the stochastic search focuses our exploration of the forest. Consequently, we only need to visit a small subset of the myriad of trees and compare their penalized likelihood values.

Our search methodology involves the specification of the allowed moves. The nature of the moves places restrictions on the structure of the search and the path through the forest. When defining moves, we strive to make it possible to pass between any pair of 'likely' trees in very few moves. If this is the case, then the search can begin at an arbitrary point, discover a likely tree, and then search the subset of likely trees very rapidly (because only a few moves are required to traverse the subset). Thankfully, the structure of the problem means that very natural moves impart this property. We only need three moves: a birth step, a death step, and a μ-step. The nature and effect of these moves is clarified below.

The moves perform the important function of increasing or reducing the joint measures (γ values) of groups of receivers in a structured, local fashion. There are no radical shifts in topology or parameter structure.

4.2 Reversible Jump Markov Chain Monte Carlo

Our stochastic search technique is founded on the reversible jump Markov Chain Monte Carlo methodology that was pioneered by Green in [13] and has subsequently been utilized in many model selection problems [1, 23, 25]. The first step when adopting this approach is to define a *target distribution* of interest. In our case, this is the posterior $p(\mathcal{T}, \boldsymbol{\mu}|\boldsymbol{x})$. The basic idea of MCMC methods is to simulate an ergodic Markov chain whose samples are asymptotically distributed according to the target distribution.

The important step is to construct the Markov chain with the desired asymptotic distribution. This involves the definition of a Markov transition kernel which specifies the probability of moving from one state $s_1 = (\mathcal{T}_1, \boldsymbol{\mu}_1)$ to another $s_2 = (\mathcal{T}_2, \boldsymbol{\mu}_2)$. When the dimension of the parameter space is fixed, the Gibbs sampler [12] or the Metropolis-Hastings method [16, 18] can be used to construct suitable transition kernels. The introduction of varying dimension complicates matters and more care must be taken. The reversible jump MCMC methodology provides a mechanism for developing transition kernels that generate the desired asymptotic distributions and navigate the perils of models of varying dimension.

4.3 The Moves

We begin our description of the algorithm by defining the moves between states. We denote the state at step k in the trajectory of the Markov chain by s_k. The birth-step and death-step moves are depicted in Figure 2.

In a birth step, a node ℓ that has more than two children is selected at random. Two of these children (nodes $c(\ell, 1)$ and $c(\ell, 2)$) are chosen, and an extra node ℓ^* is inserted in the topology. This node becomes the new parent of $c(\ell, 1)$ and $c(\ell, 2)$ and a child of node ℓ. The birth step increases the dimension of the model by adding the extra parameter μ_{ℓ^*}. Denote the μ values in the new tree $\mu'_{c(\ell,1)}$, $\mu'_{c(\ell,2)}$ and μ_{ℓ^*}.

We want to define the birth step so that we do not discard information we had about the $\boldsymbol{\mu}$ values in the previous state. Moreover, it is desirable to localize the changes we make to the likelihood structure. The goal of a birth step is to move to a new tree which, compared to the current tree, has higher $\gamma_{i,j}$ and $\gamma_{j,i}$ values when i is a descendant receiver of $c(\ell, 1)$ and j is a descendant receiver of $c(\ell, 2)$. These are the *only* $\boldsymbol{\gamma}$ values that we wish to affect; the likelihood can be factorized, and we only want to alter a small set of factors targeted by the birth move. We achieve these aims by specifying the following transformation that maps the $\boldsymbol{\mu}$ parameters of state s_1 to those of state s_2. Only the parameters of affected nodes are altered. The transformation hinges on the drawing of a random variable r from the uniform distribution $\mathcal{U}[0, 1]$.

$$\begin{aligned} \mu_{\ell^*} &= r \times \min(\mu_{c(\ell,1)}, \mu_{c(\ell,2)}), \\ \mu'_{c(\ell,1)} &= \mu_{c(\ell,1)} - \mu_{\ell^*}, \\ \mu'_{c(\ell,2)} &= \mu_{c(\ell,2)} - \mu_{\ell^*}. \end{aligned} \tag{8}$$

The reversible jump MCMC algorithm demands that transitions be *reversible* so that there is no chance of becoming stuck at a particular state. For this reason, the death-step is the exact opposite of the birth-step. A node ℓ^* with exactly two children is selected and deleted; the new parent of the two children becomes the parent of ℓ^*. A quick study of Figure 2 shows how a death-step (removing ℓ^*) can reverse the effect of the birth-step that introduced ℓ^*. The transformation for the death-step is completely deterministic (because it involves a dimension reduction). Using the same node labelling as before:

$$\begin{aligned} \mu_{c(\ell,1)} &= \mu'_{c(\ell,1)} + \mu_{\ell^*}, \\ \mu_{c(\ell,2)} &= \mu'_{c(\ell,2)} + \mu_{\ell^*}. \end{aligned} \tag{9}$$

Finally, the μ-step simply chooses a link ℓ at random and changes the value of μ_ℓ. This move is performed as a Gibbs sampling step, which means that the new value of μ_ℓ is drawn from the conditional posterior distribution. Drawing from this distribution rather than a distribution unrelated to the observations is very important because it has a major effect on the speed of convergence of the algorithm.

4.4 The Algorithm

The reversible jump MCMC algorithm proceeds as follows. Choose a starting state $s_0 = (\mathcal{T}_0, \boldsymbol{\mu}_0)$. This can be a random state, a default state such as the minimum link tree

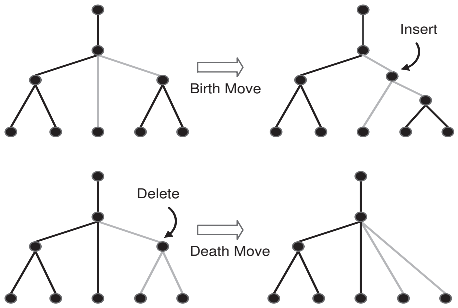

Figure 2: The birth-step and death-step moves. The birth-step selects a node with more than two children, chooses two of these children, and inserts an extra node as the new parent of these children. The death step chooses a node with two children, and deletes that node.

with random μ parameters, or a state determined by a deterministic algorithm such as that proposed in [2, 10] prior to pruning (this generally produces a fairly likely binary tree and thus initiates our exploration in a likely region of the forest)). Propose a move (birth-step, death-step or μ-step) to another state $s_1 = (\mathcal{T}_1, \boldsymbol{\mu}_1)$. Accept the proposed move with probability:

$$\min\left\{1, \frac{p(\mathcal{T}_1, \boldsymbol{\mu}_1 \,|\, \boldsymbol{x}) q(s_0 \,|\, s_1)}{p(\mathcal{T}_0, \boldsymbol{\mu}_0 \,|\, \boldsymbol{x}) q(s_1 \,|\, s_0)} \times \mathcal{J}_{f(s_1, s_0)}\right\} \tag{10}$$

Here $q(s_j|s_i)$ denotes the probability of proposing the move from state s_i to state s_j. $\mathcal{J}_{f(s_j,s_i)}$ is the Jacobian of the transformation f which maps the (possibly augmented) parameters of state s_i to those of s_j. The Jacobian is the key to addressing varying dimensionality. The transformations for the birth- and death-steps are specified by (8) and (9). In the case of the μ-step, there is no change in dimension and the Jacobian equals one.

This simple procedure is repeated many times, generating thousands of states. Whenever the Markov chain visits a tree for the first time, its log-likelihood is evaluated (by maximizing over the $\boldsymbol{\mu}$ parameter space) and stored. The goal of the algorithm is to search the forest in an efficient manner, not to generate an estimate of the posterior distribution. For this reason, we follow the procedure outlined in [4] and restart the chain whenever no new trees have been visited for a substantial period of time (a few thousand states). The chain can be restarted from the same initial state or a different state.

4.5 Setting the Penalty Parameter

The choice of λ, the penalty, determines the final topology estimate. It effectively plays the role of a threshold level; links whose likelihoods are less than a certain value will be "collapsed" by the force of the penalty term on the log-likelihood. In this paper, we follow the classical Minimum Description Length Criterion of Rissanen [24]. In rough terms, he argues that a suitable penalty can be chosen based on the "informativeness" of the data relative to the dimensionality of the parameter space. The penalty we use is $\lambda = 1/2 \log_2 N$, where N is the number of receivers.

4.6 Computational Complexity

Although the MCMC algorithm sounds computationally demanding, particularly as many thousands of states must be generated, the calculations involved in each state transition are remarkably simple. Topology estimation of relatively large trees (20-60 links) can be performed on a 600 MHz Pentium in 30-120 seconds. Considering that the probe measurements generally take on the order of minutes, there is no real penalty for the substantial performance improvement over a deterministic algorithm.

5. SIMULATION EXPERIMENTS

We conducted some simple model simulations in order to compare the performance of the maximum likelihood approach and with the Deterministic Binary Tree (DBT) classification [9]. The DBT was designed for multicast topology identification, but could be applied as a less computationally intensive alternative to our approach. In this simulation we set $\lambda = 0$, so that there is no penalty. In this case, both DBT and MPLT approaches will produce binary trees. Therefore, we considered the six receiver binary topology used in the ns-simulations in [9]. The measurements were generated according to our model and the standard deviation of the delay differences on each link was proportional to the respective mean value. We do not report all the details of the experimental setup here due to space limitations. In each experiment we sent 50 probes to each receiver pair, and computed the DBT and MPLT trees for 1000 independent simulations. The DBT algorithm correctly identified 935 trees, and the MPLT procedure identified 951 trees. Although the MPLT performed slightly better, the percentage of correctly identified trees is almost the same for both procedures. Performance differences are much greater when the variabilities of the delay difference measurements differ on different links. We performed the same experiment as before, but setting the standard deviation of one of the receiver links three times greater than before. In this scenario the DBT method correctly identified 734 tree; the MPLT method identified 912 trees correctly. This demonstrates that the maximum likelihood criterion can provide significantly better identification results than the DBT approach.

We also conducted a series of `ns` simulations [19] using the topology depicted in Figure 3. The topology has nine receivers and link bandwidths ranging from 0.5 to 10 Mb/s. The maximum depth of the physical topology is 8, and the maximum depth of the logical topology is 6.

Simulations were conducted in both a low utilization scenario and a higher utilization scenario (by varying background traffic). In the former scenario, the average utilization over all links and runs was 30 %, with a range of 5-50 %; in the latter, the average was 45 %, with a range of 10-90 %. We performed 25 runs of each scenario, with each run lasting for 8 minutes. Sandwich probes were sent into the network with a mean probe separation of 50 ms, and the receiver pairs chosen at random. On average, the full duration results in 9600 probe measurements. The spacing between the first packet and the second packet was set to 20 ms.

Figure 4(a) examines the performance of the proposed algorithm and measurement scheme as the number of probes is varied. In the algorithm, we have set the complexity penalty to $\lambda = 0.5 \log_2 N$, as discussed in Section 4.5. As expected, we observe a steady improvement in performance as the number of probes increases. The performance of the algorithm in a lightly loaded network is better than in a network experiencing higher load, because the cross-traffic induced variation in the measurements is less of a factor. Figure 4(b) examines the sensitivity of the algorithm to the choice of the penalty λ. In this instance, the performance is similar over a range of λ from approximately 1 to 3.

6. INTERNET EXPERIMENT

We have implemented a software tool called `nettomo` that performs sandwich probing measurements and estimates the topology of a tree-structured network. The software has been implemented as a set of C programs using the Unix socket library and the programs have been ported to Solaris, FreeBSD, and (some) Linux platforms. The topology estimation (data collection and inference) is performed at a source host. The program at the source sends UDP sandwich probes to a set of remote clients, which are required to run a low overhead receiver task during the measurement period. The receiver task primarily time-stamps a received UDP small packet (with the local time) and returns the time-stamped packet to the sender via a dedicated TCP connection. The sender software maintains a log of the returned packets, and at the completion of the measurement period, calculates the delay differences, the associated metrics and their variability.

We conducted Internet experiments using the topology depicted in Figure 5(a). The source for the experiments was located at Rice University. There were ten receiver clients, two located on different networks at Rice, two at separate universities in Portugal, and six located at four other US universities.

We performed the same experiments six times, on different days and at different times so that traffic conditions varied. Each experiment was conducted for a period of eight minutes, during which a sandwich probe was sent to a randomly chosen receiver-pair once every 50 ms. Without any loss, the maximum number of probes available is 8600; we experienced no probe loss in five of the six experiments, and less than 0.1 percent probe loss in the sixth.

We applied the topology identification procedure outlined in the paper, choosing a complexity penalty of 1.7, which is derived from the formula $0.5 \log_2 N$ suggested in Section 4.5 ($N = 10$, the number of receivers). When this penalty was applied, the algorithm generated the topology in Figure 5(b) in four out of the six experiments. The estimated topology places an extra shared link between the Rice computers. Upon further investigation, we discovered that there is a layer-2 switching element within the Rice network; in contrast to traceroute, the measurement procedure can detect layer-2 branching points. The proposed methodology fails to detect the backbone connection between Texas and Indianapolis. We expect that the connection is very high speed so that the queuing effects on the constituent routers are too minor to influence measurements. An alternative explanation is that the link performs systematic prefix-based load balancing, which can disrupt sandwich probe measurement (the assumption of a tree topology is no longer valid). In the other two experiments, the complexity penalty was sufficiently large to collapse the link shared only by the two Univ. Wisconsin computers.

7. CONCLUSIONS

This paper considered the problem of discovering network topology solely from host-based, unicast measurements, without internal network cooperation. We introduced a novel measurement scheme based on special-purpose unicast probes that we call "sandwich" probes. The sandwich probing scheme is delay-based, but only requires local delay difference measurements at each receiver host, so that no clock synchronization is required. The sandwich probing scheme may provide a more reliable metric than other proposed metrics (loss, delay correlations, utilization) used for topology identification. We also developed a new, penalized likelihood-based framework for topology identification. A major advantage of the framework is that it is based on a global optimality criterion, in contrast to other recent proposals for topology identification that employ suboptimal, local grouping and pruning strategies. We propose a novel Markov Chain Monte Carlo (MCMC) procedure for rapid determination of the most likely topologies.

One area for future work is to devise adaptive methods for selecting the penalty parameter λ. In the experiments reported in this paper, we set λ according to the Minimum Description Length principle. However, one may be able to better choose the penalty after computing the likelihoods of a set of highly likely trees. This suggests an alternative approach, closer to the pruning methodology adopted in [2, 9]: run a threshold test on the estimated $\boldsymbol{\mu}$ parameters associated with the set of most likely trees, reject all trees where one or more links fails the test, and choose the most likely remaining tree. There is an important advantage of the methodology over pruning techniques, however. If a topology is not a binary tree, there is a danger associated with forcing a binary tree model upon it, as is done by the algorithms in [9, 2]; incorrect end-to-end metric estimates can result in selection of a binary tree from which the correct topology cannot be obtained through pruning. The effects of measurement noise on the identification capabilities of these algorithms are far more noticeable when there is a model mismatch.

Another direction for future improvement is to incorporate a degree of adaptivity into the probing scheme. The sample variances $\{\sigma_{i,j}^2\}$ provide confidence measures for the metrics $\{x_{i,j}\}$. One could begin the probing process with a small

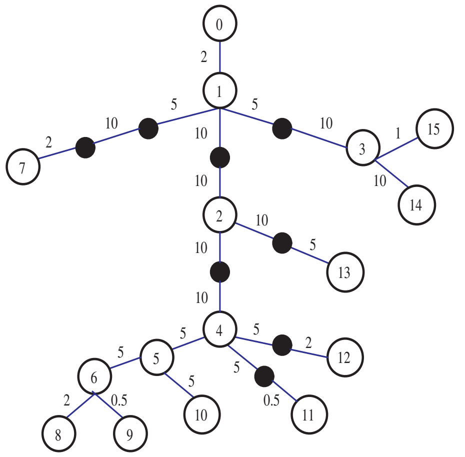

Figure 3: The topology used for the ns experiments. The black circles represent queues where no branching occurs; the hollow circles are branching nodes. The connections between the hollow circles form the logical topology. Link capacities (Mb/s) are shown next to each link.

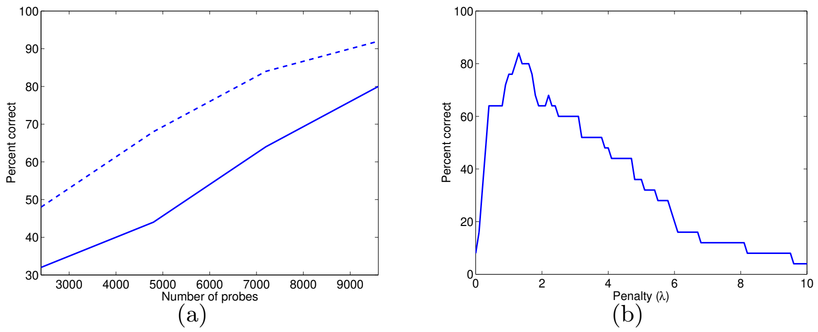

Figure 4: (a) The performance of the identification algorithm as a function of the number of probe measurements (dashed line - light utilization, solid line - heavier utilization). (b) The sensitivity of the algorithm to the choice of penalty. The plot shows the percentage of correctly identified trees as a function of the penalty λ in the case of the heavier utilization scenario and 9600 probe measurements.

number of probes (10-100) sent to each receiver pair. Then, based on the sample variances of those measurements, additional probing could be directed at those pairs with larger variances. This focused probing step would improve the stability of the topology identification process in an efficient manner, and it could drastically reduce probing requirements.

8. ADDITIONAL AUTHORS

Additional authors: Manik Gadhiok, Ryan King and Yolanda Tsang (Dept. Electrical and Computer Engineering, Rice University, email: `gadhiok,rking,ytsang@rice.edu`).

9. REFERENCES

[1] C. Andrieu, A. Doucet, W. Fitzgerald, and J. Peréz. Bayesian computational approaches to model selection. In *Nonlinear and Non Gaussian Signal Processing, Cambridge: Newton Institute Series*. Cambridge University Press, 2000.

[2] A. Bestavros, J. Byers, and K. Harfoush. Inference and labeling of metric-induced network topologies. Technical Report BUCS-2001-010, Computer Science Department, Boston University, June 2001.

[3] R. Cáceres, N. Duffield, J. Horowitz, and D. Towsley. Multicast-based inference of network-internal loss characteristics. *IEEE Trans. Info. Theory*, 45(7):2462–2480, November 1999.

[4] H. Chipman, E. George, and R. McCulloch. Bayesian CART model search (with discussion). *J. Am. Stat. Assoc.*, 93(6):937–960, 1998.

[5] M. Coates and R. Nowak. Network loss inference using unicast end-to-end measurement. In *ITC Seminar on IP Traffic, Measurement and Modelling*, Monterey, CA, Sep. 2000.

[6] M. Coates and R. Nowak. Network delay distribution inference from end-to-end unicast measurement. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proc.*, Salt Lake City, UT, May 2001.

[7] M. Coates and R. Nowak. Sequential Monte Carlo inference of internal delays in nonstationary data networks. *IEEE Trans. Signal Processing*, 50(2):366–376, Feb. 2002.

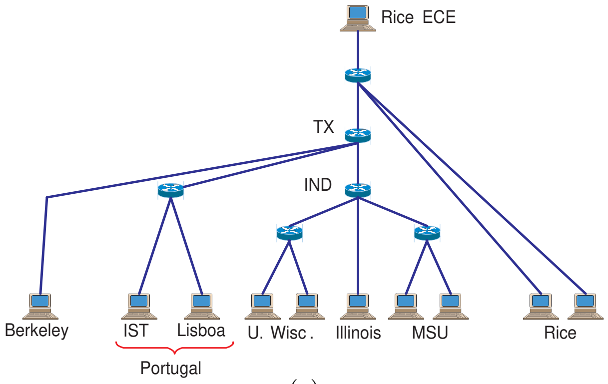

Figure 5: (a) The traceroute-determined topology of the network used for Internet experiments. This was stable for all experiments. (b) Most commonly estimated topology using the MPL criterion. The link between TX and IND is not detected, but a layer-2 switching element within the Rice network is identified. Otherwise, the estimated topology is a faithful representation of the traceroute-determined topology.

[8] N. Duffield, J. Horowitz, and F. L. Presti. Adaptive multicast topology inference. In *Proceedings of IEEE INFOCOM 2001*, Anchorage, Alaska, April 2001.

[9] N. Duffield, J. Horowitz, F. L. Presti, and D. Towsley. Multicast topology inference from end-to-end measurements. In *ITC Seminar on IP Traffic, Measurement and Modelling*, Monterey, CA, Sep. 2000.

[10] N. Duffield, J. Horowitz, F. L. Presti, and D. Towsley. Multicast topology inference from measured end-to-end loss. *IEEE Trans. Info. Theory*, 48(1):26–45, January 2002.

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

[12] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. PAMI*, 6(6):721–741, 1984.

[13] P. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732, 1995.

[14] K. Harfoush, A. Bestavros, and J. Byers. Robust identification of shared losses using end-to-end unicast probes. In *Proc. IEEE Int. Conf. Network Protocols*, Osaka, Japan, Nov. 2000. *Errata* available as Boston University CS Technical Report 2001-001.

[15] K. Harfoush, A. Bestavros, and J. Byers. Measuring bottleneck bandwidth of targeted path segments. Technical Report BUCS-2001-016, Computer Science Department, Boston University, July 2001.

[16] W. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[17] K. Lai and M. Baker. Measuring link bandwidths using a deterministic model of packet delay. In *Proc. ACM SIGCOMM 2000*, Stockholm, Sweden, Aug. 2000.

[18] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equations of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1091, 1953.

[19] UCB/LBNL/VINT network simulator ns (version 2). www.isi.edu/nsnam/ns/.

[20] V. Paxson. End-to-end Internet packet dynamics. *IEEE/ACM Trans. Networking*, 7(3):277–292, June 1999.

[21] F. L. Presti, N. Duffield, J. Horowitz, and D. Towsley. Multicast-based inference of network-internal delay distributions. Technical Report CMPSCI 99-55, University of Massachusetts, 1999.

[22] S. Ratnasamy and S. McCanne. Inference of multicast routing trees and bottleneck bandwidths using end-to-end measurements. In *Proceedings of IEEE INFOCOM 1999*, New York, NY, March 1999.

[23] S. Richardson and P. Green. On Bayesian analysis of mixtures with an unknown number of components. *J. Roy. Stat. Soc. B*, 59:731–792, 1997.

[24] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*. World Scientific, Singapore, 1989.

[25] C. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Verlag Series in Statistics, New York, 1998.

[26] D. Rubenstein, J. Kurose, and D. Towsley. Detecting shared congestion of flows via end-to-end measurement. In *Proc. ACM SIGMETRICS 2000*, Santa Clara, CA, June 2000.