Loss-based Inference of Multicast Network Topology [1]

R. Cáceres *, N.G. Duffield *, J. Horowitz §, F. Lo Presti *, §, D. Towsley §

*AT&T Labs–Research § University of Massachusetts

Abstract

The use of multicast inference on end-to-end measurement has recently been proposed as a means to infer network internal characteristics such as packet loss rate and network topology. In this paper we propose and evaluate new algorithms for multicast topology inference based on measurement of end-to-end loss. We compare their accuracy and comment on their computational complexity.

1 Introduction

Inference of network-internal characteristics from the end-to-end behavior of multicast traffic has been proposed in recent papers. Probes are multicast from the source to receivers; the record of which probes reached each receiver is used to infer internal loss probabilities. Probe loss is assumed to occur independently across links and between probes. In [1, 2] the Maximum Likelihood Estimator (MLE) of link loss probabilities was determined for general logical multicast trees. It was shown that the estimator is reasonably robust with respect to violations of this assumption, at least on the scale found in realistic network simulations. In a related paper [5], the focus was to group multicast receivers that share the same set of network bottlenecks from the source. This used the special case of the estimator for binary trees in [1] in order to estimate loss on the common portion of the path from the source to each receiver. An extension of the binary approach to treat more general trees was also proposed.

The contributions of the present paper are as follows:

(i) We propose three algorithms for topology identification; these generalize the grouping algorithms of [5]. Two of these use binary grouping as an intermediate stage of identification; the other uses the approach of [1, 2] to estimate the loss on the common portion the paths from the source to an arbitrary set of receivers (i.e. not necessarily a binary set).

(ii) We propose an algorithm that uses the MLE from [1] directly. In this approach, for each possible tree built upon the given set of receivers, the MLE for its link probabilities is found and the corresponding likelihood function is calculated. The tree that maximizes the this likelihood function is selected as the estimator. We establish that the probability of selecting the actual tree converges to 1 as the number of probes grows to infinity.

(iii) We compare the performance of the algorithms and discuss the interactions between parameter choices and underlying topology that determine their accuracy

The use of end-to-end measurement has the advantage of not requiring cooperation from network management. The use of multicast probes has the advantage over unicast probes that the network load due to probes scales better for large trees. For a more detailed discussion of implementational issues, the potential for deployment on emerging Internet measurement infrastructures, and a comparison with other approaches, we refer the reader to [1].

2 Description of the Loss Inference Methodology

We identify the physical multicast tree as comprising actual network elements (the nodes) and the communication links than join them. The logical multicast tree comprises the branch points of the physical tree, and the logical links between them. The logical links comprise one or more physical links. Thus each node in the logical tree, except the leaf nodes and possibly the root, must have 2 or more children. We can construct the logical tree from the physical tree by deleting all links with one child and adjusting the links accordingly by directly joining its parent and child.

Let $T = (V, L)$ denote a logical multicast tree with nodes V and links L. We identify one node, the root 0, to be the source of probes, and $R \subset V$ to be the set of leaf nodes (identified as the set of receivers). The set of children of node $j \in V$ is denoted by $d(j)$. Each node, k, apart from the root has a parent $j = f(k)$ such that $(j, k) \in L$. We say j is descended from k, and write $j \prec k$, if $k = f^n(j)$ for some $n \in \mathbb{N}$, where $f^n = f \circ f^{n-1}$ and $f^1 = f$. $a(S)$ will denote the $\prec$-least common ancestor of all nodes in $V \subset S$.

[1] This work was sponsored in part by the DARPA and the Air Force Research Laboratory under agreement F30602-98-2-0238.
Ramón Cáceres, Nick Duffield, Francesco Lopresti; AT&T Labs-Research, Rm. {B125, B139, B130}, 180 Park Avenue, Florham Park, NJ 07932, USA. E-mail: {ramon,duffield,lopresti}@research.att.com
Joseph Horowitz, Dept of Math. & Statistics, University of Massachusetts, Amherst, MA 01003-4515 USA; E-mail: joeh@math.umass.edu
Don Towsley, Dept. of Computer Science, University of Massachusetts, Amherst, MA 01003-4610, USA; E-mail: towsley@cs.umass.edu

We assume a Bernoulli loss model in which probes are independent and each probe is successfully transmitted across the link $(f(k), k)$ with probability α_k. Thus the progress of each probe down the tree is described by an independent copy of a stochastic process $X = (X_k)_{k \in V}$ as follows. $X_k = 1$ if the probe reaches node $k \in V$ and 0 otherwise. If $X_k = 0$, then $X_j = 0, \forall j \in d(k)$. Otherwise, $P[X_j = 1 | X_k = 1] = \alpha_j$ and $P[X_j = 0 | X_k = 1] = 1 - \alpha_j$. We adopt the convention $\alpha_0 = 1$. Last, let $\alpha = (\alpha_i)_{i \in V}$. We assume that $0 < \alpha_k < 1, \forall k \in V$. P_α will denote the distribution of X under the link probabilities α.

When a probe is sent down the tree from the root 0, we cannot observe the whole process X, but only the outcome $(X_k)_{k \in R} \in \Omega = \{0,1\}^R$ that indicates whether or not the probe reached each receiver. In [1] it was shown how the link probabilities can be inferred from the measured distribution of outcomes due to a set of independent probes when the topology $T = (V, L)$ is known. Let $R(k) \subset R$ denote the set of receivers descended from a node $k \in V$. Set

$$Y(k) = \max_{j \in R(k)} X_j, \quad \text{and} \quad \gamma(k) = \mathsf{P}[Y(k) = 1]. \tag{1}$$

$Y(k) = 1$ if a probe reaches at least one receiver descended from k. For $j \in V$ set $A(j) = \alpha_j \alpha_{f(j)} \ldots \alpha_0$, the probability that a probe reaches the node j. It can be shown that the A and the γ satisfy

$$(1 - \gamma(k)/A(k)) = \prod_{j \in d(k)} (1 - \gamma(j)/A(k)). \tag{2}$$

It was shown in Lemma 1 of [1] that $A(k)$ is the unique solution to (2) in $(\gamma(k), 1]$ provided that $\gamma(k) < \sum_{j \in d(k)} \gamma(j)$. Thus the collection of $A(k)$, and hence also the $\alpha_k = A(k)/A(f(k))$, can be determined from the $\gamma(k)$. At a node k with two offspring $\{j, j'\}$, (2) reduces to $A(k) = \gamma(j)\gamma(j)/(\gamma(j) + \gamma(j') - \gamma(k))$.

In order to estimate the $A(k)$ from measurements, n probes are multicast from the source, giving rise to outcomes $\widehat{X}^{(1)}, \ldots, \widehat{X}^{(n)}$. Define $\widehat{\gamma}(k)$ as the proportion of outcomes in which a packet was seen at a receiver descended from k, and estimate $A(k)$ by $\widehat{A}(k)$, the solution to (2) that results from using the $\widehat{\gamma}$ in place of γ. The *log-likelihood function* $\mathcal{L}(T, \alpha; \gamma)$ is the log-probability of obtaining the frequencies γ, given a set of link probabilities α in a topology T. With measured frequencies $\widehat{\gamma}$, the MLE of the link probabilities is the α that maximizes $\mathcal{L}(T, \alpha; \widehat{\gamma})$.

Theorem 1 *(i) The model is identifiable, i.e. $P_\alpha = P_{\alpha'}$ implies $\alpha = \alpha'$.*

(ii) with probability 1, for sufficiently large n, $\widehat{A}$ is the Maximum Likelihood Estimator of A.

As a consequence of the MLE property, $\widehat{A}$ is (strongly) consistent ($\widehat{A} \stackrel{n \to \infty}{\longrightarrow} A$ with probability 1), and asymptotically normal ($\sqrt{n}(\widehat{A} - A) \stackrel{n \to \infty}{\longrightarrow} G$ for some multivariate Gaussian random variable G); see [6].

3 Topology Inference by Grouping

The use of the estimate of losses on the common portion of the path between receivers in binary trees has been proposed recently in order to group multicast receivers that share the same set of bottlenecks on the path from the source [5]. In this and the succeeding section we shall investigate the analytic and experimental properties of this algorithm and some generalizations that infer general trees (i.e. those which are not necessarily binary).

We shall work exclusively with **canonical loss trees**; a loss tree consists of a tree-loss rate combination (T, α). A loss tree is said to be in canonical form if $\alpha_k < 1, \forall k \in V$ except for $k = 0$. Any tree (T, α) not in canonical form can be reduced to a loss tree, (T', α'), in canonical form such that the distribution of $(X_k)_{k \in R}$ is the same under the corresponding probabilities $\mathsf{P}_{T,\alpha}$ and $\mathsf{P}_{T',\alpha'}$.

Consider the class of maps of the form

$$Q \mapsto Q' = (Q \setminus S) \cup \{S\}, \quad \text{with} \quad S \subseteq Q, \#S > 1, \tag{3}$$

on finite sets Q. Starting with Q as the receiver set R, such an operation represents identifying the elements of S as a maximal sibling set, grouping them together and identifying the set S with their parent. This operation results in the set Q' comprising the parent node S and all remaining nodes in $Q \setminus S$. By composing a number of such maps we can build up a logical multicast tree; the problem of recovering the actual tree can then be re-expressed as that of finding an appropriate set of grouping maps. The inference algorithms described below depend on choosing as S the subset in which the estimate of common loss (i.e. loss rates on the shared portion of the path from the source to the receivers) is greatest. For general subsets $S \subset V$ we define $Y(S) = \max_{k \in S} Y(k)$ and $\gamma(S) = \mathsf{P}[Y(S) = 1]$. Let $A(S)$ denote the solution of equation (2) when the product extends over all $j \in S$.

3.1 Inference of Binary Trees

The method is simplest to explain when the tree to be inferred is known to be binary. Pick any two nodes j and k. According to Section 2 above, $\widehat{A}(\{j,k\}) := \widehat{\gamma}(j)\widehat{\gamma}(k)/(\widehat{\gamma}(k) + \widehat{\gamma}(k) - \widehat{\gamma}(\{j,k\})$ is the MLE of the probability of successful transmission from the root to the lowest common ancestor node $a(\{j,k\})$ of the receivers j, k. But observe now that in general, since each $A(i)$ can be expressed as a product of link probabilities over each link on a chain from the root node to i, then the quantity $A(i)$ is decreasing down the tree in the sense that $A(i) > A(i')$ is i' is a descendant node of i. (This assumes that all the link probabilities α_k are strictly less than 1). Hence $\widehat{A}(\{j,k\})$ should be minimized when j and k are siblings, at least asymptotically as the number of probes grows to infinity. The iteration of this procedure to infer the complete logical multicast tree is formalized as the **binary loss tree** (BLT) classification algorithm; see Figure 1. (We note that the definition of $\widehat{\gamma}$ and

1. *Input*: Receivers R and receiver traces $(X_k^i)_{k\in R}^{i=1,2,\ldots N}$
2. $R' := R$, $V := \emptyset$; $L = \emptyset$;
3. **while** $|R'| > 1$ **do**
4. **select** $S = \{S_1, S_2\} \subset R'$ that minimizes $\widehat{A}(\cdot)$;
5. $V := V \cup \{S\}$; $R' := (R' \setminus S) \cup \{S\}$;
6. **foreach** $S' \in S$ **do**
7. $\alpha(S') := A(S)/A(S')$; $L := L \cup \{(S, S')\}$;
8. **enddo**
9. **enddo**
10. *Output*: loss tree $((V, L), \alpha)$

Figure 1: Binary Loss Tree Classification Algorithm (BLT).

$\widehat{A}$ extends from subset of nodes of the actual tree to those of any potential tree in a straightforward manner).

Consider a loss tree (T, α). Let $\mathcal{A}$ be an algorithm that takes a trace of length n, $\{(X_{k,i})_{k\in R}\}_{i=1}^n$ as an input and returns a loss tree $(\widehat{T}, \widehat{\alpha})$. Let $P_{\mathcal{A}}^f = P(\widehat{T} \neq T)$ be the probability that $\mathcal{A}$ misclassifies the tree.

Theorem 2 *(i) A canonical loss tree is identifiable, i.e.,* $\mathsf{P}_{T,\alpha} = \mathsf{P}_{T',\alpha'}$ *implies* $(T, \alpha) = (T', \alpha')$.

(ii) For any binary canonical loss tree (T, α), $P_{BLT}^f \to 0$ *as* $n \to \infty$.

Sketch of Proof: Consider the *deterministic* algorithm obtained by replacing $\widehat{A}$ with A in line 4 of Figure 1. Then the tree is correctly identified, essentially because

$$a(\{i,j\}) \prec a(\{i,k\}) \implies A(\{i,j\}) < A(\{i,k\}). \quad (4)$$

By the Strong Law of Large Numbers $\widehat{\gamma}(S)$ converges $\gamma(S)$), almost surely as $n \to \infty$; by continuity $\widehat{A}(S)$ converges likewise to $A(S)$. Hence, by (4), $\widehat{A}(\{i,j\}) < \widehat{A}(\{i,k\})$ for all n sufficiently large, where $\{i,j\} \subset R'$ is any binary set that minimizes A in the deterministic algorithm. Thus $P_{BLT}^f \to 0$ as $n \to \infty$. ■

3.2 Inference of General Trees by Grouping

In this section we describe extensions of the binary loss tree classification algorithm to treat general trees; the description here will be somewhat informal. The Binary Loss Tree Pruning (BLTP) and the Binary Loss Tree Clique (BLTC) Classification Algorithms estimate loss on the common portion of the path to *pairs* of nodes; the former is inspired by a proposal in [5]. The General Loss Tree (GLT) Classification Algorithm uses the estimate from (2) of loss on the common portion of the path to *arbitrary* sets of nodes. All the algorithms are governed by a threshold $\varepsilon > 0$ which is the smallest link loss probability that can be inferred. When all link loss probabilities are greater than ε, the probability that the tree is incorrectly classified goes to zero as the number of probes grows to infinity. We compare the performance of these algorithms in Section 5.

4a. **select** $S^{(2)} = \{S_1, S_2\} \subset R'$ that minimizes $\widehat{A}(\cdot)$;
4b. $m := 2$; done:= false;
4c. **while** not done **do**
4d. $\mathcal{U} := \{U = S^{(m)} \cup S_{m+1} \mid S_{m+1} \in R' \setminus S^{(m)}, (1-\varepsilon)\widehat{A}(U) < \widehat{A}(S^{(2)})\}$;
4e. **if** $\mathcal{U} = \emptyset$ **then** $\{S := S^{(m)}$; done := true;$\}$
4f. **else select** $S^{(m+1)} \in \mathcal{U}$ that minimizes $\widehat{A}(\cdot)$;
4g. **enddo**

Figure 2: General Loss Tree Classification Algorithm (GLT): additional steps to be substituted for line 4 of Figure 1.

3.2.1 Binary Loss Tree Pruning Classification: For each $\varepsilon > 0$ we define BLTP(ε) by a two step process: (i) applying BLT to the receiver traces resulting in a binary loss tree $((V, L), \alpha)$; then (ii) for each node k with loss probability $1 - \alpha_k < \varepsilon$, remove the link $(f(k), k)$ from L and identify the endpoints in V.

3.2.2 Binary Loss Tree Clique Classification: In BLTC(ε), we group together pairs of nodes whose loss on the common path portion is close to the maximum common loss over all pairs. This amounts to replacing line 4 in Figure 1 with the following steps: (i) select $S' = \{S_1', S_2'\}$ that minimizes $\widehat{A}(\cdot)$; (ii) construct the graph G of all links (S_1'', S_2'') such that $(1-\varepsilon)\widehat{A}(\{S_1'', S_2''\}) < \widehat{A}(S')$; (iii) group as the next node S the elements of the largest connected component of G that contains S'. Note that if the grouping is done correctly, then the deterministic common loss $A(\{S_1, S_2\})$ takes the same value for all binary subsets $\{S_1, S_2\}$ of S. For finite but large n, the corresponding random $\widehat{A}(\{S_1, S_2\})$ will differ slightly. Choosing $\varepsilon > 0$ enables grouping of these nodes for finite n. However, this can also lead to the inclusion of nodes which are separated by links with loss probability less than about ε.

3.2.3 General Loss Tree Classification: GLT(ε) is essentially an extension of BLT in which the minimization in line 4 of Figure 1 extends over arbitrary subsets of R'. Since all subsets of a set of siblings have the same value of A, we employ a threshold ε to perform the grouping, similar to BLTC(ε). In practice, we build up the minimizing set S by finding (approximately) minimizing sets of increasing cardinality. More precisely, the rule to select the next node S in line 4 of Figure 1 is replaced by the lines shown in Figure 2.

4 Maximum Likelihood Classification of General Trees

A second approach to topology inference is to use the likelihood function itself for classification. Let $\mathcal{T}$ denote the set of logical multicast trees that can be laid out between the root and the set of receivers. The topology τ of a multicast tree is determined by the set U_τ of non-root nodes and the

links between them.

Let $\check{\alpha}_\tau \in [0,1]^{U_\tau}$ denote the MLE for the link probabilities α if the topology were $\tau \in \mathcal{T}$; let $\widehat{\gamma}_\tau$ denote the set of measured probabilities $\widehat{\gamma}(k)$ at nodes k of τ.

The *maximum-likelihood (ML) classifier* assigns the topology $\widehat{\tau}$ that maximizes $\mathcal{L}(\tau, \widehat{\alpha}_\tau, \widehat{\gamma}_\tau)$. We now show that, if the link probabilities are bounded away from 0 and 1, the ML-classifier is *consistent* in the sense that, w.p. 1, it identifies the correct topology as the number of probes tends to infinity. Let $\mathcal{A}_\tau^\varepsilon = [\varepsilon, 1-\varepsilon]^{U_\tau}$, where $\varepsilon > 0$, and consider trees with link probabilities $\alpha \in \mathcal{A}_\tau^\varepsilon$.

Theorem 3 *For each topology τ_0 and link probability vector $\alpha_0 \in \mathcal{A}_\tau^\varepsilon$,*

$$\lim_{n\to\infty} P_{\tau_0,\alpha_0}(\widehat{\tau} = \tau_0) = 1. \tag{5}$$

Proof: It suffices to show that, for each $\tau \neq \tau_0$,

$$\lim_{n\to\infty} P_{\tau_0,\alpha_0}(\widehat{\tau} = \tau) = 0. \tag{6}$$

Let $p(x;\tau,\alpha)$ denote the density (i.e., probability mass function) for the configuration x due to a single probe and let E_0 denote expectation with respect to P_{τ_0,α_0}. Under our assumptions, if $\tau \neq \tau_0$, the Kullback-Leibler information

$$I_0(\alpha) = \mathsf{E}_0(\log(p(X;\tau_0,\alpha_0)/p(X;\tau,\alpha))) \tag{7}$$

is a continuous function of $\alpha \in \mathcal{A}_\tau^\varepsilon$, and $I_0(\alpha) > 0$ (identifiability). Thus there is a number $\delta > 0$ such that $I_0(\alpha) \geq \delta$ for all $\alpha \in \mathcal{A}_\tau^\varepsilon$. Now

$$P_{\tau_0,\alpha_0}(\widehat{\tau} = \tau) \leq \\ P_{\tau_0,\alpha_0}(\exists \alpha \in \mathcal{A}_\tau^\varepsilon : \frac{1}{n}\sum_1^n \log\frac{p(X_i;\tau,\alpha)}{p(X_i;\tau_0,\alpha_0)} \geq 0). \tag{8}$$

Since $\alpha \in \mathcal{A}_\tau^\varepsilon$, the density $p(x;\tau,\alpha)$ is bounded away from zero, hence the conditions of Jennrich's [3] uniform strong law of large numbers are satisfied. Thus

$$\frac{1}{n}\sum_1^n \log\frac{p(X_i;\tau,\alpha)}{p(X_i;\tau_0,\alpha_0)} \longrightarrow -I_0(\alpha) \leq -\delta \tag{9}$$

uniformly in $\alpha \in \mathcal{A}_\tau^\varepsilon$, whence the last displayed probability tends to zero as $n \to \infty$. ■

5 Algorithm Comparison and Evaluation

The topology classification algorithms described above were evaluated through simulation. The ns [4] simulator was used in order to provide a realistic simulation of the occurrence of packet losses in a network.

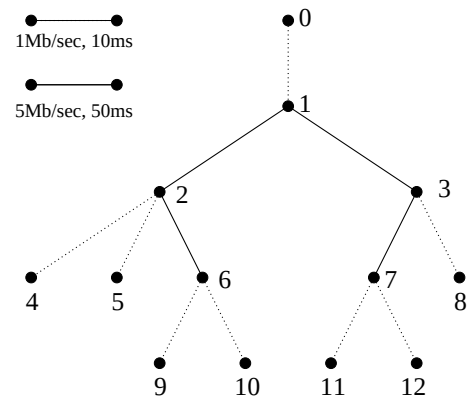

Figure 3: Simulation Topology: Links are of two types: *edge* links of 1MB/s capacity and 10ms latency, and *interior* links of 5Mb/s capacity and 50ms latency.

5.1 Comparison of Grouping Algorithms

We report results obtained in the network topology shown in Figure 3. Links in the interior of the tree have higher capacity (5Mb/sec) and latency (50ms) than those at the edge (1Mb/sec and 10ms) in order the capture the heterogeneity between the edges and core of a Wide Area Network. Probes are generated from node 0 as a Poisson process with mean interarrival time 16ms. Background traffic comprised a mix of infinite data source TCP connections (FTP) and exponential on-off sources using UDP. Packet losses were due to overflow from 4 packet buffers at each node. Since the losses were not generated explicitly from a model, they could potentially violate the Bernoulli assumptions. However, we have found in previous work [1] that the discrepancy between actual and inferred probe losses is very small. (The discrepancy with background traffic losses is greater, although still quite small). Each evaluation comprised 100 simulations each of 300s duration. Individual link loss rates ranged from as little as 0.6% to as high as 41%. Mean link loss rates ranged from slightly below 1% to 12%.

In Figures 4-9 we plot the proportion of experiments in which the topology was correctly identified as function of the number of probes, for the three algorithms, for values of the threshold ε between 0.025% and 5%. In summary, accuracy is best for intermediate ε, decreasing for larger and smaller ε. In more detail, consider for example, the BLT Pruning algorithm with 1000 probes. The topology is incorrectly identified 45 times for $\varepsilon = 0.25\%$, 14 times for $\varepsilon = 0.5\%$ and only 5 times for $\varepsilon = 1\%$. To obtain the same increase in accuracy at $\varepsilon = 0.25\%$ would require about 6000 probes. The explanation for this behavior is that smaller values of ε lead to stricter criteria for grouping nodes. Finitely many samples, fluctuations of the $\widehat{A}$ below their mean can lead to erroneous exclusion of nodes from groups. Increasing ε leads to errors when links with loss probabilities $1 - \alpha_k < \varepsilon$ are effectively ignored, so that the groups are chosen too large; observe the probability of successful identification decreases dramatically as ε increases

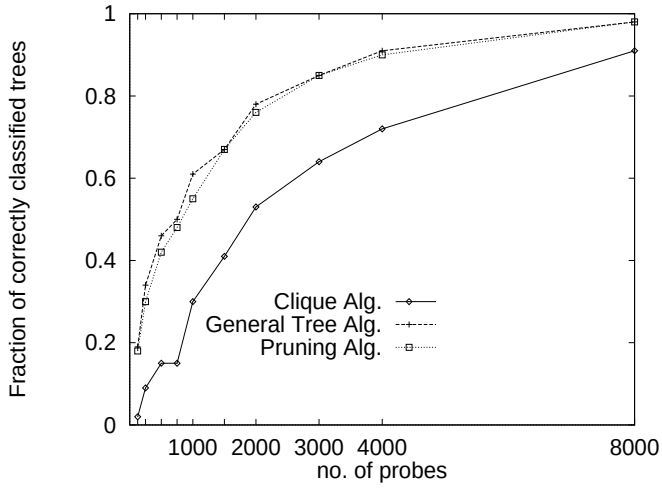

Figure 4: $\varepsilon = 0.25\%$.

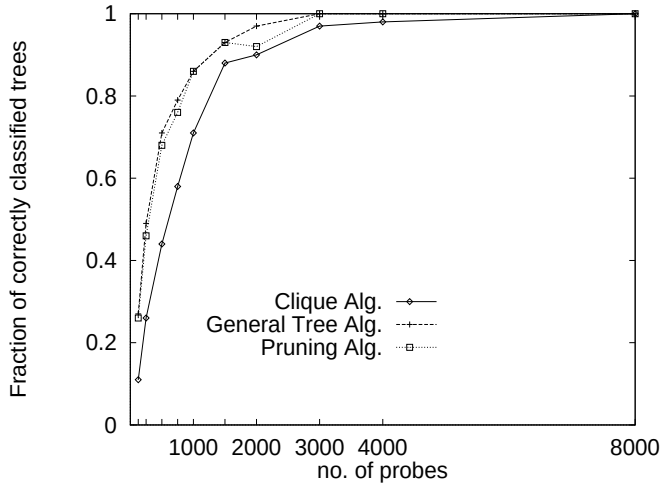

Figure 5: $\varepsilon = 0.5\%$.

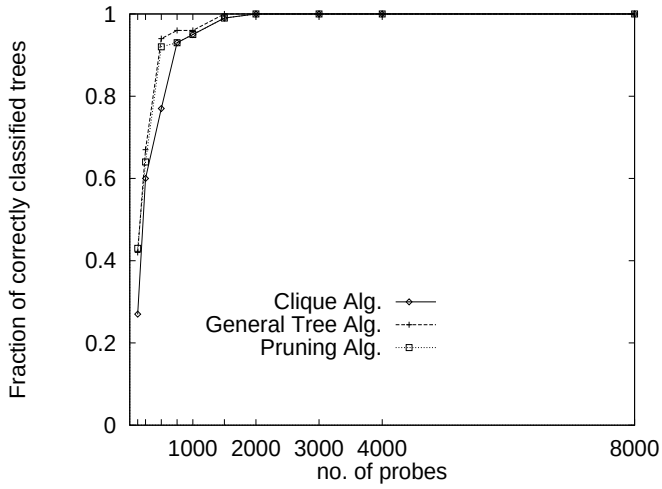

Figure 6: $\varepsilon = 1.0\%$.

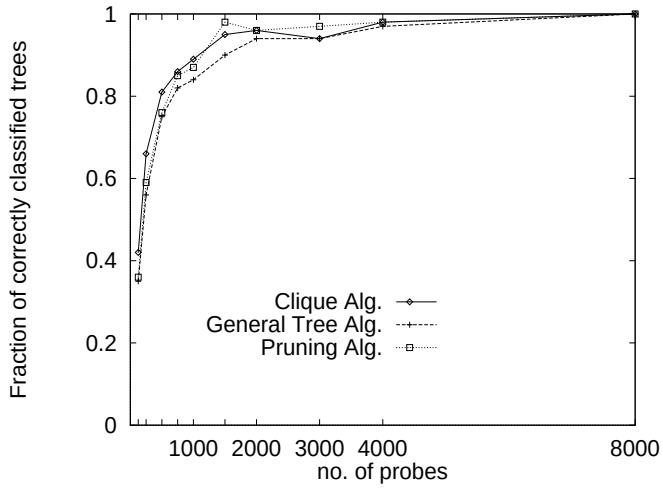

Figure 7: $\varepsilon = 2.0\%$.

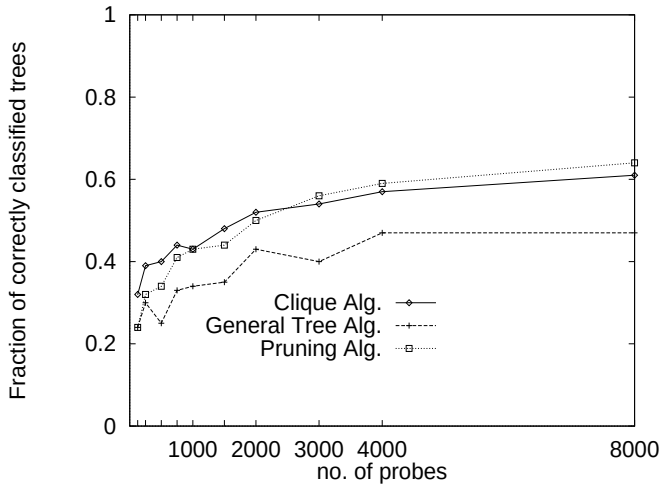

Figure 8: $\varepsilon = 3.0\%$.

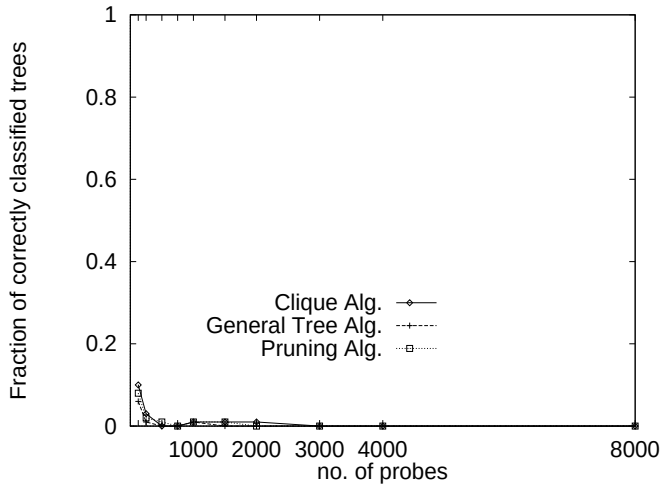

Figure 9: $\varepsilon = 5.0\%$.

beyond 2%.

We now compare the relative performance of the algorithms. We focus on $\varepsilon \leq 1\%$; for this range reasonable accuracy can be obtained for sufficiently large n. We find that the general algorithm GLT has almost uniformly better performance in this region. We believe that this is because GLT makes a more direct inference on non-binary groups than the other algorithms. The accuracy of the BLT Pruning algorithm is close to that of GLT. However, the computational complexity is less than GLT in several ways. $\widehat{A}$ is given explicitly for binary groups, but generally requires numerical root finding for GLT. In GLT the threshold ε is applied throughout the algorithm. In BLT Pruning it is applied only at the end; this will facilitate adaptive selection of ε. BLT Clique has the worst performance. We believe this is because it has a stricter condition for grouping nodes that involves checking all the possible pairs among the candidate nodes; failure of a single test amongst many can result in exclusion from a group.

5.2 Comparison of ML Classifier and Grouping

For identification of general trees, the ML classifier has the apparent advantage over the grouping algorithms that there is no explicit parameter ε that demarcates the minimum loss rates at which may be classified. However, the ML classifier remains subject to the effects of sampling noise when identifying links with low loss probabilities. This is because the likelihood function $\mathcal{L}(\tau, \alpha_\tau, \gamma_\tau)$ depends only weakly on whether links with low rates are present or absent from the topology τ. Sampling variation due to using finitely many probes can swamp such dependence.

In Figure 10 we compare the performance of the ML classifier with the BLTP Classifier. This was performed on 5-receiver trees with link loss rates chosen uniformly between 0% and 10%. We display the probability that links with loss rates greater than or equal to a given level ε are misgrouped. In this context, to correctly group a link means to place it as an ancestor of the same set of receivers in the inferred tree as in the actual tree. The figure shows that the accuracy of the ML and BLTP are very similar across a range of link loss rates.

The ML classifier was implemented as an exhaustive search on the set of trees. The rapid growth in the number of such trees makes this approach computationally unattractive; we believe that more sophisticated optimization techniques may reduce the computational cost of the ML Classifier.

6 Conclusions

We have described and evaluated two classes of algorithm for the identification of logical multicast trees. Grouping algorithms use an estimate of loss on the common path to a set of receivers, and select as siblings that set of nodes for which the loss is greatest. In this class, three algorithms

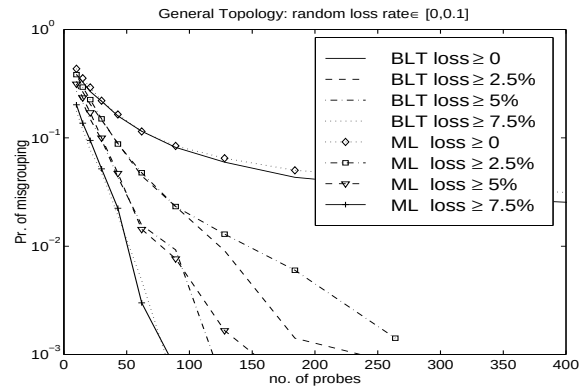

Figure 10: Comparison of ML and BLT classifier in 5-leaf tree: probability to misclassify links with give loss rate ε or greater.

were examined; Binary Loss Tree Clique, Binary Loss Tree Pruning and General Loss Tree. The last two were found to be the most accurate; BLT Pruning is the least complex in implementation and execution.

The Maximum Likelihood classifier uses the property that the link loss estimators are Maximum Likelihood Estimators; the topology with the highest likelihood for the observed data is selected as the estimator. Accuracy was close to that of the grouping algorithms, but execution times were larger due to the implementation as an exhaustive search through the space of possible trees. More sophisticated search techniques may reduce this, however, there would be no gain in accuracy over the grouping algorithms. In summary, we believe the BLT Pruning algorithms offers the best combination of accuracy and performance. Further work to be reported in an extended version of this paper includes the analytic estimation of failure probabilities.

References

[1] R. Cáceres, N.G. Duffield, J. Horowitz, D. Towsley, “Multicast-based inference of network-internal loss characteristics”, *IEEE Trans. Information Theory*, to appear, 1999.

[2] R. Cáceres, N.G. Duffield, J. Horowitz, D. Towsley, T. Bu, “ Multicast-Based Inference of Network-Internal Characteristics: Accuracy of Packet Loss Estimation”, Proceedings IEEE Infocom’99, New York, (1999).

[3] R. Jennrich, “Asymptotic properties of nonlinear least-squares estimators”, *Ann. Math. Stat* 40:633-643, 1969.

[4] `ns` – Network Simulator. For more information see `http://www-mash.cs.berkeley.edu/ns/ns.html`

[5] S. Ratnasamy & S. McCanne, “Inference of Multicast Routing Tree Topologies and Bottleneck Bandwidths using End-to-end Measurements”, Proceedings IEEE Infocom’99, New York, (1999), to appear.

[6] M.J. Schervish, “Theory of Statistics”, Springer, New York, 1995.