NETWORK TOPOLOGY DISCOVERY USING FINITE MIXTURE MODELS

Meng-Fu Shih   *Alfred O. Hero III*

Department of EECS
University of Michigan
Ann Arbor, MI 48109-2222, U.S.A.
mfshih@umich.edu, hero@eecs.umich.edu

ABSTRACT

In this article we propose a network topology estimation strategy using unicast end-to-end packet pair delay measurements that is based on mixture models for the delay covariances. An unsupervised learning algorithms is applied to estimate the number of mixture components and delay covariances. The leaf pairs are clustered by a MAP criterion and passed to a hierarchical topology construction algorithm to rebuild the tree. Results from an `ns` simulation show that our algorithm can identify a network tree with 8 leaf nodes.

1. INTRODUCTION

It is difficult to collect parameters and measurements with direct observations in today's unregulated and heterogeneous Internet environment. Statistical inference from end-to-end measurements, called network tomography, has thus become an important alternative approach. One of the problems of interest is to discover the network topology. The topology of the Internet is constantly changing due to devices going online and offline, and the corresponding routing table updates. Tools such as `traceroute` are usually used to obtain the topology information. However, they rely on the cooperation of internal routers, which has become less attractive as the traffic load and size of the Internet grow. Duffield *et al.* proposed an inference algorithm for multicast topologies [1]. Castro *et al.* investigated the case in unicast network and developed a method of topology discovery using sandwich probe measurements [3].

Here we propose to use packet pair delay measurements. The packet pair probing scheme was first used in network tomography by Coates *et al.* for estimating link loss rates [5]. Based on the delay covariances it is possible to hierarchically cluster the leaf pairs and reconstruct the logical tree network. Rather than attempt to directly maximize the likelihood over the set of possible topologies, we approximate the distribution of the averaged covariance estimates as a Gaussian mixture and find the optimal model which best fits the data using well-developed unsupervised learning algorithms. We cluster the leaf pairs according to a maximum aposteriori (MAP) criterion and hierarchically reconstruct the logical tree. This approach applies to packet pairs as well as other probing methods. An `ns` [4] simulation is implemented and the results demonstrate that our algorithm converges rapidly to the correct topology.

2. PROBLEM FORMULATION

2.1. Framework

A *logical tree* T is defined by two sets : $\mathbf{V}$, the set of nodes, and $\mathbf{E}$, the set of directed links. $\mathbf{V}$ can be expressed as the union $\mathbf{V} = \{0\} \cup \mathbf{V}_i \cup \mathbf{V}_r$, where the root node is defined as node 0, $\mathbf{V}_i$ denotes the set of internal nodes and $\mathbf{V}_r$ is the collection of leaf nodes. The root node is assumed to have one child node, and all the internal nodes must have at least two descendants.

Each logical tree derives a *logical grouping tree* which depicts the unique hierarchical clustering of all the pairs of leaf nodes according to their nearest common ancestors. More specifically, the logical grouping tree is formed by first relabelling each internal node of the original logical tree with the set of leaf pairs which has the internal node as their nearest common ancestor, and then deleting the root and leaf nodes, as well as all the links associated with them. Let $\mathcal{T}_R$ be the set of logical trees which have R leaf nodes, and $\mathcal{T}_R^g$ be the set of valid logical grouping trees with each corresponding to a R-leaf logical tree. It is obvious from the definition that there is a bijective mapping between $\mathcal{T}_R$ and $\mathcal{T}_R^g$.

To determine the logical grouping tree of the network, we need the concept of *Metric-Induced Network Topology* (MINT) introduced by Bestavros *et al.* [2], in which a metric function is defined for the set of paths in a logical tree. Coates *et al.* focused on a special case of *similarity metric* which is defined on the path from the root node to the nearest ancestor of a pair of leaf nodes [3]. Let p_i denote the path from the root to node i, $i \in \mathbf{V} \setminus \{0\}$, which

is associated with a metric γ_{p_i}. Let $a(i,j)$ be the nearest common ancestor node of the leaf pair (i,j). A similarity metric $\gamma_{i,j}$ is defined by $\gamma_{i,j} = \gamma_{p_{a(i,j)}}$. The (similarity) metrics obey the following two properties [2, 3]. (P1)**Monotonicity**: $\gamma_{p_i} \leq (\geq)\gamma_{p_j}$ if p_i is a proper subpath of p_j, $i, j \in \mathbf{V} \setminus \{0\}$ (P2)**Consistency**: $\gamma_{i,j} = \gamma_{k,l}$ if $a(i,j) = a(k,l)$ for distinct leaf nodes i, j, k, l. The two properties guarantee a logical grouping tree can be formed using the set of similarity metrics. All the leaf node pairs with the same metric value can be grouped into the same node of the grouping tree using the consistency property. The hierarchical relationship among the grouping tree nodes can be determined by rank ordering of the metric magnitudes using the monotonicity property.

2.2. Packet Pair Probing Scheme

The similarity metrics used for grouping tree construction are mainly determined by the end-to-end probing scheme and the extracted statistics of the measured probes. Here we propose to use a pair of closely-spaced time-stamped packets, called a probe pair, which are both sent from the root node but routed to two different leaf nodes. In order to form a similarity metric from the probe pair, we impose the following assumptions. (A1)**Delay Consistency**: the delays of the two packets in a probe pair on their shared path are identical with probability 1. (A2)**Spatial Independence**: the packet delays are independent over different links, either for the same packet or among different packets (A3)**Temporal Independence and Stationarity**: Delays of different packets. on the same link are identically and independently distributed (i.i.d.). Futhermore, the network topology and routing tables have to be fixed during the probing session.

The metric we propose to use is the covariance of the end-to-end probe pair delays. Each pair of leaf nodes corresponds to a binary tree rooted at node 0, called a probe tree. We label the probe trees from 1 to $\binom{R}{2}$, where $R = |\mathbf{V}_r|$. Assume that probe tree t has leaf nodes i and j, denoted by $(i,j) \in t$. Let $\mathbf{Y}_n^t = \{Y_{1,n}^t, Y_{2,n}^t\}$ be the end-to-end delays of the nth probe pair sent along probe tree t. The probe pair delays $X_{0,n}^t$, $X_{1,n}^t$ and $X_{2,n}^t$ denote delays on the shared path and the branch toward the left and right leaf nodes, respectively. They are mutually independent due to the spatial independence assumption. Note that here $a(i,j)$ is the branching node of the paths and $X_{0,n}^t$ is the delay from the root node to $a(i,j)$. Then we have $Cov(Y_{1,n}^t, Y_{2,n}^t) = Cov(X_{0,n}^t + X_{1,n}^t, X_{0,n}^t + X_{2,n}^t) = Var(X_{0,n}^t)$, which depends only on the nearest common ancestor $a(i,j)$, and hence satisfies the consistency property. The (increasing) monotonicity is guaranteed by the positivity of the variances and the spatial independence assumption. The end-to-end delay of a packet is measured at the receiver by the difference between the packet time stamp and the system time upon reception. Unlike delay tomography, the probe source,

i.e. the root node, and the receivers, i.e. the leaf nodes, do not have to sychronize their system clocks because a constant shift in delay measurement doesn't affect the covariance estimates [6].

3. FINITE MIXTURE CLUSTERING ALGORITHM

To hierarchically cluster the leaf pairs, we have to establish a probability model of data samples as follows. Consider a single probe tree t. We compute an unbiased covariance estimate for every set of N_1 collected sample probe pair delays

$$s_{t,l_1}^2 = \frac{\sum_{n=(l_1-1)N_1+1}^{l_1 N_1} \left(y_{1,n}^t - \bar{y}_{1,l_1}^t \right)\left(y_{2,n}^t - \bar{y}_{2,l_1}^t \right)}{N_1-1}, \text{ where}$$

$\bar{y}_{q,l_1}^t = \frac{1}{N_1} \sum_{n=(l_1-1)N_1+1}^{l_1 N_1} y_{q,n}^{(i,j)}$, $q = 1,2$. For every set of N_2 covariance estimates, we calculate the average

$$\bar{s}_{(i,j),l_2}^2 = \frac{\sum_{l_1=(l_2-1)N_2+1}^{l_2 N_2} s_{t,l_1}^2}{N_2} \stackrel{N_2 \to \infty}{\sim} N\left(Var(X_0^t), \frac{\sigma_t^2}{N_2} \right),$$

where in the last term the central limit theorm is involved on the sum of i.i.d. samples $\{s_{t,l_1}^2\}_{l_1}$. With a large N_2, the averaged covariance estimates are approximately Gaussian distributed with mean $Var(X_0^t)$ and variance σ_t^2/N_2, where σ_t^2 is the variance of s_{t,l_1}^2. Finally a total of N_3 such covariance averages are collected at probe tree t, i.e., $\mathbf{S}_t = \{\bar{s}_{t,l_2}^2\}_{l_2=1}^{N_3}$.

Let $\mathbf{T}_c$ denote the set of probe trees which share the same branching node c. The Gaussian distributions of all $\mathbf{S}_t$, $t \in \mathbf{T}_c$ have the same mean but not the same variance. However, as N_2 increases the differences among those variances decrease linearly. When N_2 is large enough, the variance differences are negligible and the $\mathbf{S}_t$'s approximately admit the same distribution $N\left(\mu_c, \sigma_c^2\right)$, where $\mu_c = Var(X_0^t)$ for each $t \in \mathbf{T}_c$. If we consider the whole data pool $\mathbf{S} = \{\mathbf{S}_t\}$, the samples can be viewed as i.i.d. realizations of a finite Gaussian mixture

$f(x) = \sum_{c \in \mathbf{V}_i} \frac{|\mathbf{T}_c|}{\binom{R}{2}} N\left(x; \mu_c, \sigma_c^2\right)$. Note that this is an *unconstrained* model for $\mathbf{S}$ since we do not impose any topological constraints on the network. Each mixture component corresponds to an internal node of the logical tree. The number of the components is upper bounded by $R-1$, and lower bounded by 1.

4. MIXTURE MODEL ESTIMATION AND TOPOLOGY CONSTRUCTION

Finite mixture models are widely used in data clustering and pattern recognition problems [7]. Estimation of parameters in a mixture model includes model order selection, i.e., determination of the number of mixture components, and inference of the parameters. Figueiredo and Jain [7] proposed an EM implementation with a minimum-message-length(MML) type of penalty on model over-estimation. The algorithm starts with k_{max} components, where k_{max} is se-

lected such that each true component has a good possibility to be well represented at random initialization. A mixture component is either kept or removed at each iteration according to a criterion derived from the MML penalty. After convergence, the component with the least mixture probability is removed and the algorithm is re-initialized with the remained components. This process is repeated until the lowest number of components is reached. The estimate which converges to the highest penalized likelihood gives the MML covariance estimates.

Suppose the set of D convergent estimates is $\{\hat{k}_d, \hat{\Theta}_d = \{\hat{\alpha}_{dm}, \hat{\mu}_{dm}, \hat{\sigma}^2_{dm}\}^{\hat{k}_d}_{m=1}\}^D_{d=1}$, where $\hat{k}_d$ denotes the model order, $\hat{\Theta}_d$ is the set of parameters, and each estimate d corresponds to a Gaussian mixture $\hat{f}_d(x) = \sum^{\hat{k}_d}_{m=1} \hat{\alpha}_{dm} N\left(x; \hat{\mu}_{dm}, \hat{\sigma}^2_{dm}\right)$. The parameter sets of the $\hat{k}_d$ components in $\hat{\Theta}_d$ are sorted in non-increasing order of the means $\{\hat{\mu}_{dm}\}$, and the D estimates are also sorted in decreasing order of their optimality. To classify $\mathbf{S}_t$ into one of the components, we apply a MAP rule

$$\hat{a}_d(t) = \arg\max_{m=1,\ldots,\hat{k}_d} \log P(m|\mathbf{S}_t)$$
$$= \arg\max_{m=1,\ldots,\hat{k}_d} \left\{\log \hat{\alpha}_{dm} + \sum^{N_3}_{l=1}\left[\log N\left(\bar{s}^2_{t,l}; \hat{\mu}_{dm}, \hat{\sigma}^2_{dm}\right) - \log \hat{f}_d(\bar{s}^2_{t,l})\right]\right\}.$$

We need the following definitions to hierarchically reconstruct a logical tree. A *supernode* $v(i)$ is a virtual node comprises the subtree rooted at internal node i. A leaf node j is included in $v(i)$ if it is a leaf node of the corresponding subtree, denoted by $j \prec v(i)$. Supernodes $v(i_1)$ and $v(i_2)$ are *connected through mixture component m* if there exists a $j_1 \prec v(i_1)$ and a $j_2 \prec v(i_2)$ such that $\hat{a}_d(t) = m$ for $(j_1, j_2) \in t$, denoted by $v(i_1) \overset{m}{\longleftrightarrow} v(i_2)$. Similarly, we define $j \overset{m}{\longleftrightarrow} v(i)$ for j being a leaf node not included in $v(i)$. Supernodes $v(i_1)$ and $v(i_2)$ are *fully connected through mixture component m* if for every $j_1 \prec v(i_1)$ and $j_2 \prec v(i_2)$, $\hat{a}_d(t) = m$ for $(j_1, j_2) \in t$, denoted by $v(i_1) \overset{m}{\Longleftrightarrow} v(i_2)$. Similarly, we define $j \overset{m}{\Longleftrightarrow} v(i)$ for j being a leaf node not included in $v(i)$. To successfully include the leaf pairs associated with a new component m into the tree we have to verify whether all the existing supernodes and the new leaf nodes which are connected through m to some other supernodes or new leaf nodes are also fully connected to each other. The complete topology construction algorithm is illustrated in Table 1. It starts from the optimal Gaussian mixture estimate. If it doesn't generate a valid logical tree, then it goes to the next optimal estimate, until a valid topology is found. This is a sub-optimal algorithm since it does not seek optimal constrained parameter values when a mixture model estimate and its corresponding leaf node clustering do not produce a valid tree, but it guarantees in the worst case a logic tree associated with a one-component mixture will be generated, in which every leaf node directly connects to the root node's descendant.
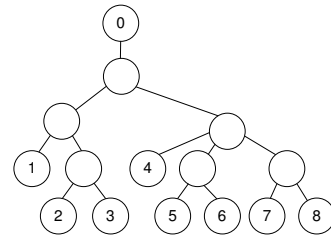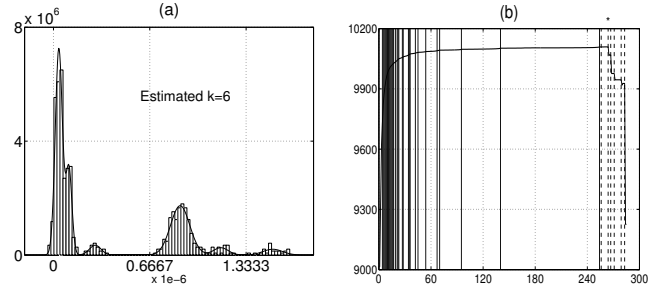

Fig. 1. `ns` simulation network.

Fig. 2. `ns` simulation results.

5. COMPUTER SIMULATION

We conducted an `ns` simulation on the network shown in Fig. 1. Each probe is a 40-byte UDP packet. Probe pairs are sent from node 0 to traverse each probe tree in a round-robin fashion with interprobing time $500\mu s$. There are a total of 72 constant-bit-rate UDP background streams with random departure times, and 144 TCP background flows with Pareto On-Off distribution. We collect $N_1 \times N_2 \times N_3 = 40 \times 40 \times 25 = 40000$ packet pairs for each probe tree. They are a lot of packets and are only presented here for proof of concept. We start the algorithm with $k_{max} = 84$ components. Fig. 2 (a) shows the optimal Gaussian mixture estimate and (b) shows the convergence curve, where the solid lines indicate the iterations at which a component is automatically removed and the dashed lines show the algorithm converges and the component with the least mixture probability is then removed manually. The asterisk points at the convergent iteration of the optimal estimate. The algorithm converges very fast in less than 300 iterations, and produces the MAP clustering in Table 2, which reconstructs the correct logical tree topology.

6. CONCLUSION AND FUTURE WORKS

In this paper we propose a method of network topology estimation using end-to-end packet pair delays. We adopt packet pair delay covariance as the similarity metric, and establish a Gaussian mixture model for the averaged covariance estimates. By applying an unsupervised learning

$$\text{Input: } \mathbf{V}_r, \left\{ \hat{k}_d, \hat{\boldsymbol{\Theta}}_d, \hat{a}_d \right\}_{d=1}^{D}, \qquad \text{Output: } T = (\mathbf{V}, \mathbf{E})$$

$d \leftarrow 1, Finish \leftarrow 0$
while $((d \leq D) \wedge (Finish = 0))$ do
$\quad \tilde{\mathbf{V}}_r \leftarrow \phi, \mathbf{V}_i \leftarrow \phi, \mathbf{V}_s \leftarrow \phi, \mathbf{E} \leftarrow \phi, m \leftarrow 1, valid \leftarrow 1$
$\quad$ while $((m \leq \hat{k}_d) \wedge (valid))$ do
$\qquad \mathbf{V}_{new} \leftarrow \left\{ j_1 \in \mathbf{V}_r \backslash \tilde{\mathbf{V}}_r : \exists j_2 \in \mathbf{V}_r \backslash \{j_1\}, (j_1, j_2) \in t \quad \hat{a}_d(t) = m \right\}$
$\qquad \mathbf{S} \leftarrow \left\{ v(i) \in \mathbf{V}_s : \left(\exists v(j) \in \mathbf{V}_s \backslash \{v(i)\} \quad v(j) \overset{m}{\longleftrightarrow} v(i) \right) \vee \left(\exists j \in \mathbf{V}_{new} \quad j \overset{m}{\longleftrightarrow} v(i) \right) \right\}$
$\qquad$ if $\left[\exists j_1, j_2 \in \mathbf{V}_{new}, (j_1, j_2) \in t \quad \hat{a}_d(t) \neq m \right] \vee \left[\exists v(i_1), v(i_2) \in \mathbf{S} \quad \sim \left(v(i_1) \overset{m}{\Longleftrightarrow} v(i_2) \right) \right] \vee$
$\qquad \left[\exists j \in \mathbf{V}_{new}, v(i) \in \mathbf{S} \quad \sim \left(j \overset{m}{\Longleftrightarrow} v(i) \right) \right]$ then $valid \leftarrow 0$
$\qquad$ else create a new internal node $w_m, \mathbf{V}_i \leftarrow \mathbf{V}_i \cup \{w_m\}$
$\qquad\quad \mathbf{E} \leftarrow \mathbf{E} \cup \{(w_m, i) : i \in \mathbf{V}_{new}\} \cup \{(w_m, i) : v(i) \in \mathbf{S}\}, \mathbf{V}_s \leftarrow \mathbf{V}_s \cup \{v(w_m)\} \backslash \mathbf{S},$
$\qquad\quad \tilde{\mathbf{V}}_r \leftarrow \tilde{\mathbf{V}}_r \cup \mathbf{V}_{new}, m \leftarrow m + 1$
$\qquad$ end if
$\quad$ end while
$\quad$ if $(valid)$ then
$\qquad \mathbf{V} \leftarrow \{0\} \cup \mathbf{V}_i \cup \mathbf{V}_r, \mathbf{E} \leftarrow \mathbf{E} \cup \left\{ (0, w_{\hat{k}_d}) \right\}, T \leftarrow (\mathbf{V}, \mathbf{E}), Finish \leftarrow 1$
$\quad$ end if
$\quad d \leftarrow d + 1$
end while

Table 1. Topology construction algorithm.

m	$\hat{\mu}_{1m}(\times 10^{-6})$	$\{[i,j] : \hat{a}_1(t) = m, (i,j) \in t\}$
1	1.5	[7,8]
2	1.14	[5,6]
3	0.845	[4,5], [4,6], [4,7], [4,8], [5,7], [5,8], [6,7], [6,8]
4	0.291	[2,3]
5	0.105	[1,3], [1,2]
6	0.042	[1,4], [1,5], [1,6], [1,7], [1,8], [2,4], [2,5], [2,6], [2,7], [2,8], [3,4], [3,5], [3,6], [3,7], [3,8]

Table 2. MAP clustering for the optimal mixture estimate ($d = 1$).

algorithm for finite mixtures, we select the mixture model candidates and estimate their parameters. The pairs of leaf nodes are clustered with a MAP criterion, and used as the input to our topology construction algorithm. `ns` simulation results demonstrate the fast and accurate performance of our methodology.

Future work will focus on developing a *topology constrained* algorithm to incorporate topological restrictions in model selection and estimation. We also would like to implement our mixture model algorithm to other probing methods, such as the mean delays of the sandwich probes in [3]. Finally, we are in the process of collecting real network data to evaluate the performance of our algorithm in a practical environment.

7. REFERENCES

[1] N.G. Duffield, J. Horowitz, F. Lo Presti, and D. Towsley, "Multicast topology inference from measured end-to-end loss," *IEEE Trans. Info. Theory*, vol.48, pp.26-45, Jan. 2002.

[2] A. Bestavros, J. Byers, K. Harfoush, "Inference and labeliing of metric-induced network topologies," *IEEE Infocom 2002*, New York, NY, June 2002.

[3] R. Castro, M. Coates, G. Liang, R. Nowak, B. Yu, "Network tomography: recent developments," to appear in *Statistical Science*.

[4] *ns* - Network Simulator. See `http://www-mash.cs.berkeley.edu/ns/ns.html`.

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

[6] A Pásztor and D. Veitch, "PC based precision timing without GPS," *ACM SIGMETRICS*, A, Marina Del Rey, Los Angeles, June 2002

[7] Figueiredo M., Jain, A.K.,"Unsupervised learning of finite mixture models," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.24, pp.381–396, Mar. 2002.