Simple Network Performance Tomography

Nick Duffield
AT&T Labs—Research
180 Park Avenue
Florham Park, NJ 07932, USA
duffield@research.att.com

ABSTRACT

In network performance tomography, characteristics of the network interior are inferred by correlating end-to-end measurements. In much previous work, the presence of correlations must be arranged at the packet level, e.g., using multicast probes or unicast emulations of them. This carries costs in deployment and limits coverage. However, it is difficult to determine performance characteristics without correlations. Some recent work has had success in reaching a lesser goal—identifying the lossiest network links—using only uncorrelated end-to-end measurements. In this paper we abstract the required properties of network performance, and show that they are independent of the particular inference algorithm used. This observation allows us to design a quick and simple inference algorithm that identifies the worst performing link in a badly performing subnetwork, with high likelihood when bad links are uncommon. We give several examples of perforance models and that exhibit the required properties. The performance of the algorithm is analyzed explicitly.

Categories and Subject Descriptors

C.2.3 [**Computer–Communications Networks**]: Network Operations—*Network monitoring*; C.4 [**Performance of Systems**]; G.3 [**Probability and Statistics**]

General Terms

Measurement, Performance, Theory

Keywords

Inference, Estimation, Performance, Networks, Correlation

1. INTRODUCTION

1.1 Motivation

Network performance tomography is the science of inferring performance characteristics of the network interior by correlating sets of end-to-end measurements. Recently, several methods have been proposed to infer link characteristics (including packet loss and delay), and the underlying network topology. Initial proposals exploited the inherent correlations between copies of a multicast packet seen at different endpoints; see [1] for a review. Subsequent work emulated this approach using clusters of diversely addressed unicast packets; see [3].

A key advantage of tomographic methods is that they require no participation from network elements other than the usual forwarding of packets. This distinguishes them from well-known tools such as traceroute and ping, that require ICMP responses to function. In some networks, ICMP response has been restricted by administrators, presumably to prevent probing from external sources. Another feature of tomography is that probing and the recovery of probe data may be embedded within transport protocols, thus co-opting suitably enabled hosts to form impromptu measurement infrastructures; see [2] and [4].

Several challenges exist in bringing these methods to widespread fruition. Multicast is not widely deployed. For methods based on unicast probing, there are development and administrative costs associated with deploying appropriate probing and data collection software. This motivates reducing such costs by developing inference methods that can work with readily available end-to-end measurements.

Recent work in this direction has been performed by Padmanabhan, Qiu and Wang [6]. They propose to use statistics gathered from (near a) web server about loss on the end-to-end paths from the server to the client. The loss rates are determined by observing TCP retransmissions. In distinction with the work mentioned above, this approach does not assume or attempt to exploit any correlations in the network experience of packets destined for different clients. Packets are only assumed to have the same probability of being lost on traversal of a given link. The set of server-to-client paths forms a tree. The aim is to use the end-to-end data to infer the loss rates on the logical links joining the branch points of the tree, at least with sufficient accuracy to identify the lossiest links.

A notable feature of the model considered in [6] is that its parameters (the loss rates on the logical links) are *not statistically identifiable* from the data (the server-to-client loss rates), meaning that there exist different sets of parameters that give rise to the same statistical distribution of data. Consider the two leaf tree of Figure 1(left), where the transmission rate on the link terminating at node k is ϕ_k (thus $1-\phi_k$ is the corresponding loss rate). The transmission probabilities from the server at node 0 to the clients at nodes 2 and 3 are the products $p_2 = \phi_1\phi_2$ and $p_3 = \phi_1\phi_3$ respectively. (The transmission probability for a path is the product of the transmission probabilities for its links). Thus the server-to-client transmission probabilities are the same when the link probabilities are adjusted as in Figure 1(right), for any multiplicative

IMC'03, October 27–29, 2003, Miami Beach, Florida, USA.

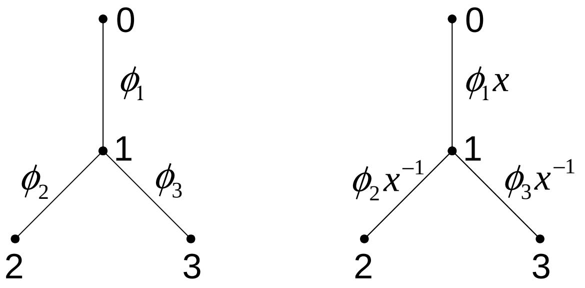

Figure 1: Absence of Identifiability from Uncorrelated Measurements: trees have different link transmission rates but identical end-to-end transmission rates.

factor x between $\max\{\phi_2, \phi_3\}$ and $1/\phi_1$. (This condition yields link probabilities less than or equal to 1). Another way to view this is that the model has three independent parameters (the link transmission probabilities ϕ_1, ϕ_2 and ϕ_3) while the data depends on only two quantities (the end-to-end transmission probabilities p_2 and p_3). We cannot uniquely determine the ϕ_i from the data.

Despite the fact that the model is not statistically identifiable, some methods proposed in [6] are quite successful in identifying the lossiest links, both in a class of model networks (particularly when lossy links are rare), and in real topologies where the lossiest links tend to be at the clients. (Although the most accurate methods are computationally very intensive). If we can understand the structural reasons why this is possible, we can use this understanding to develop classes of quick and simple estimators for the worst performing links for a range of performance characteristics. This will be the focus of the present paper.

1.2 Contribution and Summary

The key to understanding how the worst links can be found in practice rests on a structural assumption about the nature of link performance. Suppose we can classify links as "good" or "bad", with performance measures sufficiently separated. Then the performance experienced along a network path will be bad only if one of its constituent links is bad. The separation between good and bad performance means that a bad path can not arise through a combination of two or more "partially bad" links.

Moreover, if link badness is sufficiently uncommon, then when bad performance is observed on two intersecting end-to-end paths, the overwhelmingly most likely way for this to have occurred is if the common path portion contains a bad link. In this way, we can identity such bad links with high probability.

The contributions of this paper are as follows:

(a) using a set of simple point-to-point performance measures, we show that the worst performing links in a tree can, with high likelihood, be successfully identified, even though the underlying statistical model is not identifiable.

(b) we describe examples of such performance measures, and show that the network models of [6] fall (or almost fall) into the class of models that admit this approach.

(c) we describe a quick and simple algorithm—the smallest consistent failure set (SCFS) rule—for identifying, with high likelihood, the worst performing links. In doing so, we take advantage of the fact that the ability to perform such identification stems from structural properties of the underlying statistical model, rather than being dependent on the choice of inference algorithm.

(d) the SCFS algorithm is sufficiently simple that we can analyze its performance *analytically* in model networks, rather than requiring model simulations. Evaluation of the performance under perturbations from the model assumptions are currently under study: we hope to report these results in an extended paper.

The remainder of the paper is as follows. Section 2 defines the notion of separability for performance measures, and argues that it is satisfied both by performance models treated in the literature, and by some classes of experimental network performance measurements. The simple tomographic inference rule is defined to identify likely bad performing links in tree networks. Section 3 analyzes the performance of the inference rule under a network performance model under which bad links are distributed at random. We discuss the scaling behavior of the performance for deep trees, and draw some comparisons with the the findings of [6]. We conclude in Section 4 with a discussion of the work ongoing to bring this research to completion. The proofs of all theorems are omitted.

2. NETWORK PERFORMANCE MODEL

We start in Section 2.1 by recording our terminology for trees. Section 2.2 formalizes the separation of links into good and bad subsets, and Section 2.3 describes some examples. Section 2.4 describes the inference rule.

2.1 Tree Model and Terminology

The network topology is represented as a directed tree $\mathcal{T} = (V, L)$ comprising a set of nodes V joined by links in L. A packet source (e.g. a server) is located at the root node 0, while a set of destinations (e.g. clients) are located at the leaf nodes R. The interior nodes of the tree represent the branch points of the routing tree from the source to the destinations, and the links L are the logical links that link these branch points. We say node j is the parent of node k if $(j, k) \in L$, and write $j = f(k)$. Other ancestors of k are defined by $f^n(k) = f(f^{n-1}(k))$ with $f^1 = f$. We write $j \prec k$ if j is a descendant of k, i.e., if $k = f^m(j)$ for some m. The set of children of node k is $d(k) = \{j \in V : (k, j) \in L\}$. We sometimes write $U = V \setminus \{0\}$. We will often refer to the link terminating at node k as "link k". The root node 0 is assumed to have a single child, denoted by 1. If, not the tree can be disjoined into subtrees with this property.

2.2 Link Performance and Separability

Our performance model is as follows. During some measurement period, the source dispatches a set of packets to each destination. On traversing link k, each packet is subject to a performance degradation (e.g. loss or delay) according to a distribution specified by a parameter ϕ_k. If the source-destination path comprises links $k_1, \dots, k_m$, the performance degradation along the path follows a composite distribution described by the parameters $\phi = \phi_{k_1}, \dots, \phi_{k_m}$.

Let ψ be the expected value of a statistic computed from link or path performance distributions; write $\psi(\phi_k)$ and $\psi(\phi_{k_1}, \dots, \phi_{k_m})$ respectively. For each link or path, we partition the set of possible ψ values into two subsets that we we call "good" and "bad". Likewise, we call the link, or its parameter ϕ_k bad, iff the expected statistics $\psi(\phi_k)$ is bad, and we call the path $k_1, \dots, k_m$ bad iff $\psi(\phi_{k_1}, \dots, \phi_{k_m})$ is bad. The key property that captures the ability to detect the presence of badly performing links from end to end measurements is as follows:

- The partitions are called *separable* when a path is bad if and only if at least one of its constituent links is bad.

- The partitions are called *weakly separable* when a path being bad implies at least one of its constituent links is bad.

We use the word "separable" because if the good and bad link parameter sets are too close together, it will not be possible to distinguish between them in the composite path measurements. Weak separability means that paths with all good links are correctly identified, but some bad links may go undetected.

We can always arrange for weak separability by defining the set of good paths to be those with expected statistic ψ in the set $\Psi_{\text{good}} = \{\psi(\phi_{k_1}, \ldots, \phi_{k_m}) \mid \text{all } \phi_i \text{ good}\}$. The extent to which this is useful then depends on the false positive rate for good links, i.e. the probability that a path with $\psi \in \Psi_{\text{good}}$ does, in fact, contain a bad link. We now illustrate this framework with some examples.

2.3 Examples of Separable Performance

Connectivity. If a link or a path is good, it transmits all packets; if bad, it transmits none. Thus the path is bad iff at least one link of the path is bad.

High-Low Loss Model. Packets traverse link k independently with probability ϕ_k. The ranges of transmission probabilities for good and bad links are separated. Good links k have transmission rate $\phi_k > x$; bad links have transmission rate $\psi = \phi_k < y$, with $y < x^\ell$ where ℓ is the depth of the tree (i.e. maximum hop count from root to leaf). For a path traversing links $1, \ldots, m$ we take $\psi = \prod_{i=1}^{m} \phi_{k_i}$, i.e., the path transmission rate.

The minimum transmission rate on a path containing no bad link is x^ℓ, while the maximum transmission rate on other paths is y. Picking any z between y and x^ℓ, we call a path good if its transmission rate exceeds z, and bad otherwise. Then a path is bad if and only if it contains at least one bad link.

In the model LM_1 of [6], good links have loss rates $1 - \phi_k$ uniformly distributed between 0% and 1%; bad links have loss rates uniformly distributed between 5% and 10%. Taking the threshold between good and bad path transmission rates as 0.95, this model is separable if the tree depth does not exceed 5.

In the model LM_2 the bad links have loss uniformly distributed between 1% and 100%. In this case the ranges of transmission probabilities for good and bad links are contiguous. Nevertheless, if we chose 0.99^ℓ to be the threshold transmission rate separating good and bad paths of ℓ hops, then the partition is weakly separable, i.e., all paths containing only good links are designated as good. The chance for a path containing at least one bad link to be designated good is no more than $1 - 0.99^{\ell-1}$, e.g., about a 4% chance for $\ell = 5$. Thus the paths containing bad links can still be identified with high probability.

General Additive High-Low Model. The above model type generalizes to a class of models in which link performance is independent, and the statistic ϕ is any characteristic that is additive over links: $\psi(\phi_{k_1}, \ldots, \phi_{k_m}) = \psi(\phi_{k_1}) + \cdots + \psi(\phi_{k_m})$. The loss model above falls into this class if we take as ϕ, instead of the transmission probability, its logarithm. Other examples of additive statistics are delay mean and variance.

Delay Spike Model. Measurement of network round trip times (RTT) have shown the presence of "delay spikes", namely intervals of highly elevated round trip times; see [8]. To get a rough idea of what is observed, in one data set, delay spikes of median delay 16.9 standard deviations above the mean RTT had median duration $d_s = 150$ms. The spike episodes were found to be well modeled by a Poisson process, with typical mean interarrival time τ_s of the order of 10s to a few hundreds of seconds.

```
input: Topology $\mathcal{T}$; End-to-end measurements $\{X_k\}_{k \in R}$;
$Y_0 = 1$;
$W = \emptyset$;
recurse(1);
output: $W$;

subroutine recurse($k$) {
    if($k \in R$) {$Y_k = X_k$};
    else {
        $Y_k = \max_{j \in d(k)} Y_j$;
    foreach($j \in d(k)$){
        if (($Y_j == 0$) & ($Y_k == 1$)){
            $W = W \cup \{j\}$;
        }
    }
}
```

Figure 2: Recursive Implementation of SCFS rule. Recall 1 denotes the single child node of the root node 0.

We assume that for a given application, delay spike beyond to a certain level z are not tolerable. Paths with (some statistic of) the spike delay greater than z will be designated as bad.

We model of the occurrence of delay spikes as follows. Packets are potentially subject to delay spikes on each link, although links may not exhibit any delay spikes at all. We assume

(A1) Delay spikes are short enough that a given packet will likely encounter only one spike on a network path.

(A2) Spikes on a given link are assumed frequent enough that at least one packet of the set destined to a given receiver will encounter a delay spike on a link that exhibits them.

Under these assumptions, we chose ψ as the some quantile (e.g. the maximum) of the delay spike distribution. If a path measurement yields $\psi > z$ (the threshold describes above), then according to assumption (A1), a delay spike of that size was present on at least one of the links of the path: we will call such links bad. By Assumption (A2) this delay spike should be present on all the paths through the bad links. Hence, the division into good and bad links and paths is expected to be separable.

We show that the delay spike processes observed in [8] are consistent with assumptions (A1) and (A2). We assume the numbers d_s and τ_s characterize the delay spikes of a single link. (In our model, the delays are one-way, rather than RTTs).

First, (A1). The probability q for a packet to encounter more than one delay spike on a path comprising L hops is about $1 - (1 - d_s/\tau_s)^\ell - \ell(1 - d_s/\tau_s)^{\ell-1} d_s/\tau_s$. This probability increases with path length. Taking $\ell = 30$, larger than most paths today (see [5]) and $d_s = 150$ms, then q ranges from 0.02 for $\tau_s = 20$s, down to 3×10^{-5} for $\tau_s = 600$s. The chance of encountering more than one spike is very small for these values.

Now, (A2). Consider measurement over an interval of duration T with probe packets at frequency r. The average number of spikes encountered by the probes is about $n = d_s r T / \tau_s$, while the probability that at least one probe encounters at least one spike is about $p = (1 - (1 - d_s/\tau_s)^{rT}$. Consider a 10KByte/s probe stream comprising one 200 byte packet every 20ms, equivalent to a compressed audio transfer; thus $r = 50$. Assuming a measurement period of $T = 600$s, then $(n, p) = (225, 1)$ for $\tau_s = 20$s, and $(7.5, 0.9995)$ when $\tau_s = 600$s. Hence Assumption (A2) is reasonable here.

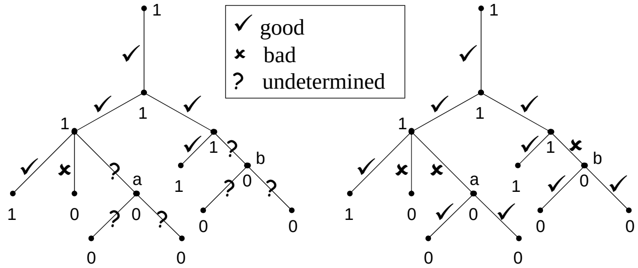

Figure 3: OPERATION OF THE SCFS RULE: Y_k shown at each node k.. Left: status of link a, b and subtree descended from them are uncertain because paths to these subtrees are bad. Right: SCFS attributes badness to common path (links a and b) and designated remaining nodes as good.

2.4 Smallest Consistent Failure Set Rule

The smallest consistent failure set (SCFS) rule designates as bad only those links nearest the root that are consistent with the observed pattern of bad paths.

Define an indicator variable Z_k to be 1 if link k is good, and 0 if it is bad; for the root node 0 set $Z_0 = 1$ by convention. For each path from the root 0 to the node k, let $X_k = 1$ if the path is good, and 0 if it is bad. Under the separability assumption, we can write $X_k = \prod_{j \succeq k} Z_j$, i.e. the product of the link indicators X_j for ancestors j of k (including k itself).

Let R_k denote the set of leaf nodes in R that are descended from k. Write $Y_k = \max_{j \in R_k} X_j$ for $k \in U$ and set $Y_0 = 1$ by convention. $Y_k = 1$ iff at least one source-destination path routed through k is good. Clearly, if $Y_k = 1$, then the path segment from 0 to k is composed entirely of good links. If $Y_k = 0$ but $Y_{f(k)} = 1$ we call the subtree rooted at k a *maximal bad subtree*. A cautious approach would be to declare as bad link k and all links in the subtree. In practice, this is probably not very useful, due to the cost of inspecting all the links for badness.

The SCFS rule takes the other extreme, by designating as bad, one link in the subtree that is very likely to be amongst the set of bad links, namely the link k. For suppose on the other hand k is not bad. Then all the path segments from k to the destinations in $R(k)$ must be bad. This is of course, possible, which is why we cannot pin down the bad links with certainty. However, if the rate of occurrence of bad links is sufficiently small, then, as we shall see, it is far more likely that the link k is bad. Anticipating this, we form an inference rule which designates link k to be bad and all its descendant links good. Put another way, we estimate Z_k by $\widehat{Z}_k = 0$, while for all links j with $j \preceq k$ we estimate Z_j by $\widehat{Z}_j = 1$.

Smallest Common Failure Set (SCFS) Rule

- Inputs: Tree $\mathcal{T}$, End-to-end measurements $\{X_k\}_{k \in R}$;
- $W' = \{k \in U \mid \max_{j \in R_k} X_k = 0\}$;
- $W = \{k \in W' \mid f(k) \notin W'\}$;
- Output: W;

In algorithmic terms, the set W can be calculated on a recursion through the tree; see Figure 2. The set W contains those links k for which node k is the root of a maximal bad subtree.

The action of the SCFS rule is illustrated in Figure 3. On the left, given the data $\{X_k\}_{k \in R}$, we display the values Y_k for each node k. Note $Y_k = X_k$ for leaf nodes k. We can only infer with certainty that the paths from the root 0 to leaf nodes k with $Y_k = 1$ nodes are good. The status of the links a and b, and the subtrees descended from them, is uncertain. The right figure shows the action of the SCFS rule. The links a and b are designated as bad, while all links in subtrees descended from these links are designated good.

A special case is when k is a bad leaf link whose parent has a good path routed through it. In this case, k can ambiguously be declared bad since the maximal bad subtree descended through k has one member, namely, k itself.

3. ANALYSIS OF SCFS PERFORMANCE

The SCFS rule codifies a parsimonious approach that might well be taken without the benefit of analysis, namely, to attribute a pattern to bad paths as being due to badness in the smallest possible set of interior links. This section analyzes the performance of the SCFS algorithm under a statistical model for the distribution of bad links in the network. Following [6] we assume that links are good or bad independently. Thus the Z_k are independent random variables, and we denote by $\alpha_k = \mathsf{P}[Z_k]$ the probability that X_k is good. In the next section we analyze the performance of the inference rule under this statistical model. For compactness we will use the notation $\overline{\alpha} = 1 - \alpha$ in what follows.

3.1 False Positives in Identifying Bad Links

We now provide a statistical justification of the inference rule in terms of its performance. Recall that the inference rule makes the estimate $\widehat{Z}_k = 0$ when node k routes no good paths to its descendant leaves ($Y_k = 0$), but a path through its parent is known to be good ($Y_{f(k)} = 1$). The false positive rate F_k for this decision is the conditional probability that the link was in fact good, i.e.,

$$F_k = \mathsf{P}[Z_k = 1 | Y_k = 0, Y_{f(k)} = 1] \tag{1}$$

To compute this, note that $Y_{f(k)} = 1 \Rightarrow X_{f(k)} = 1$ (if there is a good path through $f(k)$, the path from 0 to $f(k)$ must be good) and $Z_k = 0 \Rightarrow Y_k = 0$ (if the link k is bad, all paths through k are bad), and the Z_k are independent random variables. Thus

$$\begin{aligned} 1 - F_k &= \mathsf{P}[Z_k = 0 | Y_k = 0, Y_{f(k)} = 1] \\ &= \frac{\mathsf{P}[Z_k = 0]}{\mathsf{P}[Y_k = 0 | X_{f(k)} = 1]} \end{aligned} \tag{2}$$

Now the numerator is $\overline{\alpha}_k$. The denominator, which we denote by δ_k, can be expressed through the recursion

$$\begin{aligned} \delta_k &= \mathsf{P}[Y_k = 0 | X_{f(k)} = 1] \\ &= \mathsf{P}[Z_k = 0] + \mathsf{P}[Z_k = 1] \prod_{j \in d(k)} \mathsf{P}[Y_j = 0 | X_k = 1])) \\ &= \overline{\alpha}_k + \alpha_k \prod_{j \in d(k)} \overline{\delta}_j, \end{aligned} \tag{3}$$

with the convention that an empty product is 0 for k a leaf node.

Clearly the value of δ_k depends on the topology. First consider a perfectly balance tree with branching ratio r and uniform $\alpha_k = \alpha$. For this tree we can thus write

$$\delta_{f(k)} = D_{\alpha,r}(\delta_k) := \overline{\alpha} + \alpha \delta_k^r \tag{4}$$

As we move up the tree, the value of δ_k increases towards a limit which is a fixed point of the iteration of $D_{\alpha,r}$:

THEOREM 1. *(i) When $\alpha r > 1$, the equation $\delta = D_{\alpha,r}(\delta)$ as a unique fixed point $\delta^*(\alpha, r)$ in the interval $(0, 1)$.*

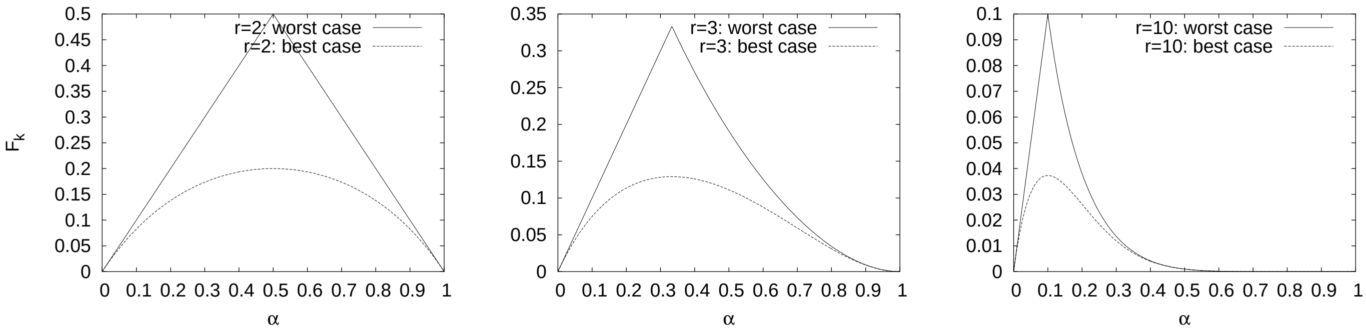

Figure 4: Bounds on false positive rate for identification of bad link, as function of the fraction α of good links, for branching ratios $r = 2, 3$ and 10. Note different vertical scales.

(ii) When $\alpha r \leq 1$, the equation $\delta = D_{\alpha,r}(\delta)$ exactly one fixed point $\delta^*(\alpha, r) = 1$.

(iii) The sequence $\delta^{(n+1)} = D_{\alpha,r}(\delta^{(n)})$ with $\delta^{(0)} = \overline{\alpha}$ is increasing.

(iv) The sequence $\{\delta^{(n)}\}$ converges to $\delta^*(\alpha, r)$.

Since $\{\delta^{(n)}\}$ is increasing, we can bound the denominator in the RHS of (2) to find an upper bound (i.e. worst case) for the false positive rate F_k. This extends to an *arbitrary* tree with non-uniform fraction of bad links.

THEOREM 2. *(i) In a perfectly balanced tree with branching ratio r and constant $\alpha_k = \alpha$,*

$$F_k \leq 1 - \frac{\overline{\alpha}}{\delta^*(\alpha, r)} \tag{5}$$

(ii) In an arbitrary tree, let $\alpha_k^{\min} = \min\{\alpha_j : j \preceq k\}$ be the minimum of probabilities α_j for links to be good on the subtree descended from k, and let $r_k^{\min} = \min\{\#d(j) : j \preceq k, j \notin R\}$ be the minimum branching ratio in the subtree descended from k.

$$F_k \leq 1 - \frac{\overline{\alpha}_k}{\delta^*(\alpha_k^{\min}, r_k^{\min})} \tag{6}$$

The worst case lower bound is for branching ratio $r = 2$. Here $\delta^*_{\alpha,2} = \min\{1, \overline{\alpha}/\alpha\}$, and hence $1 - \overline{\alpha}/\delta^*(\alpha, 2) = 1 - \max\{\alpha, \overline{\alpha}\}$. We plot $1 - \overline{\alpha}/\delta^*(\alpha, r)$ for $r = 2, 3$ and 10 in Figure 4. We also plot the best case for the given r, replacing δ^* by $\delta^{(1)}$, corresponding to the node k having r leaves as children. The following observations, read from the graphs, can be established rigorously.

- The false positive rate approaches 0 for large α (i.e. small fraction of bad links).
- For $r > 2$, the curve of the false positive rate becomes flat as α approaches 1. Hence the false positive rate is insensitive to the fraction of bad links, provided this is small.

The probability that link k is identified as bad is $P_k = \mathsf{P}[Y_k = 0, Y_{f(k)} = 1]$. Let $A_k = \prod_{j \succeq k} \alpha_k$ denote the probability for the entire path between the root 0 and node k to be good.

THEOREM 3. $P_k = A_{f^2(k)}(\overline{\delta}_{f(k)} - \overline{\delta}_k \alpha_{f(k)})$, *except* $P_1 = \delta_1$.

The total rate of false positives is thus

$$F = \sum_{k \in U} F_k P_k / \sum_{k \in U} P_k. \tag{7}$$

3.2 Coverage in Identifying Bad Links

In the previous section we saw that the link k at the head of a maximal bad subtree is increasingly likely to be bad when bad links are rare. However, we did not exclude the possibility of bad links elsewhere in the subtree. We now evaluate the performance of the inference rule in identifying all bad links. We compute the link coverage $C_k = \mathsf{P}[Z_k = 0, Y_k = 0, Y_{f(k)} = 1]$: the probability that link $k \in U$ is bad and designated as such.

THEOREM 4. $C_k = P_k \overline{\alpha}_k / \delta_k$

Using Mathematica [7], we have implemented symbolic computation of the δ_k, and hence C_k. As a summary performance statistic, we compute the coverage $C = \sum_{k \in U} C_k / \sum_{k \in U} \overline{\alpha}_k$, i.e., the average proportion of bad links that are correctly identified.

Let $\mathcal{T}_\alpha(r_1, \ldots, r_n)$ denote the perfectly balanced tree of depth n with successive branching ratios $r_1, \ldots, r_n$, and uniform probability α for a link to be good. We plot C for several such topologies in Figure 5. The left figure is for trees of depth 2 but increasing branching ratio. The coverage is relatively insensitive to the branching ratio. This reflects a trade-off: on the one hand, we have seen in Figure 4 that the probability of correct designation of a bad link at the root of a maximal bad subtree increases with the branching ratio. On the other hand, the impact of an incorrect designation increases with branching ratio, since $Y_k = 0$ but $A_k = 1$ requires a higher number of bad nodes in the subtree rooted at k. An even high number of nodes is impacted similarly when the tree depth increases: the middle figure shows that C decreases as the depth increases at constant branching ratio.

3.3 Scaling Behavior For Deep Networks

If the tree depth d increases while α remains constant, the chance α^d for a given path to be good decreases towards zero. But as networks grow, the links must perform better in order to maintain the same path quality. Thus, in modeling deep networks we consider *constant path failure rate* scaling: the chance for a link to be good is $\alpha^{1/d}$, so that the chance for a path to be good remains constant.

Figure 5(right) shows the behavior of C as the tree depth increases in the constant path failure rate scaling, using depth d trees $\mathcal{T}_{\alpha^{1/d}}(2, \ldots, 2)$ for $d = 2, 3, 4, 5$. Observe that for most α, C is almost independent of the tree depth. It can be shown that in a perfectly balanced tree with constant branching ratio $r \geq 2$ and uniform link probabilities α, the slope of C is always shallower than 1. Summarizing, the fraction of correctly identified bad links is roughly equal to the fraction of good paths in any such topology.

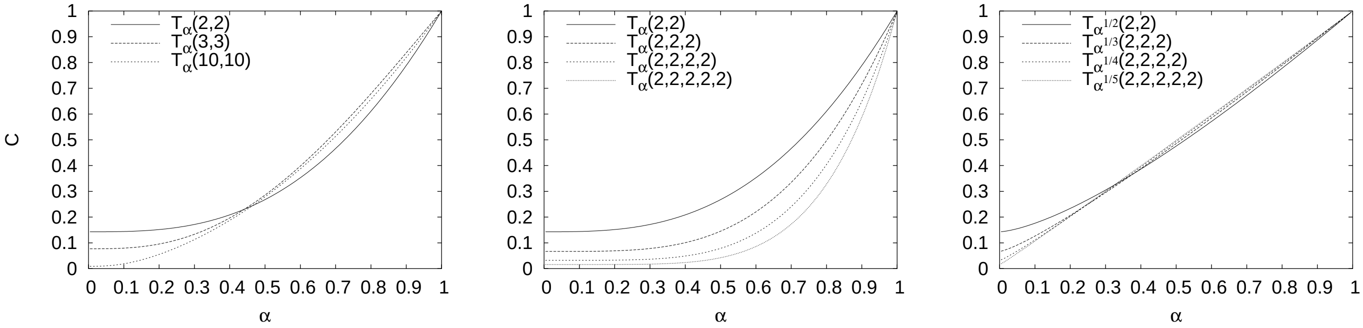

Figure 5: COVERAGE: THE PROPORTION OF CORRECTLY IDENTIFIED BAD LINKS, AS FUNCTION OF FRACTION OF GOOD LINKS. Left: insensitivity to branching ratio. Middle: decrease with increasing tree depth. Right: insensitivity to tree depth in the constant path failure rate scaling.

α	F	C	C'
0.95	0.02%—0.08%	75%—82%	95%
0.9	0.09%—0.24%	55%—67%	91%
0.8	0.4%—0.9%	29%—43%	81%

Table 1: Approx. 1000 node tree. False Positive Rate F, coverage C, and coverage C' under constant path failure scaling, as function of fraction α of good links.

3.4 Comparison with Other Approaches

We would like to compare the performance of the inference rule with the methods of [6] in identical topologies; these are random, although only the total number of nodes and maximum branching ratio are specified. To get a rough idea, we computed the coverage C and false positive rate F on several topologies comprising approximately 1,000 nodes with maximum branching ratio 10, and depth between 3 and 6. These are shown in Table 1. From these results our impression is that coverage is roughly equivalent to that of the linear programming (LP) algorithm of [6] (somewhat better when bad links are rare, somewhat worse when they are common), while the false positive rate is at least as good as that of any method considered there. In computational complexity, we expect our algorithm to be less complex than LP, and far less complex than the general most accurate method presented, Gibbs Sampling. We also show the coverage rate C' in the constant path failure scaling. This is barely sensitive to topology, and approximately equal to the proportion of good links.

4. DISCUSSION AND FURTHER WORK

This paper has argued that when network link performance characteristics can be well separated into two categories, good and bad, a simple inference algorithm—that attributes path failure to the smallest set of consistent link failures—can be effective in identifying candidate bad links on a tree from end-to-end measurements. This approach is justified by the observation that when bad links are uncommon, the two or more badly performing paths likely have a bad link in their intersection. Moreover, the likelihood for this to happen is relatively insensitive to changes if the fraction of (uncommon) bad links. Conversely, the false positive rate is very low in this regime, because only those (rare) good links at the head of maximal bad subtrees are falsely deemed bad.

On the negative side, a bad link will not be identified if there another bad link between it and the root. For some applications, this need not be regarded as a deficiency. Suppose only limited time and/or resources are available to "repair" bad links, i.e. make them good. Depending on context, this could be achieved by replacing a bad component, or by rerouting traffic away from them. In repairing the link at the head of a maximal bad subtree, not only was the link likely bad, but repairing has potential benefit to the largest number of downstream paths. Repeating the algorithm after further measurements could then identify likely downstream bad links.

Finally, the notion of separability can also be applied to performance measurements taken over a set of time intervals. Assume that in each interval t, link k is good with probability α_k, independently for different k and t. (α_k may depend on the interval width). The measured data comprises the good/bad state $X_{k,t}$ observed at each receiver k in interval t. Under the assumption of separability, the two-state Maximum Likelihood Estimator from [1] can be used to estimate the α_k. Both this method and the work described in the rest of the paper are to be evaluated experimentally in future work.

5. REFERENCES

[1] A. Adams, T. Bu, R. Cáceres, N.G. Duffield, T. Friedman, J. Horowitz, F. Lo Presti, S.B. Moon, V. Paxson, D. Towsley, "The Use of End-to-End Multicast Measurements for Characterizing Internal Network Behavior", IEEE Communications Magazine, May 2000.

[2] R. Caceres, N.G. Duffield, T. Friedman, "Impromptu measurement infrastructures using RTP", Proc. IEEE Infocom 2002, New York, June 23-27, 2002.

[3] M. Coates, A. Hero, R. Nowak B. Yu, "Internet Tomography", IEEE Signal Processing Magazine, May 2002.

[4] Y. Tsang, M. Coates and R. Nowak, "Passive Unicast Network Tomography based on TCP Monitoring", Rice University, ECE Department Technical Report TR-0005, 2000.

[5] "Packet Wingspan Distribution", NLANR. See `http://www.nlanr.net/NA/Learn/wingspan.html`

[6] V. N. Padmanabhan, L. Qiu, and H. Wang, "Server-based Inference of Internet Link Lossiness", IEEE Infocom 2003, San Francisco, CA, USA April 2003.

[7] Wolfram Research, Inc., Mathematica, Version 4, Champaign, IL, 1999.

[8] Y. Zhang, N.G. Duffield, V. Paxson, S. Shenker, "On the Constancy of Internet Path Properties", ACM SIGCOMM Internet Measurement Workshop 2001, San Francisco, CA, November 1-2, 2001.