Network Tomography of Binary Network Performance Characteristics

Nick Duffield

AT&T Labs–Research
180 Park Avenue, Florham Park, NJ 07932, USA
E-mail: duffield@research.att.com

Abstract

In network performance tomography, characteristics of the network interior, such as link loss and packet latency, are inferred from correlated end-to-end measurements. Most work to date is based on exploiting packet level correlations, e.g., of multicast packets or unicast emulations of them. However, these methods are often limited in scope—multicast is not widely deployed—or require deployment of additional hardware or software infrastructure.

Some recent work has been successful in reaching a less detailed goal: identifying the lossiest network links using only uncorrelated end-to-end measurements. In this paper we abstract the properties of network performance that allow this to be done and exploit them with a quick and simple inference algorithm that, with high likelihood, identifies the worst performing links. We give several examples of real network performance measures that exhibit the required properties. Moreover, the algorithm is sufficiently simple that we can analyze its performance explicitly.

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. Several methods have been proposed over the last few years to infer link level packet loss and latency, and even the underlying network topology. Initial work exploited the inherent correlations between copies of a multicast packet seen at different endpoints; see [6, 15, 13, 5] and the review in [1]. Subsequent work emulated this approach using clusters of diversely addressed unicast packets [8], and other packet group techniques; see [9, 10, 20, 21, 22, 23], and [8] for a review. Probing and measurement collection functions for tomography have been embedded within transport protocols, thus co-opting suitably enabled hosts to form impromptu measurement infrastructures; see [7] and [20].

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 from routers in order to function. In practice, ICMP response is restricted by some network administrators (presumably to prevent probing from external sources).

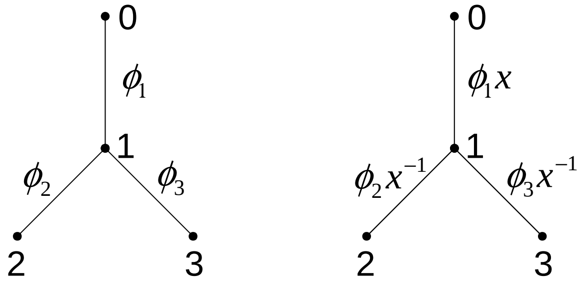

Figure 1: ABSENCE OF IDENTIFIABILITY FROM UNCORRELATED MEASUREMENTS: trees have different link transmission probabilities but identical end-to-end transmission probabilities.

There are several challenges in bringing network performance tomography to fruition and enabling widespread performance tomography. Multicast is not widely deployed. Even for methods based on unicast probing, there are development and administrative costs associated with deploying probing and data collection software. This has motivated the goal of reducing such costs by developing inference methods that can work with readily available end-to-end measurements.

1.2 The Need for Correlated Measurements

The requirement of network tomography for correlated measurements is illustrated by the following model. 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 source at node 0 to the leaves 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 end-to-end transmission probabilities are the same when the link probabilities are adjusted as in Figure 1(right), for any multiplicative factor x between $\max\{\phi_2, \phi_3\}$ and $1/\phi_1$. (This condition yields link probabilities less than or equal to 1). Consequently, independent measurement of the (two) transmission rates from the root 0 to each of the leaf nodes 2 and 3 is insufficient to determine the (three) link transmission probabilities ϕ_1, ϕ_2 and ϕ_3 uniquely. The ϕ_i are not *not statistically identifiable* from the data, meaning that different sets of parameters exist that give rise to the same statistical distribution of data.

1.3 Measurement and Packet-Level Correlation

The inherent *packet level* correlation of multicast packets can be exploited for tomography. When a given multicast probe is observed at multiple end points, the contribution to packet performance from the common portion of the packets' paths is identical. This is the property that makes the link performance parameter identifiable. Unicast tomography aims to reproduce similar correlations in groups of unicast packets. One approach has been to emulate a multicast probe with a "stripe" of closely spaced unicast packets with different IP destination addresses. The idea is that in the common portion of the packet path, the performance experienced by the packets will be strongly correlated. If the correlation were perfect, the behavior of the probes, and the inferences drawn, would be identical to that of a notional multicast packet that followed the same routing tree. Experimental studies using stripes to the same destination confirm that the correlation is strong, although not perfect. The paper [14] proposed enhancing this approach with a form of data selection that gives more weight to stripes that exhibited the strongest correlation. A related proposal in [9] is to take the imperfect correlations explicitly into account through introducing more parameters into the link model, then to reduce back the number of independent parameters by coupling the parameters through a queueing model in order to render the model identifiable.

1.4 Inference in the Absence of Packet Level Correlations

Despite the methodological advances described in the previous section, the need to install measurement software at receivers represents a barrier to widespread deployment. A recent approach to overcome these barriers has been proposed in [18]. By measuring the packet stream at or near a web server, loss statistics for the end-to-end paths from the server to each client are determined by observing TCP retransmissions. In distinction from the work mentioned above, this approach does not assume or attempt to exploit any packet-level correlations in the network experience of packets destined for different clients. Packets are only assumed to have the same probability of being lost on given link, independent of the path they take through it. 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.

From the discussion in Section 1.2 it should be clear that the link rates in the model of [18] are not identifiable. Nevertheless some of the inference methods proposed in [18] are quite successful in identifying the lossier 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. However, the most accurate methods are computationally very intensive. By understanding the structural properties that underpin these methods, we aim instead 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.

1.5 Performance Level Correlation

The ability to identify the worst performing links relies on a structural assumption about link performance. Even if we do not perform correlated end-to-end measurements at the individual packet level, it is still reasonable to expect that two distinct packet streams that pass through a given link over the same period of time would exhibit some correlation in performance at a statistical level. A model for this is as follows. When a packet stream traverses a link, each packet may be subjected to a performance impairment: it can be lost or delayed. If each packet's impairment were known, one could calculate a summary statistic (e.g. loss rate or mean delay) that would be mapped down to a binary performance measure by setting a threshold. When the statistic exceeds the threshold, the performance is classified as "bad"; otherwise, it is classified as "good". Clearly, the same classification could be made for paths comprising multiple links, although different thresholds might be used in each case. The above scheme is generic and makes no specific assumptions. The possibility for performance inference comes if the binary performance measure satisfies the following conditions:

(A1) There is a class of packet streams such that all streams in the class experience either good or bad performance. Accordingly, we say that the link is good or bad.

(A2) A path is bad if and only if at least one link on the path is bad.

In this case we say that the performance measure is *separable*. The interpretation of separability is as follows. (A1) says that the binary performance ("good/bad") is perfectly correlated for packet streams in the class. (A2) says that a bad path cannot arise through a set of "partially" bad links. This property gives us the terminology of separability, meaning that the characteristics of good and bad paths are sufficiently distinct.

In Section 2 we will describe a number of network models in which the separability assumption, or a weaker version of it, hold. For now we observe that a separable binary performance model maps exactly onto the model for loss of a single multicast packet propagating down a multicast tree, with badness corresponding to packet loss. In more detail, (A2) is analogous to saying that if a packet is lost on a link, it reaches no leaves descended from that node; (A1) is analogous to saying that if a packet does not reach a leaf, it must have been lost on some link en route. This structural equivalence means that we have, in principle, all methods available from multicast loss inference at our disposal in order to infer the distribution of link badness.

1.6 Contribution and Summary

In this paper we develop the framework outlined above and show how it can be used to infer the locations of badly performing links.

(a) Section 2 defines the notion of separability for performance measures and argues that it is satisfied both by performance models treated in the literature, including those of [18]. We also introduce a notion of weak separability in which good links always give rise to good paths (but the converse need not be true). We show that any binary performance model can always be adjusted so that weak separability holds.

(b) Section 3 describes a static algorithm—the smallest consistent failure set (SCFS) algorithm—for inferring the locations of bad links in a routing tree, using a single measurement of the good/bad status of each source-to-leaf path.

(c) The SCFS algorithm is sufficiently simple that its performance can be analyzed explicitly. In Section 4 we derive its false positive rate and detection rate for identifying bad links under the assumptions of strict separability. We show that the false positive rate is very small for a likely range of probabilities for a link to be bad. We confirm the results of the analysis with some model-based computations.

(d) Although the false positive rate is low, the detection rate for bad links can be noticeably less than 1. Section 5 describes an algorithm in which iterative application of measurement and inference is use to detect all bad links. We compute the overhead, i.e., the amount by which the number of candidate bad links exceed the actual number of bad links. We show that the excess is quite small for likely model parameters.

(e) Section 6 extends the analysis of Section 4 to the general weakly separable case. We define the notion of a critical link as a link which makes all paths through it bad. We obtain bounds for the false positive rate and the detection rate in terms of general characteristics of the critical link. We illustrate with some measured path performance data from the internet.

(f) Section 7 compares the performance of SCFS with the algorithms in [18] and another method recently proposed in [3]. SCFS has a noticeably better false positive rate than the other methods.

(g) We conclude in Section 8. In a discussion of potential further work, we outline how time series of path measurements can be used to infer the probabilities of link badness when these vary according to a stochastic process. This is achieved by mapping the problem into the multicast loss inference problem that was solved in [6].

(h) Proofs of the Theorems are deferred to the Appendix.

1.7 Related Work

The approach of this paper was first proposed in the conference report [11], where the analytical results in Section 4 for the strictly separable case were presented without proof. The current paper supplies the proofs; the methods and analysis in Sections 5 (exhaustive inspection) and 6 (the weakly separable case) are new.

Partly in response to our paper [11], a recent paper [3] reported network level simulations of SCFS, some of the algorithms from [18], and a new proposed algorithm COBALT. In a comparison of different inference algorithms, SCFS was found to have the lowest false positive rate, being about one third the rate of the next best algorithm. The detection rate (or coverage, i.e., the proportion of bad links correctly identified) was as good as other methods when bad links were rare but fell off when they were more common. The strong experimental performance of SCFS against other algorithms is one motivation for completing our theoretical performance study in the present paper. These results are discussed further in Section 7.

We mention also some recent work in which measurements of sets of packets (rather than individual packets) were correlated for tomographic purposes. The paper [2] proposed correlating flow records in order to identify congested links. The idea here is that the throughput of elastic traffic flows will become correlated during a common congested period. In [12], aggregate loss statistics reported by multicast session users using the RTP protocol are correlated in order to infer link loss rates. The idea is that even though the loss statistics are aggregated over multiple packets, correlations due to loss of individual packets are still visible. Fault isolation in multicast networks using scoped multicast traceroute (mtrace) has been proposed in [19]. Unlike the tomographic methods, this requires participation by network routers (to respond to mtrace requests).

2 Network and 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.

2.1 Tree Model and Terminology

We assume that the network topology is known. The 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 connect 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 $d(k) = \{j \in V : (k,j) \in L\}$. For convenience we sometimes write $U = V \setminus \{0\}$, i.e., the set of all non-root nodes. 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. ℓ_k will denote the depth of the node k from the root.

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, a packet is subject to a performance degradation (e.g. loss or delay) according to a distribution specified by a parameter ϕ_k. The degradation is independent across different links and different packets. If the source-destination path comprises links $k_1, \ldots, k_m$, the performance degradation along the path follows a composite distribution described by the parameters $\phi = \{\phi_{k_1}, \ldots, \phi_{k_m}\}$.

Let ψ be the expected value of some statistic computed from link or path performance distributions; we write $\psi_k(\phi_k)$ and $\psi_{k_1,\ldots,k_m}(\phi_{k_1}, \ldots, \phi_{k_m})$ respectively. For each link or path we partition the set $\Psi_{k_1,\ldots,k_m}$ of possible ψ values into two subsets that we we call "good" and "bad". Likewise, we call the link or path (or their parameters) bad if and only if the expected statistic $\psi_{k_1,\ldots,k_m}(\phi_{k_1}, \ldots, \phi_{k_m})$ 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:

Definition

- 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.

Note that we can always arrange for weak separability by defining $\Psi^{\mathrm{good}}_{k_1,\ldots,k_m} = \{\psi(\phi_{k_1},\ldots,\phi_{k_m}) \mid \psi(\phi_{k_i}) \in \Psi^{\mathrm{good}}_{k_i}\}$. The extent to which this is useful then depends how frequently a good path does, in fact, contain a bad link. We return to this question in Section 6. For now we illustrate the separability framework with some examples.

2.3 Examples of Separable Performance Models

Connectivity If a link or a path is good, it transmits all packets; if bad, it transmits none. Thus the path is bad if and only if 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 [18], 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.

General Loss Model In a more general loss model, given a threshold link transmission probability t, we call the link k good if its transmission probability $\phi_k > t$; otherwise it is bad. The model is weakly separable if each comprising ℓ links is designated as good if and only if its path transmission probability exceeds ϕ_k^ℓ. The model is not in general strictly separable unless further conditions are imposed on the distribution of transmission probabilities, e.g, as in the LM_1 model described above.

In the related model LM_2 from [18], 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, with threshold $t = 0.99$. We chose $t^\ell = 0.99^\ell$ to be the threshold transmission rate separating good and bad paths of ℓ hops, then the partition is weakly separable: all paths containing only good links are designated as good.

Consider a path $\{k_1, \ldots, k_\ell\}$. Conditioned on one of the links, say k_i, being bad, the chance for a path to be designated good is

$$\mathsf{P}[\prod_{j=1}^{\ell} \phi_j > t^\ell \mid i \text{ bad}] = \mathsf{P}[\prod_{j=1}^{\ell} \phi_j > t^\ell \mid \phi_i < t] \leq \mathsf{P}[\phi_i > t^\ell \mid \phi_i < t] = \mathsf{P}[\phi_i \in (t^\ell, t) \mid i \text{ bad}] \tag{1}$$

Note that this bound depends on the probability for a link to be bad only through the threshold t; otherwise it depends only on the distribution of loss rates of bad links. For the LM_2 model, the bound is $1 - 0.99^{\ell-1}$, e.g., about a 4% chance for $\ell = 5$. Thus the paths containing bad links can still be identified with high probability. We make a systematic study of the impact of departures from strictly separability in Section 6, including some examples based on network measurements of packet loss.

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:

$$\psi_{k_1,\ldots,k_m}(\phi_{k_1}, \ldots, \phi_{k_m}) = \sum_{i=1}^{m} \psi_{k_i}(\phi_{k_i}) \tag{2}$$

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 [25]. 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. We assume that for a given application, delay spikes with round-trip times exceeding a certain level z are not tolerable. Consequently, 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 [25] are consistent with assumptions (A1) and (A2). Here we attribute delay spikes to individual links, and characterize them by mean duration d_s and mean interarrival time τ_s. (In our model, the delays are one-way, rather than RTTs).

First, (A1). The probability for a packet to encounter a delay spike on a single link is d_s/τ_s. Assuming that spikes occur independently, the probability q for a packet to encounter more than one delay spike on a path comprising ℓ hops is $q = 1 - (1 - d_s/\tau_s)^\ell - \ell(1 - d_s/\tau_s)^{\ell-1} d_s/\tau_s$. This probability increases with path length. Taking $\ell = 30$, larger than most paths today (see [17]) 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. Thus the chance of encountering more than one spike is very small for the observed spike characteristics.

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 $n = d_s rT/\tau_s$, 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 10 kByte/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 in this case; at least a handful of probes will encounter a spike during the measurement interval on average, and the chance for at least one probe-spike encounter is close to 1.

3 Smallest Consistent Failure Set (SCFS) Inference Algorithm

This section defines the algorithm for inferring the identity of bad links. The Smallest Consistent Failure Set (SCFS) algorithm 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

$$X_k = \prod_{j \succeq k} Z_j \tag{3}$$

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. $Y_k = 1$ if and only if 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 algorithm takes the other extreme by designating as bad the link in the subtree that is most likely to be amongst the set of bad links, namely, the link k. For suppose, on the other hand, that 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

```
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 algorithm. Recall 1 denotes the single child node of the root node 0.

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 algorithm 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) Algorithm

1. **Input:** Tree $\mathcal{T}$, End-to-end measurements $\{X_k\}_{k \in R}$;
2. $W' = \{k \in U \mid \max_{j \in R_k} X_j = 0\}$;
3. $W = \{k \in W' \mid f(k) \notin W'\}$;
4. **Output:** W;

An explicit implementation of the SCFS algorithm determines W by recursion on 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 algorithm 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. Hence, the status of the links a and b, and the subtrees descended from them is uncertain. The right figure shows the action of the inference algorithm. The links a and b are designated as bad while all the links in the 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 unambiguously be declared bad since the maximal bad subtree descended through k has one member, namely, k itself.

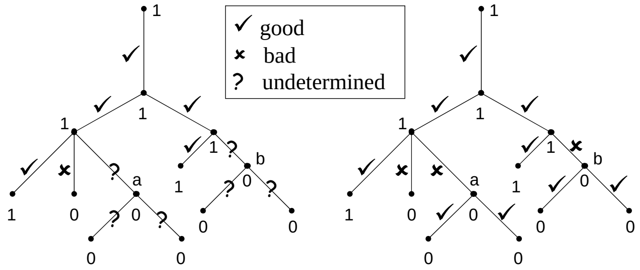

Figure 3: OPERATION OF THE SCFS ALGORITHM: Y_k shown at each node k. Left: status of link a, b and subtree descended from them are uncertain because paths to these subtrees are bad. Right: SCFS attributes badness to common path (links a and b) and designates remaining nodes as good.

4 SCFS Performance Analysis: Strictly Separable Case

The SCFS algorithm codifies a parsimonious approach that might well be taken without the benefit of analysis, specifically, attributing a pattern of bad paths as being due to badness in the smallest possible set of interior links consistent with the pattern. This section analyzes the performance of the SCFS algorithm under a statistical model for the distribution of bad links in the network. Following [18] 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 algorithm under this statistical model. For compactness we will use the notation $\overline{\alpha} = 1 - \alpha$ in what follows.

4.1 False Positive Rate

We now provide a justification of the inference algorithm by analyzing its performance. Recall that the inference algorithm designates link k as bad 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 (FPR) associated with a link is the probability that the link was designated as bad when it was in fact good. The FPR for link k is thus

$$\mathrm{FPR}_k = \mathsf{P}[Y_k = 0, Y_{f(k)} = 1 | Z_k = 1] \tag{4}$$

The tree average FPR is the expected number of false positives divided by the expected number of good nodes, i.e.,

$$\mathrm{FPR} = \frac{\sum_{k \in U} \mathrm{FPR}_k \alpha_k}{\sum_{k \in U} \alpha_k} \tag{5}$$

The main work of this section is to find bounds for the FPR under various levels of generality. To this end, we will find it useful to define a number of subsidiary quantities. Set

$$\beta_k := \mathsf{P}[Y_k = 1 | X_{f(k)} = 1] \tag{6}$$

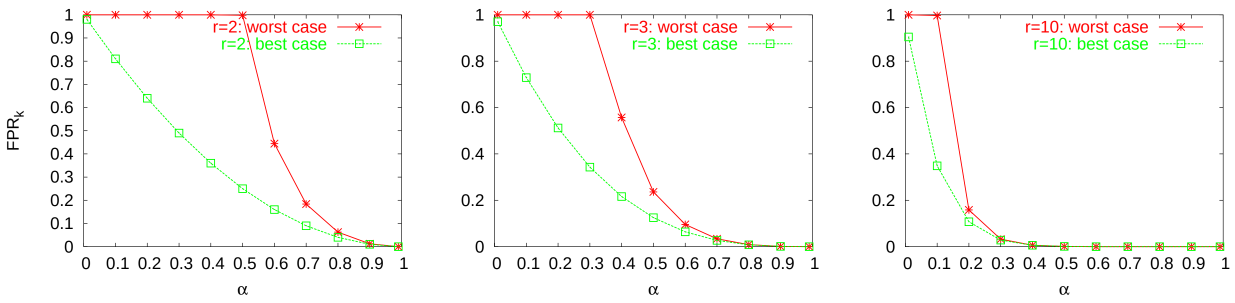

Figure 4: Bounds for FPR_k for identification of bad link, as function of the fraction α of good links, for branching ratios $r = 2, 3$ and 10.

Let $Y_{f(k),k} = \max_{j \in R_{f(k)} \setminus R_k} X_j$. This takes the value 1 if and only if a packet reaches at least one of the leaves descended from $f(k)$, but not including those leaves descended through k. Set $\gamma_{f(k),k} = \mathsf{P}[Y_{f(k),k} = 1]$. An exact expression for the FPR is as follows.

Theorem 1 $\mathrm{FPR}_k = (1 - \beta_k/\alpha_k)\gamma_{f(k),k}$ *except* $\mathrm{FPR}_1 = 1 - \beta_1/\alpha_1$.

Now β_k is expressed through the recursion:

$$\begin{aligned}\beta_k &= 1 - \mathsf{P}[Y_k = 0 | X_{f(k)} = 1] \\ &= 1 - \mathsf{P}[Z_k = 0] - \mathsf{P}[Z_k = 1] \prod_{j \in d(k)} \mathsf{P}[Y_j = 0 | X_k = 1])) \\ &= \alpha_k (1 - \prod_{j \in d(k)} \overline{\beta}_j), \end{aligned} \tag{7}$$

with the convention that for leaf nodes $k \in R$, an empty product is 0. The value of β_k depends on the topology. First consider a uniform tree with branching ratio r and $\alpha_k = \alpha$. Since β_k must be equal for all siblings, we can write

$$\beta_{f(k)} = B_{\alpha,r}(\beta_k) := \alpha(1 - \overline{\beta}_k^r) \tag{8}$$

As we move up the tree, the value of β_k decreases towards a limit which is a fixed point of the iteration of $B_{\alpha,r}$. The following Theorem summarizes the main technical properties of the iteration that we shall employ.

Theorem 2 *(i) When $\alpha r > 1$, the equation $\beta = B_{\alpha,r}(\beta)$ as a unique fixed point $\beta^*(\alpha, r)$ in the interval $(0, 1)$.*

(ii) When $\alpha r \leq 1$, the equation $\beta = B_{\alpha,r}(\beta)$ has exactly one fixed point $\beta^*(\alpha, r) = 0$.

(iii) The sequence $\beta^{(n+1)} = B_{\alpha,r}(\beta^{(n)})$ with $\beta^{(0)} = \alpha$ is decreasing.

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

(v) $\beta^*(\alpha, r) \geq \beta^*(\alpha, r')$ *for* $r > r'$*.*

(vi) $\beta^*(\alpha, r) \geq \beta^*(\alpha', r)$ *for* $\alpha > \alpha'$

Since $\{\beta^{(n)}\}$ is decreasing, we can bound the expression of (57) to find an upper bound on the false positive rate. This extends to an *arbitrary* tree with a non-uniform fraction of bad links. 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.

For the special case $r = 2$, $\beta^*(\alpha, 2)$ can be computed explicitly as a solution to $\beta = \alpha(1 - (1-\beta)^2)$, namely

$$\beta^*(\alpha, 2) = \begin{cases} 1 - \overline{\alpha}/\alpha, & \alpha > 1/2 \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

Combined with the bound Theorem 2(v), this property enables us to establish explicit bounds on the FPR in general topologies, as we now show.

Theorem 3 *(i) In a perfectly balanced tree with branching ratio* r *and constant* $\alpha_k = \alpha$*,*

$$\mathrm{FPR}_k \leq F(\alpha, r) := 1 - \frac{\beta^*(\alpha, r)}{\alpha} \leq F(\alpha, 2) = \min\{1, (\overline{\alpha}/\alpha)^2\} \tag{10}$$

(ii) In an arbitrary tree, then $\beta_j \geq \beta^*(\alpha_k^{\min}, r_k^{\min})$ *for any* $j \preceq k$*, and hence*

$$\mathrm{FPR}_k \leq 1 - \frac{\beta^*(\alpha_k^{\min}, r_k^{\min})}{\alpha_k} \tag{11}$$

We plot the bound $F(\alpha, r)$ for $r = 2, 3$ and 10 in Figure 4. We call this the worst case bound, since by Theorem 2(iii), it is exceeded at no node in the tree. On the other hand, we know $\mathrm{FPR}_k \leq 1 - \beta^{(1)}/\alpha$ for a node k whose children are leaf nodes. We call this the best case bound. We observe the following general behavior in Figure 4.

- FPR_k approaches 1 for large α (i.e. small fraction of bad links).
- The curve of FPR_k becomes flat as α approaches 1. Hence FPR_k is insensitive to the fraction $\overline{\alpha}$ of bad links, provided this is small.

This behavior is confirmed by the following:

Theorem 4 *As* $\overline{\alpha} \to 0$*,*

(i) $\beta^*(\alpha, r) = \alpha - \overline{\alpha}^r(1 + O(\overline{\alpha}))$.

(ii) $F(\alpha, r) = \overline{\alpha}^r(1 + O(\overline{\alpha}))$.

4.2 Detection Rate for 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 algorithm in identifying all bad links. The detection rate (or coverage) of bad links is the probability that a bad link will be designated as bad, i.e.,

$$\mathcal{C}_k = \mathsf{P}[Y_k = 0, Y_{f(k)} = 1 | Z_k = 0] \tag{12}$$

The detection rate $\mathcal{C}$ of bad links over the whole network is the expected total number of bad links designated as bad divided by the total number of bad links, i.e.,

$$\mathcal{C} = \frac{\sum_{k \in U} \mathcal{C}_k \overline{\alpha}_k}{\sum_{k \in U} \overline{\alpha}_k} \tag{13}$$

We now establish lower bounds for the detection rate. 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 composed of good links.

Theorem 5 *(i)* $\mathcal{C}_k = \gamma_{f(k),k}$ *except* $C_1 = 1$.

(ii) $\gamma_{f(k),k} = A_{f^2(k)}(\beta_{f(k)} - \beta_k \alpha_{f(k)})/\overline{\beta}_k$

(iii) In a uniform tree with $\alpha_k = \alpha$ *and branching ratio* r

$$\gamma_{f(k),k} \geq A_{f^2(k)} \overline{\alpha}/(1/\beta^*(\alpha, r) - 1) \tag{14}$$

(iv) In a general tree

$$\gamma_{f(k),k} \geq A_{f(k)}(1/\alpha^{\min}_{f(k)} - 1)/(1/\beta^{\min}_{f(k)} - 1) \geq A_{f(k)} \max\{0, 1 - \overline{\alpha}^{\min}_{f(k)}/\alpha^{\min}_{f(k)}\} \tag{15}$$

For completeness we remark that the chance DB_k that link k is designated bad is

$$DB_k = \mathsf{P}[Y_k = 0, Y_{f(k)} = 1] = \mathsf{P}[Y_k = 0 | X_{f(k)} = 1]\mathsf{P}[Y_{f(k),k} = 1] = \overline{\beta}_k \gamma_{f(k),k} \tag{16}$$

This takes the value $A_{f^2(k)}(\beta_{f(k)} - \beta_k \alpha_{f(k)})$, except $DB_1 = \overline{\beta}_1$.

We have implemented symbolic computation of C_k using Mathematica [24]. Let $T_\alpha(r_1, \ldots, r_n)$ denote the perfectly balanced tree of depth $n+1$ with successive branching ratios $r_1, \ldots, r_n$, and uniform probability α for a link to be good. We plot C for several topologies in Figure 5. The left figure is for trees of depth 2 but increasing branching ratio. The detection rate 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 higher 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.

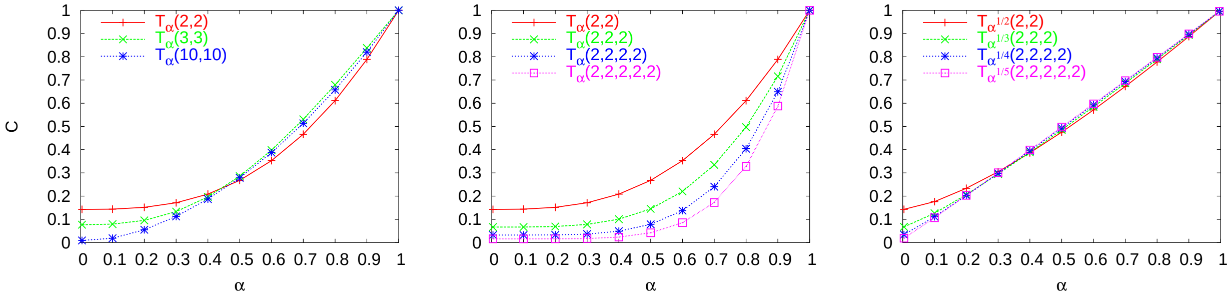

Figure 5: Detection Rate: the Proportion of Correctly Identified Bad Links, as function of fraction of good links. Left: insensitivity to branching ratio. Middle: decrease with increasing tree depth. Right: insensitivity to tree depth in the constant path failure rate scaling.

4.3 Asymptotic Behavior when Bad Links are Rare

We now examine the behavior of FPR_k and $\mathcal{C}_k$ for small $\overline{\alpha}$, i.e., when bad links are rare. We expect considerable simplification of the analytic results in this regime. Firstly, FPR_k should be approximately $\prod_{j\in d(k)} \overline{\alpha}_j$ for small $\overline{\alpha}$, i.e., the chance for all child links to be bad. Secondly, the detection rate computed in the previous section was found to be insensitive to (large) branching ratio.

These observations are made precise by the following result. Let $S_k = \sum_{j \succeq k} \overline{\alpha}_k$ and $T_k = \prod_{j\in d(k)} \overline{\alpha}_j$ with $S_j = T_j = 0$ for $j = 0$ and $j \in R$. Let $\alpha_+ = \max_{k\in U} \alpha_k$ and $\alpha_- = \min_{k\in U} \alpha_k$. I_Ω denotes the indicator function of the set Ω.

Theorem 6 *Consider the limit in which $\alpha_k \to 1$ for all $k \in U$, with $\overline{\alpha}_-/\overline{\alpha}_+$ bounded.*

(i) $\beta_k = \alpha_k - T_k(1 + O(\overline{\alpha}_-))$.

(ii) $\gamma_{f(k),k} = 1 - S_{f(k)} - T_{f(k)}/\overline{\alpha}_k + O(\overline{\alpha}_-^2)$

(iii) $\mathrm{FPR}_k = T_k(1 + O(\overline{\alpha}_-))$ *and hence* $\mathrm{FPR} = (\#U)^{-1} \sum_{k\in U} T_k(1 + O(\overline{\alpha}_-))$.

(iv) $\mathcal{C} = 1 - \frac{\sum_k \overline{\alpha}_k S_{f(k)} + T_{f(k)}}{\sum_k \overline{\alpha}_k} + O(\overline{\alpha}_-^2)$

(v) $\mathcal{C} \to 1$ *and* $\mathcal{C} \geq 1 - \ell\overline{\alpha}_- + O(\overline{\alpha}_-^2)$ *in the limit* $\overline{\alpha} \to 0$*, where* $\ell = \max_k \ell_k$ *is the depth of the tree.*

(vi) Let $\mathcal{C}(\alpha)$ denote the detection rate of bad links in a tree with uniform $\alpha_k = \alpha$.

$$\mathcal{C}'(1) = (\#U)^{-1} \sum_{k\in U} \ell_k - 1 + I_{\{\#d(f(k))=2\}} \tag{17}$$

(vii) In a perfectly balanced tree with depth ℓ and branching ratio r and uniform $\alpha_k = \alpha$,

$$\mathcal{C}'(1) = \begin{cases} \frac{\ell r^\ell}{r^\ell - 1} - \frac{r}{r-1} & \textit{if } r \neq 2 \\ \frac{\ell - 1}{1 - 2^{-\ell}} & \textit{if } r = 2 \end{cases} \tag{18}$$

$T_\alpha(2,2)$	exact	approx.	bound
α	$\mathcal{C}$	$1-\mathcal{C}'(1)\overline{\alpha}$	$1-\ell\overline{\alpha}$
0.95	0.890	0.886	0.85
0.90	0.789	0.771	0.7
0.80	0.611	0.543	0.4

$T_\alpha(2,2,2,2)$	exact	approx.	bound
α	$\mathcal{C}$	$1-\mathcal{C}'(1)\overline{\alpha}$	$1-\ell\overline{\alpha}$
0.95	0.810	0.794	0.75
0.90	0.649	0.587	0.5
0.80	0.404	0.174	0

$T_\alpha(5,5)$	exact	approx.	bound
α	$\mathcal{C}$	$1-\mathcal{C}'(1)\overline{\alpha}$	$1-\ell\overline{\alpha}$
0.95	0.913	0.911	0.85
0.90	0.831	0.823	0.7
0.80	0.676	0.645	0.4

$T_\alpha(5,5,5,5)$	exact	approx.	bound
α	$\mathcal{C}$	$1-\mathcal{C}'(1)\overline{\alpha}$	$1-\ell\overline{\alpha}$
0.95	0.825	0.812	0.75
0.90	0.675	0.625	0.5
0.80	0.436	0.250	0

Table 1: Detection Rate $\mathcal{C}$: Exact vs. Approximation and Bound from Theorem 6 for several uniform trees.

Example: LM_2 model. Consider again the LM_2 type model from Section 2.3. Specifically, consider link probabilities $\alpha_k = \alpha$ and consider the regime with small $\overline{\alpha}$. Then

$$\mathrm{FPR}_k \approx (\overline{\alpha})^{\#d(k)} \tag{19}$$

$$\mathcal{C}_k \approx 1 - \ell_k\,\overline{\alpha} - (\overline{\alpha})^{\#d(k)-1} \tag{20}$$

Take a binary tree ($\#d(k) = 2$) of depth 5 with each link having a probability $\alpha = 0.95$ to have a good transmission rate exceeding the threshold $t = 0.99$. Put another way, only 5% of links have a loss rate greater than 1%. Then $\mathrm{FPR}_k \approx 0.0025$ while the worst case $\mathcal{C}_k$ (for $\ell_k = 5$) is 0.7

In the case of uniform $\alpha_k = \alpha$, we can use the derivative $C'(1)$ from Theorem 6(iv) to form the approximation

$$\mathcal{C} \approx 1 - \mathcal{C}'(1)\overline{\alpha} \tag{21}$$

for small $\overline{\alpha}$. We compare this approximation, together with the bound from Theorem 6(iv), against the exact value of $\mathcal{C}$ in Table 1. The approximation works well (accurate to within a couple of percent in the cases examined) for low loss rates with $\alpha = 0.95$, and reasonably well (accurate to within about 10%) when $\alpha = 0.9$. Agreement is better for larger branching ratios and shallower trees.

4.4 Scaling Behavior For Deep Networks

If the tree depth d increases while α remains constant, then the chance α^d for a given path to be good decreases towards zero. But over the long timescales of network buildout, as network path lengths 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, in which 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 $\mathcal{C}$ as the tree depth increases in the constant path failure rate scaling, using depth d trees $T_{\alpha^{1/d}}(2,\ldots,2)$ for $d = 2,3,4,5$. Observe that for most α, $\mathcal{C}$ is almost independent

```
START: Conduct end-to-end measurements.
Apply SCFS algorithm to measurements.
if (no candidate bad links exist) {
        exit
}
else {
        inspect candidate bad links and repair any that are bad.
        go to START.
}
```

Figure 6: Exhaustive Inspection Algorithm under SCFS

of the tree depth. In fact, it can be shown that in a perfectly balanced tree with constant branching ratio $r > 2$ and uniform link probabilities α, the slope of $\mathcal{C}$ is always shallower than 1. Summarizing, we can say that the fraction of correctly identified bad links is roughly equal to the fraction of good paths in any such topology.

5 The Overhead for Exhaustive Inspection

5.1 Exhaustive Inspection

What is the cost of false positives? One way to measure this is to ask how many candidate bad links must be inspected before all true bad links are found. We focus on an iterative scheme in which we repeat measurement, inspection and (if necessary) repair of candidate bad links until no more bad links remain. This **exhaustive inspection** algorithm is described in Figure 6. In this scheme we assume an initial assignment of good and bad links. The first iteration starts with end-to-end measurements to which the SCFS algorithm is applied. Candidate bad links are then inspected and repaired if necessary. After this step, any inspected link k is known to be good; either the inspection found it good, or it was repaired and hence made good. Likewise all ancestors j of such a link k are also known to be good, since $Y_{f(k)} = X_{f(k)} = 1$ for all candidate links k. In the next iteration it remains only to determine the good/bad status of nodes descended from those just inspected. Iteration stops when no candidate bad links remain; at this point, all links were either good initially, or were bad and repaired.

5.2 Inspection Overhead: Computation

Let I denote the expected total number of inspections carried out using the exhaustive inspection algorithm. The expected number of bad links is $B = \sum_{k \in U} \overline{\alpha}_k$. The **inspection overhead** IO is the expected amount by which the number of inspections carried out in the exhaustive inspection algorithm exceeds the expected number of bad links, expressed as a fraction of the latter , i.e.,

$$\mathrm{IO} = (I - B)/B \tag{22}$$

We compute I as follows. First note that inspection and repair proceeds top down from the root node 0. Thus, after inspection and (if necessary) repair of node k, it is known that $X_k = 1$, even if $Y_k = 0$ before the repair. We let Q_k denote the event that $X_k = 1$ after inspection. Let η_k be the number of inspections carried out at link k and all links in the subtree below it. Define

$$L_k = \mathsf{E}[\eta_k | Q_{f(k)}, Y_k = 0] \quad M_k = \mathsf{E}[\eta_k | Q_{f(k)}] \quad N_k = \mathsf{E}[\eta_k | Y_{f(k)} = 1] \tag{23}$$

Since $X_0 = 1$ we have

$$I = M_1 \quad \text{and hence} \quad \mathrm{IO} = \frac{M_1}{\sum_{k \in U} \overline{\alpha}_k} - 1 \tag{24}$$

These quantities L, M, N obey the following recursions:

$$\begin{aligned} M_k &= \mathsf{E}[\eta_k | Q_{f(k)}] && (25)\\ &= \mathsf{P}[Y_k = 1 | X_{f(k)} = 1] \sum_{j \in d(k)} \mathsf{E}[\eta_j | Y_k = 1] && (26)\\ &\quad + \mathsf{P}[Y_k = 0, Z_k = 0 | X_{f(k)} = 1] \left(1 + \sum_{j \in d(k)} \mathsf{E}[\eta_j | Q_k] \right) && (27)\\ &\quad + \mathsf{P}[Y_k = 0, Z_k = 1 | X_{f(k)} = 1] \left(1 + \sum_{j \in d(k)} \mathsf{E}[\eta_j | Q_k, Y_j = 0] \right) && (28) \end{aligned}$$

Eq. (26) represents the case $Y_k = 1$: link k is not inspected. In (27) link k is inspected because $Y_k = 0$. Inspection reveals a bad link ($Z_k = 0$). The link is repaired (setting $Q_k = 1$). The links below k, i.e., the variables $\{Z_k : k \prec j\}$ are not constrained since $Y_k = 0$ is attributable to $Z_k = 0$. In (28), link k is inspected because $Y_k = 0$, but is found to be good ($Z_k = 1$): this was a false positive. This fact constrains $\{Z_k : k \preceq j\}$ because although the path to k is now known to be good, no path below k is good. Hence the subsequent inspections are conditioned on Q_k and $Y_k = 0$. The recursions for N_k and L_k follow in a similar manner:

$$\begin{aligned} N_k &= \mathsf{E}[\eta_k | Y_{f(k)} = 1] && (29)\\ &= \mathsf{P}[Y_k = 1 | Y_{f(k)} = 1] \sum_{j \in d(k)} \mathsf{E}[\eta_j | Y_k = 1] && (30)\\ &\quad + \mathsf{P}[Y_k = 0, Z_k = 0 | Y_{f(k)} = 1] \left(1 + \sum_{j \in d(k)} \mathsf{E}[\eta_j | Q_k] \right) && (31)\\ &\quad + \mathsf{P}[Y_k = 0, Z_k = 1 | Y_{f(k)} = 1] \left(1 + \sum_{j \in d(k)} \mathsf{E}[\eta_j | Q_k, Y_j = 0] \right) && (32) \end{aligned}$$

$$\begin{aligned} L_k &= \mathsf{E}[\eta_k | Q_{f(k)}, Y_k = 0] && (33)\\ &= 1 + \mathsf{P}[Z_k = 0 | X_{f(k)} = 1, Y_k = 0] \sum_{j \in d(k)} \mathsf{E}[\eta_j | Q_k] && (34)\\ &\quad + \mathsf{P}[Z_k = 1 | X_{f(k)} = 1, Y_k = 0] \sum_{j \in d(k)} \mathsf{E}[\eta_j | Q_k, Y_j = 0] && (35) \end{aligned}$$

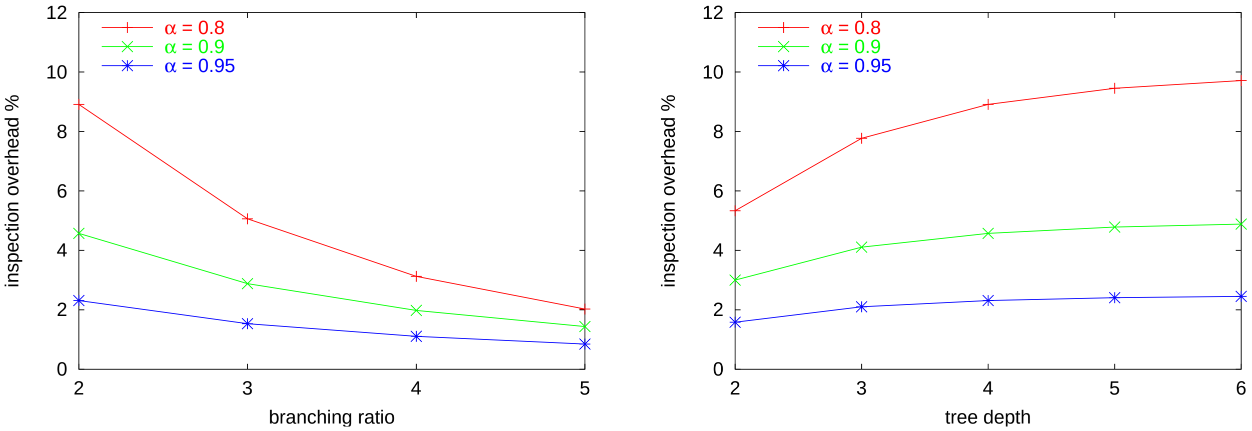

Figure 7: Inspection Overhead (as a percentage) for some uniform trees. Left: as a function of branching ratio. Right: as a function of tree depth.

Simplifying, and using the fact that $\beta_k = \gamma_k / A_{f(k)}$, we have

$$M_k = \overline{\beta}_k + \beta_k \sum_{j\in d(k)} N_j + \overline{\alpha}_k \sum_{j\in d(k)} M_j + (\alpha_k - \beta_k) \sum_{j\in d(k)} L_j \tag{36}$$

$$N_k = \frac{\overline{\beta}_k \gamma_{f(k),k}}{\gamma_{f(k)}} + \frac{\gamma_k}{\gamma_{f(k)}} \sum_{j\in d(k)} N_j + \frac{\overline{\alpha}_k \gamma_{f(k),k}}{\gamma_{f(k)}} \sum_{j\in d(k)} M_j + \frac{(\alpha_k - \beta_k)\gamma_{f(k),k}}{\gamma_{f(k)}} \sum_{j\in d(k)} L_j \tag{37}$$

$$L_k = 1 + \frac{\overline{\alpha}_k}{\overline{\beta}_k} \sum_{j\in d(k)} M_j + \left(1 - \frac{\overline{\alpha}_k}{\overline{\beta}_k}\right) \sum_{j\in d(k)} L_j \tag{38}$$

The edge conditions for $k \in R$ are

$$M_k = \overline{\alpha}_k, \quad N_k = \overline{\alpha}_k \gamma_{f(k),k} / \gamma_k, \quad L_k = 1 \tag{39}$$

5.3 Inspection Overhead: Performance and Conclusions

The inspection overhead for some uniform trees is computed and displayed in Figure 7. The left figure confirms that overhead decreases with branching ratio. This is to be expected from Theorem 6(v): as the branching ratio increases the false positive rate decreases since a given node the requires more (child) nodes in order to be falsely designated as bad. The right figure shows that overhead increases relatively slowly with tree depth. Note that in all cases the overhead is quite small, being less than 10% in the worst case considered, namely, a binary tree of depth 6 with bad links occurring with a 20% probability.

From the analysis of this section we conclude that iterative inspection of candidate bad links provides a very effective way of incorporating linkwise measurements (from inspection) into the network in order to locate all bad links. This holds even for trees that are deep and in which bad links are not vanishingly rare.

6 SCFS Performance: Weakly Separable Case

The analysis so far has been based on the assumption that statistics ϕ satisfy the separability assumption. How does the performance of the SCFS algorithm behave under violations of separability? In this section we consider the weakly separable case; in Section 2.2 we noted that weak separability can always be arranged through judicious choice in the definition of the good paths.

6.1 Critical Links

In the weakly separable case a good path $(0, k_1, \ldots, k_\ell)$ may have bad links, i.e., while $Z_{k_i} = 1,\ i = 1, \ldots, \ell$ implies $X_{k_\ell} = 1$, the converse is not true. We introduce a more refined notion of bad links, that of **critical links**, which helps us fill this gap. A link is critical if no path that contains it can be good. More precisely, let $\mathcal{P}_k$ denote the set of partial paths that pass through k, i.e.,

$$\mathcal{P}_k = \{(j_1, j_2, \ldots, j_m) :\ m = 1, 2, \ldots,\ j_i = f(k_{j+1}), i = 1, \ldots, m-1,\ k = j_i \text{ some } i\} \tag{40}$$

Since we work with a tree, each such path $p \in \mathcal{P}_k$ is determined by its end points, call them $s(p)$ and $e(p)$, where $s(p) \succ e(p)$. Let $X(p)$ be a random variable taking the value 1 if the path from $s(p)$ to $e(p)$ is good, while taking the value 0 if it is bad.

For each link k we define the set of critical ψ values $\Psi_k^{\text{crit}} \subset \Psi_k$ to be the set for which

$$\psi_k(\phi_k) \in \Psi_k^{\text{crit}} \Rightarrow \sup_{p \in \mathcal{P}_k} X(p) = 0 \tag{41}$$

Note that $\psi_k(\phi_k) \in \Psi_k^{\text{crit}}$ implies in particular that $X_k = 0$ and $Y_k = 0$. We will say that the link (or its parameter ϕ_k) is critical if $\psi_k(\phi_k) \in \Psi_k^{\text{crit}}$. Note that the event that k is critical is independent of the $\{Z_j :\ j \neq k\}$, since the implication is required to hold for all choices of these Z_j. Clearly, critical links in a weakly separable model are also bad, since if link k were both critical and good, then any path through k with all other links good would itself be good, in contradiction with $Y_k = 0$.

The point of this definition that if a bad link is, with high probability, critical, then bad links are very likely to cause bad paths. Thus, the analysis of Section 2.2 for the separable case will hold approximately. We define the **critical probability** for link k:

$$K_k = \mathsf{P}[k \text{ is critical } | k \text{ is bad }]. \tag{42}$$

One way to understand K_k is that $\overline{K}_k = 1 - K_k$ bounds the chance for a bad link to be part of a good path, since

$$\mathsf{P}[Y_k = 1 | Z_k = 0] \leq \mathsf{P}[\sup_{p \in \mathcal{P}_k} X(p) = 1 | Z_k = 0] \leq 1 - \mathsf{P}[k \text{ is critical } | Z_k = 0] = \overline{K}_k \tag{43}$$

Example: General Loss Model. We take a loss model with link transmission probabilities ϕ_k, and t^ℓ is the threshold transmission probability between good and bad paths of length ℓ. Consider a path comprising links $k_1, \ldots, k_\ell$. Link k_i is critical if $\prod_{j=1}^{\ell} \phi_{k_j} < t^\ell$ for any possible set of $\{\phi_{k_j} : k_j \neq k_i\}$. Thus k_i is critical if and only if $\phi_{k_i} < t^\ell$. Thus for any link k

$$K_k = \mathsf{P}[\phi_k < t^{\ell_{\max}(k)}]/\mathsf{P}[\phi_k < t] \tag{44}$$

where $\ell_{\max}(k)$ is the length of the longest path through k. Note that $1 - K_k$ is just the bound from (1) applied to the longest path length $\ell_{\max}(k)$. K_k depends of the probabilities of the good and bad states through the threshold t since $\alpha_k = \mathsf{P}[\phi_k \geq t]$.

Now a weakly separable model should approximate the strictly separable case well if K_k is close to one. From (44) we see this is likely to be the case if the mass of the conditional distribution of ϕ_k given k bad is distributed mostly away from the threshold t. We illustrate this with an example and a counterexample.

Example: $\boldsymbol{LM_2}$ model. Consider an LM_2 type model from Section 2.3, where good links i have transmission probabilities ϕ_k uniformly distributed in $(t, 1]$ and bad links have transmission probabilities ϕ_i uniformly distributed in $[0, t]$. According to (44),

$$K_k = t^{\ell_{\max}(k) - 1} \approx 1 - \bar{t}(\ell_{\max}(k) - 1) \tag{45}$$

Hence we expect a reasonable approximation to the separable case to require at least that $\bar{t}\ell_{\max}(k) \ll 1$.

Counterexample: Power Laws. Suppose that the distribution of link transmission rates is governed by the power law: $\mathsf{P}[\phi_k \leq x] = x^p$. Let $t \in (0, 1)$ be the boundary between good and bad transmission rates. Then

$$K_k = t^{p(\ell_{\max}(k) - 1)} \tag{46}$$

Thus K_k tends to be small when $p > 1$. This is because, in this case, the density of ϕ_k is increasing and hence the transmission rates of bad links are bunched towards the threshold t: the good and bad transmission rates are far from separable.

6.2 False Positive Rate and Detection Rate

To analyze the behavior of SCFS, we will compare the actual process X with the idealized process X' that would arise under strict separability with the same link probabilities $\{\alpha_k\}$. Thus

$$X_k \geq X'_k := \prod_{j \succeq k} Z_j \tag{47}$$

Similarly, we define $Y'_k = \max_{j \in R_k} X'_j$. We will also need $Y_{f(k),k} = \max_{j \in R_{f(k)} \setminus R_k} X_j$ and its separable analog $Y'_{f(k),k} = \max_{j \in R_{f(k)} \setminus R_k} X'_j$. In what follows, the quantities β_k, γ_k and $\gamma_{f(k),k}$ will refer to the quantities defined for the strictly separable analog.

Theorem 7 *In the weakly separable case,*

(i) $\mathrm{FPR}_k \leq (1 - \beta_k/\alpha_k)\gamma_{f(k),k} + \Theta_k$ *where* $\Theta_k = \sum_{j \succeq k} \overline{K}_j \overline{\alpha}_j / \alpha_k$

(ii) $C_1 = 1$ *and the detection rate is bounded below as* $\mathcal{C}_k \geq K_k \gamma_{f(k),k}$.

Comparing with Theorem 1 we see that Θ_k represents the increase in FPR_k that is attributable to departures from separability. When K_k is close to one, then clearly the bounds of Theorem 7 are close to the the exact expressions in the strictly separable case.

Example: $\boldsymbol{LM_2}$ model. Consider again the LM_2 type model from Section 2.3. Specifically, consider link probabilities $\alpha_k = \alpha$ and consider the regime with small $\overline{\alpha}$ and critical threshold hold t close to 1. Then

$$\mathrm{FPR}_k \approx (\overline{\alpha})^{\#d(k)} + \overline{\alpha}\,\overline{t}\,\ell_k^2 \tag{48}$$

$$\mathcal{C}_k \approx 1 - \ell_k\,\overline{\alpha} - (\overline{\alpha})^{\#d(k)-1} - \ell_k\,\overline{t} \tag{49}$$

Comparing with (19) and (20), observe that the last terms in each expression represent the contribution due to weak separability. Consider, as before, a binary tree ($\#d(k) = 2$) of depth 5, whose links have a probability $\alpha = 0.95$ to be good, meaning that their transmission rate exceeds the threshold $t = 0.99$. In the worst case ($\ell_k = 5$) we find $\mathrm{FPR}_k \approx 0.015$ (compared with 0.0025 in the strictly separable case) while $\mathcal{C}_k \approx 0.65$ (compared with 0.7). Although the effect on FPR is quite marked, it still remains at about 1% in this example.

Example: Generalized Loss Model with Measured Rates. Ideally, we would wish to determine the critical probability for a distribution of actual loss rates over a set of network links. Such data is not generally available. As a proxy we use the distribution of time-averaged loss rate measured across a set of internet paths in [25]. The data comprises 3,779 individual rates, each of which represents an average over one hour. Their cumulative distribution is displayed in Figure 8; note the logarithmic horizontal axis. From this graph, we can read off the dependence of the threshold (between the good and bad states) on the probability α to be in the good state: this is the cumulative probability for loss to attain at most the threshold value. For illustration, for a threshold loss of 1% (i.e. a threshold transmission rate 0.99) the α value is about 0.8.

Figure 9 shows the associated critical probability K (on the left) and bound Θ on the increase in the false positive probability (on the right) for link depths $\ell = 5, 15, 50$. We see that the critical probability exhibits a roughly U-shaped, with a minimum roughly at loss rates between 0.01 and 0.03, depending on link depth. In the depth 5 case considered previously, the minimum critical probability is about 0.2 found near the loss threshold 1% (i.e., the transmission threshold $t = 0.99$), rising to about 0.8 for loss rate 10^{-5}. The critical probabilities are small for deeper links.

The bound Θ is (mostly) decreasing as a function of loss threshold. At 1% loss threshold, Θ is about 0.19. For loss thresholds above about 1%, there is little dependence of Θ on the link depth. We can compare with the FPR in the strictly separable case as follows. First, for a given threshold loss threshold we read of the corresponding α value from the CDF in Figure 8. According to Theorem 3 we can bound the FPR in the strictly separable case by $1 - \beta^*(\alpha, r)/\alpha$, which has worst case branching ratio $r = 2$, leading to the bound $\mathrm{FPR} \leq (\overline{\alpha}/\alpha)^2$.

For loss threshold 1% we read off $\alpha \approx 0.8$ from Figure 8, leading to $\mathrm{FPR} \approx 0.06$ in the separable case; compare with $\Theta \approx 0.19$. For loss threshold 10% we read off $\alpha \approx 0.98$ from Figure 8, leading to $\mathrm{FPR} \approx 0.0004$ in the separable case; compare with $\Theta \approx 0.014$.

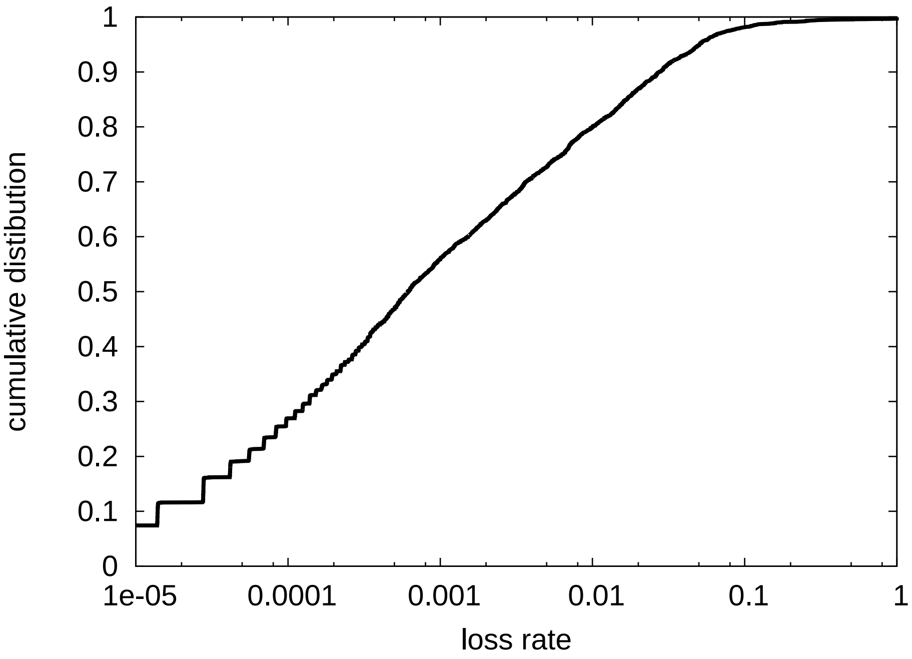

Figure 8: Experimental Path Loss Rate: Cumulative Distribution Function

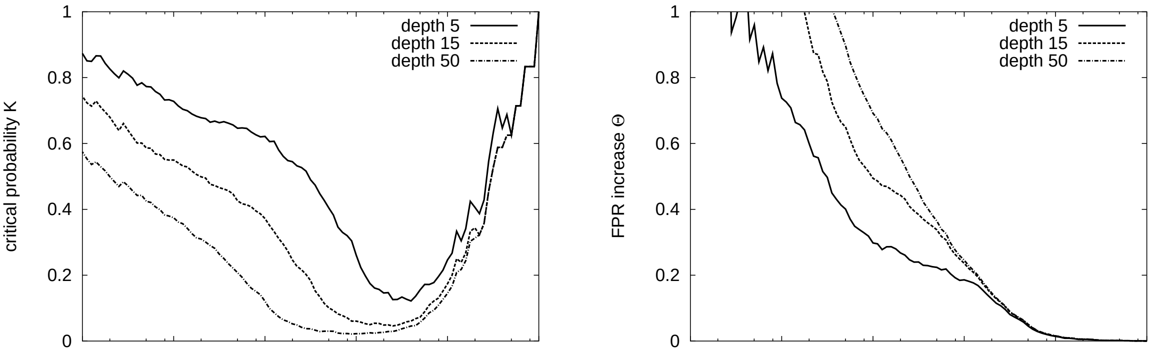

Figure 9: Critical Probability K and FPR increase Θ as a function of loss threshold, assuming the link loss rate distribution of Figure 8, for link depths $\ell = 5, 15, 50$

α	F	C	C^0
0.95	0.0%—0.08%	59%—81%	95%
0.9	0.0%—0.22%	35%—66%	90%
0.8	0.0%—0.34%	11%—42%	80%

Table 2: Approx. 1000 node tree. False Positive Rate F, detection rate C, and detection rate C^0 under constant path failure scaling, as function of fraction α of good links.

7 Performance Comparisons with Other Methods

Model Comparisons. We first describe some comparisons of our model analysis with the simulation results for three methods (random, linear programming and Gibbs) applied to the LM 1 model in [18]. Both our analysis and the simulations used topologies of 1,000 nodes with maximum branching ratio 10. Our model calculations ran over 10 topologies containing roughly 1,000 nodes with depth between 3 and 10. This set included uniform topologies with branching ratio 2 and 10, namely $T_\alpha(2,2,2,2,2,2,2,2,2,2)$ and $T_\alpha(10,10,10)$ respectively.

Table 2 shows the range false positive rates F and detection rate C for SCFS over all model calculations . Comparing with Figure 3 of [18] it appears that for $\alpha = 0.95$, the SCFS false positive rate is at least as good as any method simulated, while the detection rate is as good as the linear programming and Gibbs methods. (The “random” method from [18] had higher detection rate but a very high false positive rate). However, comparing with Figure 4 of [18], it is evident that while the false positive rate of SCFS remains quite small (less than 1%), as α decreases, the detection rate falls off steeply compared with the other methods.

Simulation Comparisons. Partly in response to [11], a recent paper [3] reported network level simulations of SCFS, the random and Gibbs algorithms from [18], and a new proposed algorithm COBALT. The model had somewhat separated loss regimes (good links had losses from 0 to 0.5%), bad links had losses between 1% and 3%. In our terminology, this would only be strictly separable for trees with maximum depth 2; otherwise it would be weakly separable. The network topologies were generated using the BRITE two-level hierarchical model [4] and comprised 800 nodes connect by 1,400 links. `ns-2` was used to simulate the download of large files from a server by 100 randomly chosen clients. In this context, bad links are those whose loss rate exceeds a given level. In the comparison, SCFS was found to have the lowest false positive rate, being about one third the rate of the next best algorithm. The detection rate was as good as other methods when bad links were rare (5% of links were bad) but fell off when they were more common (20% of links were bad). But even for the other methods, the detection rate was far from perfect, being around 65% for the range of α considered (0.95, 0.9 and 0.8).

Computational Aspects. We expect our algorithm to be less complex than LP and far less complex than the general most accurate method presented, Gibbs Sampling. In [3], running times for SCFS and COBALT were found to be an order of magnitude less than for Gibbs.

Constant Path Failure Rate Scaling. Table 2 also shows the detection rate C^0 in the constant path failure scaling. This is barely sensitive to topology, and approximately equal to the proportion of good links.

Discussion. To some extent, the set of algorithms studied here and in other papers [18] and [3] exhibit a trade-off between false positives and false negatives (as represented by imperfect detection rate). Any choice between algorithms must take account of the relative costs of false positives and negatives in the target application. We believe that in the networking context the false positive cost is typically quite high due to the administrative costs of inspecting potentially faulty components. This favors a scheme with a low false positive rate.

8 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 can be effective in identifying candidate bad links on a tree from end-to-end measurements. The algorithm, which attributes path failure to the smallest set of consistent link failures, is justified by the observation that when bad links are uncommon, two or more badly performing intersecting paths likely have a bad link in their intersection. Moreover, the likelihood for this to happen is relatively insensitive to changes in the fraction of bad links if this fraction is small. Conversely, the false positive rate is very low in this regime because it is very rare that a good link will lie at the head of a maximal bad subtree.

On the negative side, with single execution of SCFS the detection rate for bad links is less than unity. We regard this as the price paid for using uncorrelated measurements. Previous work on tomography used measurements correlated at the packet level, and estimators of link loss rates and packet latencies were unbiased under the same packet and linkwise independence conditions that we assumed in our setup in Section 2. Thus, misidentification of bad links only occurred due to statistical variability of the estimators and vanished as the number of probes grew. In the SCFS approach, we can, in fact, achieve a unit detection rate by iterating the SCFS approach with limited link inspection. For some applications, this need not be regarded as a deficiency. Suppose the cost to “repair” bad links, i.e., to make them good, is high. Depending on context, repair may entail replacing a bad component or rerouting traffic away from it. The overhead in repeated inspection is small if the false positive rate is low. This motivated the Exhaustive Inspection algorithm of Section 5.

In this paper we assume that the ambient failure probabilities are known a priori; these determine the boundary between good and bad states. Another potential approach is to adaptively set the threshold between good and bad based on clustering properties of the measured end-to-end path characteristics. The difficulty with this approach is that there may be trivial clustering properties even when bad links occur. A clear example is when the first link next to the root suffers heavy performance impairment but all others have no impairment. In this case, the measured end-to-end properties will appear the same at all leaves, so clustering does not help in setting the boundary between good and bad performance.

We now outline some generalizations of the present work that we would like to investigate in the future.

Enlarged State Space. A natural generalization of our work is to increase the size of the state space beyond the two states in the good/bad classification. One benefit of this would be to further reduce the false positive probability. Consider a good node k with two children, each of which lie at the head of a maximally bad subtree. In the two state classification, the good node k is classified as bad. However, if the bad state is split into substates, then a separation between the path states measured on each subtree would most likely indicate separate causes of badness, rather than a common cause in badness of link k. A downside of enlarging the state space is a more complex relation between link and path states.

General Network Topologies. To extend the method from tree topologies to general network topologies, we can take the approach of [5] and cover a network with a set of trees, and conduct measurements on each tree. An obvious approach is to infer on each tree independently using the methods of this paper. Exhaustive inspection in the manner of Section 5 can share information on known good links amongst the different inference problems which may potentially reduce the number of iterations needed to render all links good.

Inference from Measurement Time Series. Suppose now that rather than being static, the good/bad status of each link can fluctuate over time. This corresponds to the Gilbert model of [18]. Consider the delay spike example of Section 2.3. Divide time into consecutive intervals $\{S_t : t = 1, 2, \ldots\}$ of equal duration. If a delay spike of sufficient size occurs on a link during interval S_t, then that link is bad for that interval. Note that the probability α_i for link i to be good during an interval is a nonincreasing function of the interval duration. In the stationary case, α_i is proportional to the duration of the measurement interval. Let $Z_{i,t}$ be the good/bad indicators for link i in interval t, and define the path indicator $X_{i,t} = \prod_{j \succeq i} Z_{i,t}$ accordingly. Suppose we now assume that:

(i) The tree topology is the same for all measurement intervals

(ii) The delay spike model is separable

(iii) Delay spikes are independent over different links

(iv) Delay spikes are independent over different time intervals

Under these assumptions it is evident that the link probabilities α_i can be inferred from the timeseries $\{X_{i,t} : i \in R,\ t = 1, 2, \ldots\}$ of path good/bad status as measured at the leaf node, using the methods of [6]. We remark that the topology itself can also be inferred from the same measurements. We defer to another paper a study of the effectiveness of this method and its behavior under weak separability.

Experimental Evaluation. Much of the work of this paper has been devoted to performance analysis of the SCFS algorithm and its iterative generalization. An important and complementary approach will be to evaluate performance under representative network topologies and patterns of performance degradation, as determined from network measurements. In particular, it is desirable to quantify the trade-off in practice between reducing network measurement complexity (as compared with packet-level correlated measurements) and increasing false positives and negatives for the detection of bad links.

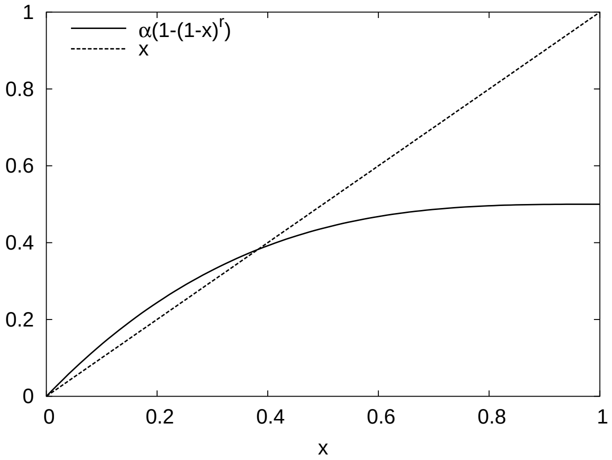

Figure 10: Illustration of $B_{\alpha,r}(x) = \alpha(1-(1-x)^r)$ in case where $\alpha r > 1$: $r = 3$ and $\alpha = 1/2$. Intersection with function x in $(0, 1)$ is fixed point $\beta^*(\alpha, r)$

Appendix: Proofs of Theorems

Proof of Theorem 1

$$\begin{aligned} \mathrm{FPR}_1 &= \mathsf{P}[Y_1 = 0|Z_1 = 1] = \mathsf{P}[Y_1 = 0, Z_1 = 1]/\mathsf{P}[Z_1 = 1] && (50)\\ &= (\mathsf{P}[Y_1 = 0] - \mathsf{P}[Z_1 = 0])/\mathsf{P}[Z_1 = 1] = 1 - \beta_1/\alpha_1 && (51) \end{aligned}$$

For all other k:

$$\begin{aligned} \mathrm{FPR}_k &= \mathsf{P}[Y_k = 0, Y_{f(k)} = 1|Z_k = 1] && (52)\\ &= \mathsf{P}[Y_k = 0, Y_{f(k)} = 1, Z_k = 1]/\mathsf{P}[Z_k = 1] && (53)\\ &= (\mathsf{P}[Y_k = 0, Y_{f(k)} = 1] - \mathsf{P}[Y_k = 0, Y_{f(k)} = 1, Z_k = 0])/\mathsf{P}[Z_k = 1] && (54)\\ &= (\mathsf{P}[Y_k = 0, Y_{f(k),k} = 1] - \mathsf{P}[Z_k = 0, Y_{f(k),k} = 1])/\mathsf{P}[Z_k = 1] && (55)\\ &= (\mathsf{P}[Y_k = 0|X_{f(k)} = 1] - \mathsf{P}[Z_k = 0|Y_{f(k),k} = 1])\mathsf{P}[Y_{f(k),k} = 1]/\mathsf{P}[Z_k = 1] && (56)\\ &= (1 - \beta_k/\alpha_k)\gamma_{f(k),k} \quad \blacksquare && (57) \end{aligned}$$

Proof of Theorem 2 : ***(i,ii)*** For $r \geq 2$ and $\alpha > 0$, $B_{\alpha,r}$ is a strictly concave increasing function from $[0, 1]$ onto $[0, \alpha]$, with $B'_{\alpha,r}(0) = \alpha r$ and $B'_{\alpha,r}(1) = 0$. Hence the equation $B_{\alpha,r}(\beta) = \beta$ is a solution β^* in $(0, 1)$ if and only if $r\alpha > 1$, and this solution is unique in $(0, 1)$. Otherwise there is only one solution, namely, $\beta^* = 0$. See Figure 10 for $r = 3$, $\alpha = 1/2$.

(iii,iv) $B_{\alpha,r}(\alpha) < \alpha$. Since $B_{\alpha,r}$ is increasing, $B^{\circ(n+1)}_{\alpha,r}(\alpha) < B^{\circ n}_{\alpha,r}(\alpha)$. (Here $B^{\circ n}$ denotes an n-fold composition). Hence the sequence $\{\beta^{(n)}\}$ is decreasing. Since $B_{\alpha,r}$ is continuous on $[0, 1]$ and bounded, the sequence is bounded, and hence convergent to some limit $\check{\beta}$ by the monotone convergence theorem. Since $B_{\alpha,r}$ is continuous, $\check{\beta}$ must be a fixed point of $B_{\alpha,r}$. When $\alpha r > 1$, this fixed point cannot be 0, since

$B_{\alpha,r}(0) = 0$ and $B'_{\alpha,r}(0) = \alpha r > 1$ implies $B_{\alpha,r}(x) > x$ in some neighborhood $(0, \varepsilon)$. Thus $\{\beta^{(n)}\}$ cannot be a decreasing sequence converging to 0.

(v) Suppose $B^{\circ n}_{\alpha,r}(\alpha) \geq B^{\circ n}_{\alpha,r'}(\alpha)$. (This is trivially true for the starting value $B^{\circ 0}_{\alpha,r}(\alpha) = \alpha$. Then

$$B^{\circ(n+1)}_{\alpha,r}(\alpha) = B_{\alpha,r}(B^{\circ n}_{\alpha,r}(\alpha)) \geq B_{\alpha,r'}(B^{\circ n}_{\alpha,r}(\alpha)) \geq B_{\alpha,r'}(B^{\circ n}_{\alpha,r'}(\alpha)) = B^{\circ(n+1)}_{\alpha,r'}(\alpha) \tag{58}$$

since $B_{\alpha,r}(x)$ is increasing in both α and r. The result follows from (iv) on taking the limit $n \to \infty$.

(vi) Suppose $B^{\circ n}_{\alpha,r}(\alpha) \geq B^{\circ n}_{\alpha',r}(\alpha')$. (This is trivially true for the starting value $B^{\circ 0}_{\alpha,r}(\alpha) = \alpha$. Then

$$B^{\circ(n+1)}_{\alpha,r}(\alpha) = B_{\alpha,r}(B^{\circ n}_{\alpha,r}(\alpha)) \geq B_{\alpha',r}(B^{\circ n}_{\alpha,r}(\alpha)) \geq B_{\alpha',r}(B^{\circ n}_{\alpha',r}(\alpha')) = B^{\circ(n+1)}_{\alpha',r}(\alpha') \tag{59}$$

since $B_{\alpha,r}(x)$ is increasing in both α and r. The result follows from (iv) on taking the limit $n \to \infty$. ∎

Proof of Theorem 3 : *(i)* By Theorem 2(iii), $\beta^*(\alpha, r) < \beta_{f(k)} < \beta_k$, and hence $\mathrm{FPR}_k \leq 1 - \beta^*(\alpha, r)/\alpha$. By Theorem 2(v), the greatest upper bound is obtained for $r = 2$. In this case $\beta^*(\alpha, 2) = \max\{0, 1 - \overline{\alpha}/\alpha\}$, and hence $F(\alpha, 2) = \min\{1, (\overline{\alpha}/\alpha)^2\}$.

(ii) Suppose $j \preceq k$ with $j \notin R$. Then

$$\begin{aligned} \beta_j &= \alpha_j(1 - \prod_{i \in d(j)} (1 - \beta_i)) && (60)\\ &\geq \alpha_k^{\min}(1 - (1 - \min_{i \in d(j)} \beta_i)^{\# d(j)}) && (61)\\ &\geq \alpha_k^{\min}(1 - (1 - \min_{i \in d(j)} \beta_i)^{r_k^{\min}}) && (62)\\ &= B_{\alpha_k^{\min}, r_k^{\min}}(\min_{i \in d(j)} \beta_i) && (63) \end{aligned}$$

For a leaf node i, $\beta_i = \alpha_i \geq \alpha_k^{\min} \geq \beta^*(\alpha_k^{\min}, r_k^{\min})$. We now proceed by induction. Suppose $\beta_i > \beta^*(\alpha_k^{\min}, r_k^{\min})$ for all $i \in d(j)$. Since $B_{\alpha,r}(\cdot)$ is increasing, (60) implies that

$$\beta_j \geq B_{\alpha_k^{\min}, r_k^{\min}}(\beta^*(\alpha_k^{\min}, r_k^{\min})) = \beta^*(\alpha_k^{\min}, r_k^{\min}). \tag{64}$$

The bound on FPR_k then follows from Theorem 1. ∎

Proof of Theorem 4 : From Theorem 2(v) and (9), when $\overline{\alpha} < 1/2$ then $\beta^*(\alpha, r) \geq \beta^*(\alpha, 2) = 1 - \overline{\alpha}/\alpha$. Thus $\beta^*(\alpha, r) = \alpha(1 - (1 - \beta^*(\alpha, r)^r) \geq \alpha(1 - (\overline{\alpha}/\alpha)^r) \geq \alpha - \overline{\alpha}^r \alpha^{1-r}$. This establishes (i), and (ii) follows easily from the definition of $F(\alpha, r)$ in (10).∎

Proof of Theorem 5 (i)

$$\begin{aligned} \mathsf{P}[Y_k = 0, Y_{f(k)} = 1 | Z_k = 0] &= \mathsf{P}[Y_k = 0, Y_{f(k),k} = 1 | Z_k = 0] && (65)\\ &= \mathsf{P}[Y_{f(k),k} = 1 | Z_k = 0] \qquad (Z_k = 0 \Rightarrow Y_k = 0) && (66)\\ &= \mathsf{P}[Y_{f(k),k} = 1] \qquad (Y_{f(k),k}, Z_k \text{ are independent}) && (67) \end{aligned}$$

(ii)

$$\begin{aligned}
\gamma_{f(k),k} &= \mathsf{P}[X_{f(k)} = 1] \left(1 - \prod_{j\in d(f(k))\setminus\{k\}} \mathsf{P}[Y_j = 0|X_{f(k)} = 1]\right) && (68)\\
&= A_{f(k)} \left(1 - \prod_{j\in d(f(k))\setminus\{k\}} \overline{\beta}_j\right) && (69)\\
&= A_{f^2(k)}(\beta_{f(k)} - \alpha_{f(k)}\beta_k)/\overline{\beta}_k \quad \text{(by (7))} && (70)
\end{aligned}$$

(iii,iv) Applying the bound on β_j from Theorem 3(ii) to (69)

$$\begin{aligned}
\gamma_{f(k),k} &\geq A_{f(k)} \left(1 - \overline{\beta}^*(\alpha^{\min}_{f(k)}, r^{\min}_{f(k)})^{\#d(f(k))-1}\right) && (71)\\
&\geq A_{f(k)} \left(1 - \overline{\beta}^*(\alpha^{\min}_{f(k)}, r^{\min}_{f(k)})^{r^{\min}_{f(k)}-1}\right) && (72)\\
&= A_{f(k)}(1/\alpha^{\min}_{f(k)} - 1)/(1/\beta^{\min}_{f(k)} - 1) && (73)
\end{aligned}$$

This establishes the first inequality in (iv), of which (iii) is a special case. The second inequality in (iv) follows from Theorem 2(iii) using $\beta(\alpha, 2) = \max\{0, 1 - \overline{\alpha}/\alpha\}$. ∎

Proof of Theorem 6: (i) When $k \in R$, $\beta_k = \alpha_k$. When $k = f(j)$ for some $j \in R$, then $\beta_k = \alpha_k(1 - T_k)$. The remaining cases we prove by induction. Suppose (i) holds for all $j \in d(k)$. Then

$$\begin{aligned}
\beta_k &= \alpha_k \left(1 - \prod_{j\in d(k)} (\overline{\alpha}_j + T_j(1 + O(\overline{\alpha}_-)))\right) && (74)\\
&= \alpha_k(1 - T_k \prod_{j\in d(k)} (1 + (T_j/\overline{\alpha}_j)(1 + O(\overline{\alpha}_-)))) && (75)\\
&= \alpha_k(1 - T_k(1 + O(\overline{\alpha}_-))) = \alpha_k - T_k(1 + O(\overline{\alpha}_-)) && (76)
\end{aligned}$$

since $T_j/\overline{\alpha}_j \leq (\overline{\alpha}_-)^2/\overline{\alpha}_+$ and $\overline{\alpha}_-/\overline{\alpha}_+$ is bounded.

(ii) Similarly to the proof of (i),

$$\begin{aligned}
\gamma_{f(k),k} &= A_{f(k)}(1 - \prod_{j\in d(f(k))\setminus\{k\}} \beta_j) && (77)\\
&= (1 - S_{f(k)} + O(\overline{\alpha}_-^2)) \left(1 - \prod_{j\in d(f(k))\setminus\{k\}} (\overline{\alpha}_j + T_j(1 + O(\overline{\alpha}_-)))\right) && (78)\\
&= 1 - S_{f(k)} - T_{f(k)}/\overline{\alpha}_k + O(\overline{\alpha}_-^2) && (79)
\end{aligned}$$

(iii) follows from (i), (ii) and Theorem 1.

(iv) follows from (ii) and (13).

(v) From (iv) we see that $1 - \mathcal{C}$ is $O(\overline{\alpha}_-)$ and hence $\mathcal{C} \to 1$ in the limit. The lower bound on $\mathcal{C}$ follows because $S_{f(k)} \leq (\ell_k - 1)\overline{\alpha}_- \leq (\ell - 1)\overline{\alpha}_-$, while $T_k \leq \overline{\alpha}_-$.

(vi). In the uniform case $\alpha_k = \alpha$, $S_k = \overline{\alpha}\ell_k$ while $T_k = \overline{\alpha}^{\#d(k)}$. The latter's contribution to $(1-\mathcal{C})/\overline{\alpha}$ is then $\overline{\alpha}^{\#d(k)-2}$, yielding 1 in the limit if $\#d(k) = 2$, and 0 otherwise.

The forms in (vii) then follow by summation: for $r \neq 2$ we have $\sum_{i=1}^{\ell}(i-1)r^{i-1}/\sum_{i=1}^{\ell} r^{i-1}$. For $r = 2$, we take this sum and add $\sum_{i=2}^{\ell} r^{i-1}/\sum_{i=1}^{\ell} r^{i-1}$. ∎

Proof of Theorem 7 : The bounds established previously would still hold were X, Y to be replaced by $X'Y'$. Our strategy is to find out how closely these bounds hold. We now derive bounds for the various terms in (54). We first bound $\mathsf{P}[Y_k = 0, Y_{f(k)} = 1, Z_k = 0]$ below. Since $\{Y_k = 0, Y_{f(k)} = 1\} = \{Y_k = 0, Y_{f(k),k} = 1\}$,

$$\begin{aligned}
\mathsf{P}[Z_k = 0, Y_k = 0, Y_{f(k),k} = 1] &= \mathsf{P}[Y_k = 0 | Z_k = 0, Y_{f(k),k} = 1]\mathsf{P}[Z_k = 0 | Y_{f(k),k} = 1]\mathsf{P}[Y_{f(k),k} = 1] \\
&\geq \mathsf{P}[k \text{ is critical } | Z_k = 0, Y_{f(k)} = 1]\overline{\alpha}_k\mathsf{P}[Y_{f(k),k} = 1] \\
&= K_k\overline{\alpha}_k\mathsf{P}[Y_{f(k),k} = 1] \qquad (80)
\end{aligned}$$

Here the inequality follows from (41) since k critical implies $Y_k = 0$ (for a weakly separable model) while Z_k is independent of $Y_{f(k),k} = 1$, and the final inequality follows since criticality of k is independent of $\{\phi_j : j \neq k\}$.

We next bound $\mathsf{P}[Y_k = 0, Y_{f(k)} = 1]$ above. Since $X \geq X'$ and $Y \geq Y'$,

$$\begin{aligned}
\mathsf{P}[Y_k = 0, Y_{f(k)} = 1] &= \mathsf{P}[Y_k = 0, Y_{f(k),k} = 1] \\
&\leq \mathsf{P}[Y'_k = 0, Y_{f(k),k} = 1] \\
&= \mathsf{P}[Y'_k = 0, X'_{f(k)} = 1, Y_{f(k),k} = 1] + \mathsf{P}[X'_{f(k)} = 0, Y_{f(k),k} = 1] \qquad (81)
\end{aligned}$$

The first term in (81) is equal to

$$\begin{aligned}
\mathsf{P}[Y'_k = 0 | X'_{f(k)} = 1, Y_{f(k),k} = 1]\mathsf{P}[X'_{f(k)} = 1, Y_{f(k),k} = 1] &\leq \mathsf{P}[Y'_k = 0 | X'_{f(k)} = 1]P[Y_{f(k),k} = 1] \\
&= \overline{\beta}_k P[Y_{f(k),k} = 1] \qquad (82)
\end{aligned}$$

The second term in (81) is bounded above by $\sum_{j \succ k}\mathsf{P}[Y_{f(k),k} = 1 | Z_j = 0]\overline{\alpha}_j \leq \sum_{j \succ k}\overline{K}_j\overline{\alpha}_j$. Combining with (82) we obtain the upper bound sought:

$$\mathsf{P}[Y_k = 0, Y_{f(k)} = 1] \leq \overline{\beta}_k\mathsf{P}[Y_{f(k),k} = 1] + \sum_{j \succ k}\overline{K}_j\overline{\alpha}_j. \qquad (83)$$

The upper bound on FPR_k now follows by inserting (80) and (83) into (54):

$$\alpha_k\mathrm{FPR}_k \leq \left(\overline{\beta}_k - K_k\overline{\alpha}_k\right)\mathsf{P}[Y_{f(k),k} = 1] + \sum_{j \succ k}\overline{K}_j\overline{\alpha}_j \qquad (84)$$

Since $\beta_k \leq \alpha_k$ and $K_k \leq 1$, the term in parenthesis in (84) is non-negative and hence

$$\alpha_k\mathrm{FPR}_k \leq (\alpha_k - \beta_k)\gamma_{f(k),k} + \sum_{j \succeq k}\overline{K}_j\overline{\alpha}_j \qquad (85)$$

(ii) For $k = 1$, the detection rate is $C_1 = 1$. Otherwise the detection rate is

$$\begin{aligned}
\mathcal{C}_k &= P[Y_k = 0, Y_{f(k)} = 1 | Z_k = 0] = \mathsf{P}[Y_k = 0, Y_{f(k),k} = 1, Z_k = 0]/\mathsf{P}[Z_k = 0] \qquad (86) \\
&\geq K_k\mathsf{P}[Y_{f(k),k}] \geq K_k\gamma_{f(k),k} \quad ∎ \qquad (87)
\end{aligned}$$

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] D. Arifler, G. de Veciana, and B. L. Evans, “Network Tomography Based on Flow Level Measurements”, Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Proc., Montreal, Canada, May 17-21, 2004.

[3] A. Batsakis, T. Malik, A. Terzis “Practical Passive Lossy Link Inference”, PAM 2005 (Passive and Active Measurement Workshop), Boston, MA, MArch 31-April 01, 2005.

[4] BRITE: Boston university Representative Internet Topology gEnerator. See: http://www.cs.bu.edu/brite/

[5] T. Bu, N. Duffield, F. Lo Presti, D. Towsley, “Network tomography on general topologies”, Proceedings ACM Sigmetrics 2002, Marina Del Rey, CA, June 15-19, 2002.

[6] R. Cáceres, N.G. Duffield, J.Horowitz D. Towsley. “Multicast-Based Inference of Network Internal Loss Characteristics,” *IEEE Trans. on Information Theory*, **45**(7), 2462-2480, 1999.

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

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

[9] M. Coates, R. Nowak. “Network loss inference using unicast end-to-end measurement, *Proc. ITC Conf. IP Traffic, Modeling and Management*, Sept. 2000.

[10] Mark Coates, Rui Castro, Robert Nowak, Manik Gadhiok, Ryan King and Yolanda Tsang, “Maximum Likelihood Network Topology Identification from Edge-Based Unicast Measurements”, ACM Sigmetric 2002, Marina Del Rey, California, June 2002.

[11] N.G. Duffield, “Simple Network Performance Tomography”, ACM SIGCOMM Internet Measurement Conference 2003, Miami Beach, Fl, October 27-29, 2003

[12] N.G. Duffield, V. Arya, R. Bellino, T. Friedman, J. Horowitz, T. Turletti, D. Towsley “Network Tomography from Aggregate Loss Reports”, Proc Peformance 2005, Juan-les-Pins, France, Octoer 3-7, 2005.

[13] N.G. Duffield, J. Horowitz, F. Lo Presti, D. Towsley, “Multicast Topology Inference from Measured End-to-End Loss”, IEEE Trans. on Information Theory, vol. 48, pp. 26–45, 2002.

[14] N.G. Duffield, F. Lo Presti, V. Paxson, D. Towsley, “Inferring link loss using striped unicast probes,” in Proc. IEEE Infocom 2001, Anchorage, Alaska, April 22-26, 2001.

[15] F. Lo Presti, N.G. Duffield, J. Horowitz, and D. Towsley. Multicast-based inference of network-internal delay distributions. *IEEE/ACM Trans. Netw.*, 10(6):761–775, 2002.

[16] Network Reliability Council (NRC) Reliability Issues - Changing Technologies Focus Group, “New Wireline Access Technologies Subteam Final Report”, February 22, 1996. See: http://www.nric.org/pubs/nric2/fg3/4nwat.pdf

[17] “Packet Wingspan Distribution”, NLANR. See http://www.nlanr.net/NA/Learn/wingspan.html

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

[19] Anoop Reddy, Deborah Estrin, Ramesh Govindan, “Fault Isolation in Multicast Trees”, ACM SIGCOMM, Stockholm, Sweden, August 2000.

[20] Yolanda Tsang, Mark Coates and Robert Nowak, “Passive Unicast Network Tomography using EM Algorithms”, IEEE International Conference on Acoustics, Speech, and Signal Processing, Salt Lake City, Utah, May 2001, Volume 3, pp. 1469-1472.

[21] Yolanda Tsang, Mark Coates and Robert Nowak, “Nonparametric Internet Tomography”, IEEE International Conference on Acoustics, Speech, and Signal Processing, Orlando, Florida, May 2002, Volume 2, pp. 2045-2048.

[22] Yolanda Tsang, Mark Coates and Robert Nowak, “Network Delay Tomography”, IEEE Transaction of Signal Processing in Networking, Aug. 2003, Volume 51, Issue 8, pp. 2125-2136.

[23] Yolanda Tsang, Mehmet Yildiz, Robert Nowak and Paul Barford, “Network Radar: Tomography from Round Trip Time Measurement”, ACM Internet Measurement Conference, October, 2004, Taormina, Sicily, Italy, pp. 175-180.

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

[25] 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.