Network Tomography on Correlated Links

Denisa Ghita, Katerina Argyraki, Patrick Thiran
School of Computer and Communication Sciences
École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

ABSTRACT

Network tomography establishes linear relationships between the characteristics of individual links and those of end-to-end paths. It has been proved that these relationships can be used to infer the characteristics of links from end-to-end measurements, provided that links are not correlated, i.e., the status of one link is independent from the status of other links.

In this paper, we consider the problem of identifying link characteristics from end-to-end measurements when links are "correlated," i.e., the status of one link may depend on the status of other links. There are several practical scenarios in which this can happen; for instance, if we know the network topology at the IP-link or at the domain-link level, then links from the same local-area network or the same administrative domain are potentially correlated, since they may be sharing physical links, network equipment, even management processes.

We formally prove that, under certain well defined conditions, network tomography works when links are correlated, in particular, it is possible to identify the probability that each link is congested from end-to-end measurements. We also present a practical algorithm that computes these probabilities. We evaluate our algorithm through extensive simulations and show that it is accurate in a variety of realistic congestion scenarios.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations—*network monitoring*

General Terms

Algorithms, Measurement, Performance

Keywords

network performance tomography, link correlation

IMC'10, November 1–3, 2010, Melbourne, Australia.
Copyright 2010 ACM 978-1-4503-0057-5/10/11 ...$10.00.

1. INTRODUCTION

Network performance tomography infers the characteristics of network links from end-to-end path measurements. Such inference can be a powerful monitoring tool in situations where explicitly measuring the characteristics of each link is impractical or is not an option. For instance, consider the scenario where the operator of an Internet service provider (ISP) wants to estimate the quality of service offered by a peer. In this case, the operator needs to estimate the quality of the peer's links without having direct access to these links. Alternatively, consider the scenario where an operator wants to monitor the quality of a link in her own domain. With conventional network equipment, this can be done with the help of test traffic, e.g., traceroute probes or custom probes explicitly exchanged between the two end-points of the link; such probes, however, are typically generated and processed by the general-purpose processor that handles the control plane of the corresponding network equipment (as opposed to the specialized hardware that handles the data plane), hence, can meet a different fate (e.g., significantly different latency) than the one met by the rest of the traffic.

Network performance tomography formulates the problem of inferring link characteristics from end-to-end path measurements as a system of linear equations, where the known entities are the available path measurements and the network topology, while the link characteristics constitute the unknowns. There exist different tomographic algorithms, depending on which link characteristics we are interested in (e.g., latency, loss, congestion status) and how we conduct the end-to-end measurements (e.g., using multicast probes, packet trains, or normal traffic). Existing algorithms, however, form their linear equations assuming that links are not correlated, i.e., that the latency, loss, or congestion status of one link is independent from that of other links.

There exist at least two practical scenarios in which ignoring link correlation is not reasonable. (i) Consider the scenario where an operator uses network tomography to monitor the quality of links in her domain in a non-intrusive manner and relies on traceroute to discover the domain's topology. As a result, she misses all nodes that do not respond to traceroute, necessarily including all network elements operating below layer 3. In the resulting network graph, nodes represent layer-3 elements; hence, links between nodes located in the same local-area network are potentially correlated, because they may be sharing physical links. (ii) Consider the scenario where the operator of one administrative domain wants to monitor a set of neighboring domains at the granularity of domain-level (as opposed to physical or IP-level) links; this may be because the operator does not have visibility into the internals of other domains, e.g., because they use multi-protocol label switching (MPLS) for internal routing; it may also be because the operator does not *care* to have such visibility—she is only trying to determine whether the neighbors are honoring a service-level agreement (SLA). In the resulting network graph, intermediate nodes represent border routers; hence, links between nodes located in the same administrative domain are potentially correlated, because they may be sharing physical links, as well as management processes.

In this paper, we take the first step toward applying network tomography on correlated links. We consider a network model where we know the network topology and which links are potentially correlated, but not to what extent (i.e., we do not assume that we know the degree of correlation between links). For instance, we can use our model to describe the fact that all links that are located in the same local-area network and/or all links that are managed by the same administrative entity are potentially correlated. Assuming this model and that we can perform unicast end-to-end path measurements, we seek to characterize the congestion behavior of each link, in particular, how frequently each link is congested (we formally define "congestion" in Section 2).

We make two contributions. First, we formally prove that, under certain well defined conditions, by using end-to-end measurements, it is feasible to identify, for each link, the probability that the link is congested, even in the presence of correlated links (Section 3). Second, we show how to actually compute these probabilities, i.e., we present an algorithm that takes as input (i) the network topology, (ii) which links may be correlated, and (iii) end-to-end path measurements, and outputs, for each link, the probability that the link is congested (Section 4). We evaluate our algorithm through extensive simulations and show that it is accurate in a variety of congestion scenarios (Section 5).

2. SETUP

2.1 Network Model

Links and Paths. We model the network as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each node $v_i \in \mathcal{V}$ represents a network element that generates, receives, and/or relays network traffic, e.g., an end-host, an Ethernet switch, or an IP router. Each edge $e_k \in \mathcal{E}$ represents a logical link between two network elements. We use the term "logical" to emphasize the fact that an edge does not necessarily represent a physical link; it may also represent an IP-level or domain-level link—in general, a sequence of physical links between two network elements.

The underlying nature of each node and edge depends on the method used for building the network graph. For instance, if an operator relies on traceroute to build the network graph, and all layer-3 elements in her network respond to traceroute, then each node in the resulting graph represents a layer-3 network element, while each edge represents an IP-level link. In the rest of the paper, unless otherwise specified, we will use the term "link" to refer to an edge in the network graph, i.e., a logical link between two network elements.

We define a "path" as a sequence of links whose congestion status (defined below) can be determined through end-to-end measurements. We denote the set of paths in the network by $\mathcal{P}$. If a path $P_i \in \mathcal{P}$ traverses a link $e_k \in \mathcal{E}$, then we write $e_k \in P_i$. A path never crosses a link more than once, i.e., there are no loops. All links participate in at least one path, i.e., there are no unused links.

We also define the "path coverage" function ψ, which maps a set of links $A \subseteq \mathcal{E}$ to the set of paths that they "cover," i.e., the set of paths that traverse at least one of these links:

$$\psi\left(A\right) = \{\ P_i \in \mathcal{P} \mid P_i \ni e_k \text{ for some } e_k \in A\ \}. \tag{1}$$

We denote by $|\psi\left(A\right)|$ the number of paths in $\psi\left(A\right)$. For example, in Figure 1(a), we have: $\psi\left(\{e_1, e_2\}\right) = \{P_1, P_2, P_3\}$.

Congestion. We divide time into even slots called "snapshots," such that each experiment involves a finite sequence of N snapshots.

We model the congestion behavior of link e_k during an experiment as a stationary random process: We define e_k's "packet-loss rate" during the n-th snapshot of the experiment as the fraction of packets that are not delivered to their next link out of all the packets that arrive at e_k during the snapshot. We say that e_k is "congested" (resp. "good") during the n-th snapshot, if its packet-loss rate is above (resp. below or equal to) a threshold t_l (we follow the model proposed in [10], where all links have the same link-congestion threshold t_l). For each link e_k and each snapshot n, we define a random variable $X_{e_k}(n)$ as follows:

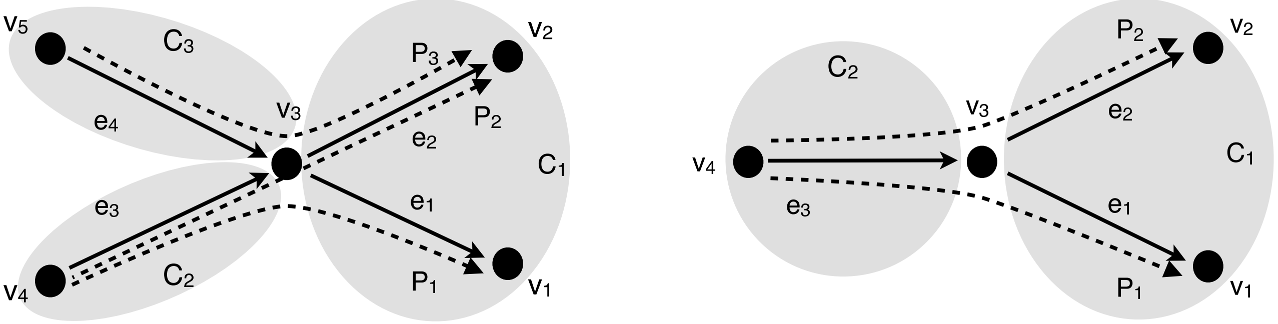
$$X_{e_k}(n) = \begin{cases} 1, & \text{if } e_k \text{ is congested during the } n\text{-th snapshot} \\ 0, & \text{otherwise.} \end{cases}$$

For a given link e_k, all random variables $X_{e_k}(n), n = 1..N$, are identically distributed; hence, for simplicity, we use X_{e_k} to denote any of them. This implies that the congestion probability of each link remains the same throughout the experiment. The scenario in which link e_k is always congested (resp. good) is a special, degenerate case of our model, where X_{e_k} is always 1 (resp. 0).

Similarly, we model the congestion behavior of path P_i during an experiment as a stationary random process: We define P_i's "packet-loss rate" during the n-th snapshot of the experiment as the fraction of packets that are not delivered to their destination out of all packets sent on P_i during that snapshot. We say that P_i is "congested" (resp. "good") during the n-th snapshot, if its packet-loss rate is above (resp. below or equal to) a threshold $t_p = 1 - (1 - t_l)^d$, where d is the number of links traversed by P_i. For each path P_i and each snapshot n, we define a random variable $Y_{P_i}(n)$ as follows:

$$Y_{P_i}(n) = \begin{cases} 1, & \text{if } P_i \text{ is congested during the } n\text{-th snapshot} \\ 0, & \text{otherwise.} \end{cases}$$

For a given path P_i, all random variables $Y_{P_i}(n), n = 1..N$, are identically distributed; hence, for simplicity, we use Y_{P_i} to denote any of them. Finally, Y_{P_i} has expected value $E(Y_{P_i}) = \mathbb{P}(Y_{P_i} = 1)$, equal to the fraction of snapshots during which P_i is congested. In other words, if, during the experiment, we observe that P_i is congested during half of the snapshots, we model this by saying that $\mathbb{P}(Y_{P_i} = 1) = 0.5$. The scenario in which path P_i is always congested (resp. good) is a special, degenerate case of our model, where Y_{P_i} is always 1 (resp. 0).



(a) A topology where Assumption 4 holds, i.e., each correlation subset covers a distinct set of paths. Links $\mathcal{E} = \{e_1, e_2, e_3, e_4\}$. Paths $\mathcal{P} = \{P_1, P_2, P_3\}$. Correlation sets $\mathcal{C} = \{\{e_1, e_2\}, \{e_3\}, \{e_4\}\}$. Correlation subsets $\tilde{\mathcal{C}} = \{\{e_1\}, \{e_2\}, \{e_1, e_2\}, \{e_3\}, \{e_4\}\}$.

(b) A topology where Assumption 4 does not hold, because correlation subsets $\{e_1, e_2\}$ and $\{e_3\}$ cover the same set of paths $\{P_1, P_2\}$. Links $\mathcal{E} = \{e_1, e_2, e_3\}$. Paths $\mathcal{P} = \{P_1, P_2\}$. Correlation sets $\mathcal{C} = \{\{e_1, e_2\}, \{e_3\}\}$. Correlation subsets $\tilde{\mathcal{C}} = \{\{e_1\}, \{e_2\}, \{e_1, e_2\}, \{e_3\}\}$.

Figure 1: Two toy topologies, where Assumption 4 holds (on the left) and does not hold (on the right).

Link Correlation. We say that two links e_k and e_l are "independent" or "uncorrelated" when the random variables X_{e_k} and X_{e_l} are independent (or, equivalently, since they are Bernoulli random variables, uncorrelated) from one another; this corresponds to the situation where e_k's congestion status during a snapshot cannot affect e_l's congestion status during the same or any other snapshot. Otherwise, we say that the two links are "correlated."

Previous work has generally assumed that links are uncorrelated. More precisely, the algorithm from [13] requires that any two links that participate in at least one common path are uncorrelated with one another, while the algorithms from [10, 12] explicitly require that any link is uncorrelated with any other link.

Instead, we assume that each link may be correlated with a specific set of other links, and we assume that we know which links may be correlated with one another. Based on this knowledge, we group links into "correlation sets," such that two links from the same correlation set may be correlated with one another, but not with links from other correlation sets. For instance, in Figure 1(a), we know that links e_1 and e_2 may be correlated (e.g., because they are sharing a physical link), whereas each of links e_3 and e_4 may not be correlated with any other link; hence, we assign links e_1 and e_2 to the same correlation set C_1, link e_3 to correlation set C_2, and link e_4 to correlation set C_3.

More formally, consider a partition $\mathcal{C} = \{C_1, C_2, ...C_{|\mathcal{C}|}\}$ of $\mathcal{E}$, such that:

$$\forall e_k, e_l,\ k \neq l, \quad \text{s.t.} \quad e_k \in C_p,\ e_l \in C_q,\ p \neq q,$$
$$e_k \text{ is uncorrelated with } e_l;$$

we call each set of links $C_p \in \mathcal{C}$ a "correlation set." In the scenario where each link in the network is uncorrelated with any other, there are $|\mathcal{E}|$ correlation sets, one for each link in the network, i.e., $\mathcal{C} = \{\{e_1\}, \{e_2\}, ...\{e_{|\mathcal{E}|}\}\}$. At the other extreme, if all links in the network are correlated with one another, there is only one correlation set that consists of all links in the network, i.e., $\mathcal{C} = \{\{e_1, e_2, ...e_{|\mathcal{E}|}\}\}$.

In our analysis, we will often refer to a "correlation subset" $A \subseteq C_p,\ A \neq \emptyset$, i.e., a non-empty subset of a correlation set. We will also refer to the set of all possible correlation subsets:

$$\tilde{\mathcal{C}} = \{\ A \subseteq \mathcal{E} \mid A \neq \emptyset \text{ and } A \subseteq C_p \text{ for some } C_p \in \mathcal{C}\ \}.$$

For instance, in Figure 1(a), we have $\mathcal{C} = \{\{e_1, e_2\}, \{e_3\}, \{e_4\}\}$ and $\tilde{\mathcal{C}} = \{\{e_1\}, \{e_2\}, \{e_1, e_2\}, \{e_3\}, \{e_4\}\}$.

2.2 Assumptions

We make four assumptions. The first three are inherited from previous work on tomography, whereas the fourth assumption is new—and key to the contribution of this paper.

ASSUMPTION 1. *Stability: The set of paths $\mathcal{P}$ remains unchanged during each experiment.*

This assumption is common in all tomographic algorithms. In practice, if a path changes during a certain snapshot, we discard the measurements collected during that snapshot and stop the current experiment, i.e., we apply our algorithms only on sequences of snapshots during which $\mathcal{P}$ remains unchanged.

ASSUMPTION 2. *Separability: A path is good if and only if all the links it traverses are good. A path is congested if and only if at least one of the links it traverses is congested.*

This assumption is common in Boolean-tomography algorithms [10, 12]. It is closely related to the problem of setting the link-congestion threshold t_l, defined in Section 2.1. We use $t_l = 0.01$, which has been shown to work well for mesh topologies and introduce negligible error [10].

ASSUMPTION 3. *Stationarity: The congestion behavior of any link during an experiment can be modeled as a stationary random process.*

This assumption is inherent in our congestion model (inherited from [12]), since we use the identically distributed random variables $X_{e_k}(n), n = 1..N$, to represent link e_k's congestion status during subsequent snapshots. We would like to clarify that this does not mean that the congestion status of link e_k remains the same during the experiment, only that the *probability* of link e_k being congested remains the same. We need this assumption in order to keep the theoretical analysis (Section 3) simple—to model e_k's congestion status as a non-stationary random process, we would need to assume that each of the random variables $X_{e_k}(n), n = 1..N$, has a potentially different probability distribution.

Assumption 4. *Identifiability: Given any two correlation subsets $A, B \in \tilde{\mathcal{C}}$, $A \neq B$, it holds true that $\psi(A) \neq \psi(B)$, i.e., A and B are not traversed by exactly the same paths.*

This generalizes a fundamental assumption of network tomography, that any two *links* are not traversed by exactly the same paths [18]. Intuitively, this earlier assumption captured the fact that, when two links participate in exactly the same paths, there is no way to differentiate between the two links based on end-to-end observations. We are generalizing this to say that, when two *groups of correlated links* participate in exactly the same paths (and assuming we know nothing about the nature of the correlation), there is no way to differentiate between the two groups based on end-to-end observations. Indeed, in the extreme case where each link in the network is uncorrelated with any other, our assumption becomes exactly the earlier assumption.

3. FEASIBILITY

3.1 Problem Statement and Result

Suppose we can perform unicast end-to-end path measurements, i.e., measure the probability that any path or combination of paths is congested. We want to determine whether, given this information, it is feasible to identify the probability that a particular set of links (or a particular link) is congested.

Theorem 1. *If Assumptions 1, 2, 3 and 4 hold, the probability that any set of links is congested is identifiable for all possible sets of links.*

Proof. In the Appendix, Section A. □

To illustrate this result, we first consider the scenario depicted in Figure 1(a), where Assumption 4 holds. Here is the set of paths $\psi(A)$ covered by each correlation subset $A \in \tilde{\mathcal{C}}$:

$A \in \tilde{\mathcal{C}}$	$\psi(A)$
$\{e_1\}$	$\{P_1\}$
$\{e_2\}$	$\{P_2, P_3\}$
$\{e_1, e_2\}$	$\{P_1, P_2, P_3\}$
$\{e_3\}$	$\{P_1, P_2\}$
$\{e_4\}$	$\{P_3\}$

Indeed, each correlation subset A covers a distinct set of paths $\psi(A)$. Intuitively, this allows us to measure the probability that the paths covered by each correlation subset are congested and infer, from that, the probability that the links in each correlation subset are congested.

To illustrate the challenge introduced by link correlation, we also consider the scenario depicted in Figure 1(b), where Assumption 4 does not hold. In this case, we have:

$A \in \tilde{\mathcal{C}}$	$\psi(A)$
$\{e_1\}$	$\{P_1\}$
$\{e_2\}$	$\{P_2\}$
$\{e_1, e_2\}$	$\{P_1, P_2\}$
$\{e_3\}$	$\{P_1, P_2\}$

i.e., correlation subsets $\{e_3\}$ and $\{e_1, e_2\}$ cover the same set of paths $\{P_1, P_2\}$. As a result, we cannot distinguish between the probability that e_3 is congested and the probability that e_1 and e_2 are both congested.

If links e_1 and e_2 were uncorrelated, we would not have this problem. We could do the following: (i) Measure the probability that P_1 is the only congested path, which is equal to the probability that e_1 is the only congested link; from that, compute the probability that e_1 is congested, as shown in [12]. (ii) Repeat for P_2 and e_2. (iii) *Using the fact that e_1 and e_2 are uncorrelated*, compute the probability that e_1 and e_2 are both congested as the product of the probability that link e_1 is congested and of the probability that link e_2 is congested. (iv) Measure the probability that P_1 and P_2 are both congested, which is a function of the probability that e_3 is congested and the probability that e_1 and e_2 are both congested. Since we have already computed the latter, we could now compute the probability that e_3 is congested. This is, at a high level, the algorithm followed in [12].

The problem is that, in our setup, e_1 and e_2 are correlated, which means that we cannot perform step (iii). We can go directly to step (iv) and measure the probability that P_1 and P_2 are both congested; however, since we do not know the probability that e_1 and e_2 are both congested, we cannot draw any conclusions about e_3.

3.2 Proof Illustration

We now illustrate how our proof works using the example of Figure 1(a). Again, we assume that we can measure the probability that any set of paths is congested and we want to identify the probability that each set of links is congested, i.e., $\mathbb{P}(X_{e_{1..4}} = 1)$, $\mathbb{P}(X_{e_1} = 1, X_{e_2} = 1)$, etc.

As already hinted, the challenge in proving Theorem 1 comes from the fact that links may be correlated. As a result, we cannot easily compute joint probabilities, e.g., we cannot compute $\mathbb{P}(X_{e_k} = 1, X_{e_l} = 1)$ simply by multiplying $\mathbb{P}(X_{e_k} = 1)$ and $\mathbb{P}(X_{e_l} = 1)$. Yet to prove that we can identify $\mathbb{P}(X_{e_k} = 1)$ for all links e_k, we do need to compute many such joint probabilities. In fact, we need to compute the probability that all the links in A are congested *for every possible correlation subset $A \in \tilde{\mathcal{C}}$*. Hence, the core of our proof—and a key contribution of this work—consists of showing that we can, indeed, compute all these $|\tilde{\mathcal{C}}|$ probabilities.

Definitions and Notation. We start by defining and providing compact notation for certain terms that appear frequently in our illustration. All defined symbols are summarized in Table 1 in the Appendix.

▷ S^p is a random set equal to the set of all congested links in correlation set C_p during a snapshot:

$$S^p \equiv \{\ e_k \in C_p \mid X_{e_k} = 1\ \}.$$

Since $X_{e_k} \in C_p$ and $X_{e_l} \in C_q \neq C_p$ are independent, it is also the case that S^p and $S^q \neq S^p$ are independent. In the example of Figure 1(a), we have three such independent random sets, S^1, S^2, and S^3.

▷ The "network state," denoted by $\mathcal{S}$, is a random set equal to the set of all congested links during a snapshot:

$$\mathcal{S} \equiv \bigcup_{p=1..|\mathcal{C}|} S^p.$$

E.g., in Figure 1(a), $\mathcal{S} \equiv S^1 \cup S^2 \cup S^3$.

$\triangleright$ $\psi(\mathcal{S})$ is a random set equal to the set of congested paths during a snapshot:

$$\psi(\mathcal{S}) \equiv \bigcup_{p=1..|\mathcal{C}|} \psi(S^p).$$

Given these definitions, we will refer to the following events, given a correlation subset $A \subseteq C_p$:

$\triangleright$ $(S^p = A)$ is the event that the links in A are the only congested links in $C_p \supseteq A$. E.g., in Figure 1(a), $(S^1 = \{e_1\})$ is the event that e_1 is congested and e_2 is good.

$\triangleright$ $(\psi(\mathcal{S}) = \psi(A))$ is the event that the paths covered by A are the only congested paths in the network. E.g., in Figure 1(a), $(\psi(\mathcal{S}) = \psi(\{e_1\}))$ is the event that P_1 is congested, while P_2 and P_3 are good.

$\triangleright$ Finally, for each correlation subset $A \subseteq C_p$, we define its "congestion factor" α_A as follows:

$$\alpha_A \;=\; \mathbb{P}(S^p = A) \;/\; \mathbb{P}(S^p = \emptyset). \qquad (2)$$

This expresses how often the links in $A \subseteq C_p$ are congested compared to how often *all* links in the correlation set C_p are good.

Setup. Through end-to-end measurements, we can measure the probability that all paths are good, i.e.,

$$\mathbb{P}(\psi(\mathcal{S}) = \emptyset) \;=\; \mathbb{P}(S^1 = \emptyset)\,\mathbb{P}(S^2 = \emptyset)\,\mathbb{P}(S^3 = \emptyset). \qquad (3)$$

Moreover, given a correlation subset A, we can measure the probability that the paths in $\psi(A)$ are the *only* congested paths in the network, i.e., $\mathbb{P}(\psi(\mathcal{S}) = \psi(A))$.

Step 1. Consider correlation subset $\{e_1\}$ and the set of paths it covers, $\psi(\{e_1\}) = \{P_1\}$. Consider the event that P_1 is the only congested path in the network. If this is the case, then e_1 must be the only congested link, i.e, the network can only be in state $\mathcal{S} = \{e_1\}$:

S^1	S^2	S^3	$\mathcal{S}$	$\psi(\mathcal{S})$
$\{e_1\}$	$\emptyset$	$\emptyset$	$\{e_1\}$	$\{P_1\}$

Hence, we can write:

$$\mathbb{P}(\psi(S) = \psi(\{e_1\})) \;=\; \mathbb{P}(S^1 = \{e_1\})\,\mathbb{P}(S^2 = \emptyset)\,\mathbb{P}(S^3 = \emptyset).$$

If we divide this by Eq. 3, we get

$$\frac{\mathbb{P}(\psi(S) = \psi(\{e_1\}))}{\mathbb{P}(\psi(S) = \emptyset)} \;=\; \frac{\mathbb{P}(S^1 = \{e_1\})}{\mathbb{P}(S^1 = \emptyset)} \;=\; \alpha_{\{e_1\}}.$$

Since both the numerator and denominator of the left-most term can be measured, we can compute $\alpha_{\{e_1\}}$.

Step 2. Consider correlation subset $\{e_3\}$ and the set of paths it covers, $\psi(\{e_3\}) = \{P_1, P_2\}$. Consider the event that P_1 and P_2 are the only congested paths in the network. If this is the case, either e_3 is the only congested link, or e_1 and e_3 are the only congested links, i.e., the network can only be in state $\mathcal{S} = \{e_3\}$ or in state $\mathcal{S} = \{e_1, e_3\}$:

S^1	S^2	S^3	$\mathcal{S}$	$\psi(\mathcal{S})$
$\emptyset$	$\{e_3\}$	$\emptyset$	$\{e_3\}$	$\{P_1, P_2\}$
$\{e_1\}$	$\{e_3\}$	$\emptyset$	$\{e_1, e_3\}$	$\{P_1, P_2\}$

Hence, we can write:

$$\begin{aligned}\mathbb{P}(\psi(\mathcal{S}) = \psi(\{e_3\})) \;=\; & \\ & \mathbb{P}(S^1 = \emptyset)\,\mathbb{P}(S^2 = \{e_3\})\,\mathbb{P}(S^3 = \emptyset) \;+ \\ & \mathbb{P}(S^1 = \{e_1\})\,\mathbb{P}(S^2 = \{e_3\})\,\mathbb{P}(S^3 = \emptyset).\end{aligned}$$

If we divide this by Eq. 3, we get

$$\frac{\mathbb{P}(\psi(\mathcal{S}) = \psi(\{e_3\}))}{\mathbb{P}(\psi(\mathcal{S}) = \emptyset)} \;=\; (1 + \alpha_{\{e_1\}})\,\alpha_{\{e_3\}}.$$

Since both the numerator and denominator of the left-most term can be measured, and we have already computed $\alpha_{\{e_1\}}$, we can now compute $\alpha_{\{e_3\}}$.

Step 3. With the same rationale, we compute all congestion factors α_A, $\forall A \in \tilde{\mathcal{C}}$. The gist is that, thanks to Assumption 4, we can *order* the correlation subsets and compute their congestion factors such that each factor depends only on terms that can be measured or have already been computed. In our particular example, the ordering we follow is $\langle \{e_1\}, \{e_4\}, \{e_3\}, \{e_2\}, \{e_1, e_2\} \rangle$.

Step 4. According to Lemma 3 (Appendix, Section A.3), once we have computed all congestion factors, we can derive $\mathbb{P}(X_{e_{1..4}} = 1)$ and $\mathbb{P}(X_{e_1} = 1, X_{e_2} = 1)$. Once we know these 5 probabilities, we can easily compute the rest, e.g., $\mathbb{P}(X_{e_1} = 1, X_{e_3} = 1) \;=\; \mathbb{P}(X_{e_1} = 1)\,\mathbb{P}(X_{e_3} = 1)$.

3.3 Practical Significance

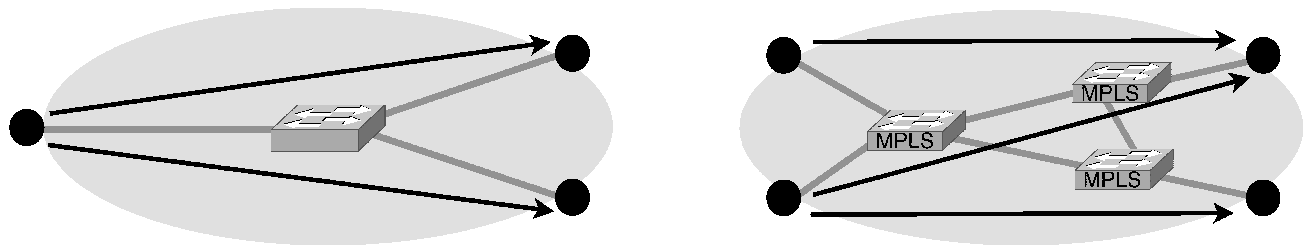
We now answer certain questions regarding the practical significance of our result.

$\triangleright$**In what kind of scenarios are links correlated?** There are at least two situations that can lead to link correlation: (i) a congested network resource is shared by multiple links; (ii) congestion is caused by a traffic pattern that involves a particular set of links.

The first situation can occur when an operator is using network tomography to monitor the quality of links in a network and relies on traceroute to discover the network topology. As a result, she misses all nodes that do not respond to traceroute, e.g., all Ethernet and MPLS switches. In the resulting network graph, each node represents a layer-3 element, while each edge represents a sequence of physical links between two layer-3 elements. Hence, two distinct edges in the network graph (i.e., two distinct logical links) may share physical links; when one of these shared physical links is congested, both logical links are congested, which means that the two logical links are correlated.

The second situation can occur when a badly written network protocol, a distributed application, or even a denial-of-service attack causes a particular set of source nodes to generate high-rate traffic to a particular set of destination nodes. For instance, consider a group of end-hosts participating in a distributed game, which causes them to periodically exchange high-rate traffic; or, a botnet master commanding a group of compromised end-hosts to periodically send high-rate traffic to a group of public-access sites. Either scenario can result in a particular set of links becoming congested and de-congested at the same times, i.e., becoming correlated.

$\triangleright$**In which of these scenarios is our result useful?** In the scenarios where the operator knows which links are most likely to be correlated.



(a) A local-area network. The undiscovered node in the middle is an Ethernet switch.

(b) An administrative domain. The undiscovered nodes in the middle are MPLS switches.

Figure 2: Two scenarios where logical links are correlated because they are sharing physical links. Each figure shows a part of a network graph. Black nodes (circles) represent IP routers discovered with traceroute, while black arrows represent logical links between discovered IP routers. Gray nodes represent undiscovered network elements (that do not respond to traceroute), while gray lines represent physical links.

For instance, consider the scenario where an operator uses network tomography to monitor the quality of links in her domain and relies on traceroute to discover the domain's topology. This may seem unreasonable at first—one would assume that an operator already knows the topology of her own domain. Yet, in practice, the operator of a large network (e.g., a university-campus network) does not always have access to all areas and equipment of the network. Moreover, given that paths change in response to network conditions, the operator does not always know which links compose each path. In this context, it makes sense for the operator to map each local-area network discovered through traceroute to one correlation set (Figure 2(a)). The links in each correlation set are potentially correlated, because they may be sharing physical links and/or management processes.

Alternatively, consider the scenario where the operator of one administrative domain uses network tomography to determine whether a set of neighboring domains are honoring their SLA, but does not have visibility into the internals of these domains, because they use MPLS for internal routing. In this context, it makes sense for the operator to map each administrative domain to one correlation set (Figure 2(b)). As above, the links in each correlation set are potentially correlated, because they may be sharing physical links and/or management processes.

Our result is not useful in scenarios where the operator does not know which links may be correlated, e.g., when an unpredictable traffic pattern affects the congestion status of multiple otherwise uncorrelated links. For instance, consider the denial-of-service scenario mentioned above. Unless the botnet's structure and targets are known, there is no way to guess the link-correlation pattern caused by the attack. Hence, an operator using our algorithm to identify the links affected by the attack would mislabel these links as uncorrelated, causing the algorithm to yield inaccurate results.

▷In what topologies does Assumption 4 hold?
In topologies where each intermediate node touches multiple correlation sets, like the one depicted in Figure 1(a).

Assumption 4 does not hold when there exists an intermediate node that has all its ingress links in one correlation set and all its egress links also in one (the same or different) correlation set; this causes the correlation subset formed by its ingress links and the correlation subset formed by its egress links to cover exactly the same paths (all the paths that traverse the node). This situation occurs when an intermediate node touches one correlation set and sometimes even when it touches two correlation sets, as in Figure 1(b). In contrast, in Figure 1(a), there is no intermediate node that touches one or two correlation sets, and Assumption 4 holds.

▷What happens when Assumption 4 does not hold?
Our algorithm cannot accurately compute the congestion probability of those links that belong to correlation subsets for which the assumption does not hold—we will refer to these links as "unidentifiable."

One way to deal with this situation is to try to alter the topology, i.e., add nodes and paths to the system, such that Assumption 4 holds. For instance, consider the topology of Figure 1(b), where the assumption does not hold; by adding node v_5 and path P_3 we get the topology of Figure 1(a), where the assumption does hold. If altering the topology is not an option (e.g., because the operator does not have access to additional nodes for performing end-to-end measurements from/to them), we can act as if the unidentifiable links were uncorrelated; as a result, we compute their congestion probability inaccurately, but, at least, we can still compute the congestion probability of the remaining links accurately. We will look at this effect in Section 5.

Moreover, in certain cases, when Assumption 4 does not hold, we can apply a transformation to the network topology (merge certain consecutive links) so that it does. This is akin to the merging of consecutive links traversed by the same paths in the context of traditional network performance tomography. We will now outline how our transformation relates to traditional tomography and how it works, however, for lack of space, we will not provide here a formal description and analysis.

A fundamental assumption of network performance tomography is that any two links are not traversed by exactly the same paths. However, if two such "indistinguishable" links occur *consecutively* in the network, they can be elegantly abstracted away into a single "merged" link, such that the above fundamental assumption holds. This transformation enables tomographic algorithms to work, albeit at decreased granularity, i.e., they can characterize each link in the transformed network graph (including the merged links), but not the original links that were merged.

We are in a similar situation: According to our Assumption 4, any two correlation subsets must not be traversed by exactly the same paths. However, if two such subsets occur consecutively in the network, we can abstract them away into a single subset of merged links, such that the assumption 4 holds. This transformation enables our algorithm to work, albeit at decreased granularity, i.e., it can accurately characterize each link in the transformed network graph, but not the original links that were merged.

The transformation works as follows: If intermediate node v has all its ingress links in one correlation set and all its egress links also in one correlation set, we remove v and all its adjacent links from the network graph; if, in the original network graph, there was a path that went consecutively through nodes v_{last}, v, and v_{next}, then we draw a merged link from v_{last} to v_{next}. For instance, to transform the topology of Figure 1(b), we remove node v_3 and its adjacent links (links e_1, e_2, and e_3) and draw, in their place, two merged links, one from v_4 to v_1 and one from v_4 to v_2; as a result, we end up with a single correlation set that contains two (merged) links.

▷**Why not assign all links to one correlation set?**

If we do that, Assumption 4 does not hold for any part of the network graph, hence, our algorithm does not yield any useful results.

If we assign all links in the network graph to the same correlation set, Assumption 4 does not hold. If we apply the transformation described above, we end up with a transformed network graph where each link corresponds to an end-to-end path. At that point, we can compute the congestion probability of each link (in the transformed network graph) through end-to-end measurements—network tomography cannot provide any additional information. For instance, consider a topology that consists of the nodes and links depicted in Figure 1(a), but where all four links belong to the same correlation set. In that topology, Assumption 4 does not hold, because node v_3 has all its ingress and egress links in the same correlation set. To apply our transformation, we remove v_3 and all its adjacent links (i.e., all four links), and draw three merged links in their place: from v_4 to v_1, from v_4 to v_2, and from v_5 to v_2. As a result, we get a transformed network graph like the one of Figure 2(b), where each link corresponds to an end-to-end path, which means that we can measure its congestion probability through end-to-end measurements.

▷**Can our result help determine whether a link was congested or not?**

Yes. Our result states that it is feasible to compute the probability that any set of links is congested. Obtaining this information is the first step toward accurately computing which particular set of links were congested during each snapshot.

There exist several tomographic algorithms that observe which set of paths were congested during each snapshot and try to determine which set of links were congested during the last snapshot [13, 10, 12]. This is an "ill-posed inverse problem," i.e., there are typically multiple sets of links that, if congested, would have led to the observed outcome. Each algorithm uses its own technique to choose which of all the feasible solutions is the most likely: the first two algorithms favor solutions that involve a small number of congested links, whereas the last one explicitly computes which of the feasible solutions is the most likely (albeit assuming no link correlation).

Our result implies that the last approach (explicitly computing which feasible solution is the most likely) can also work in the presence of link correlations (as long as we have observed the network long enough to have computed the probability of each solution). As part of our future work, we plan to build an algorithm that determines which particular set of links were congested during each snapshot using this approach.

4. ALGORITHM

In this section, we present an algorithm that takes as input unicast end-to-end measurements and outputs, for each link, the probability that the link is congested, i.e., $\mathbb{P}(X_{e_k} = 1)$ for all links e_k in $\mathcal{E}$.

We already know that such an algorithm exists from Theorem 1 (which tells us that we can identify the probability that any set of links is congested, hence, also the probability that any individual link is congested). Moreover, we already have such an algorithm from the proof of Theorem 1, which is a proof by construction (we will refer to it as the "theorem algorithm"). Unfortunately, the theorem algorithm is impractical, because it requires an amount of computation that depends on the number of correlation subsets in the network, which grows exponentially with the size of correlation sets. For instance, if a correlation set consists of 100 links—a plausible number, if the correlation set represents an Autonomous System (AS), it introduces more than 10^{30} correlation subsets.

We now describe a more practical algorithm, which requires an amount of computation that depends only on the number of links in the network. Our algorithm achieves this by leveraging the following observation: The theorem algorithm takes long, because it computes the probability of congestion of every possible combination of links. However, we are not interested in all these probabilities; we are only interested in the probability of congestion of every *individual* link. Hence, our algorithm tries to form just enough equations to solve for these unknowns.

We first illustrate through the example of Figure 1(a):

▷ Consider the case where path P_1 is good. Using Assumption 2 (when a path is good, all its links are necessarily good), we can write:

$$\begin{aligned}
\mathbb{P}(Y_{P_1} = 0) &= \mathbb{P}(X_{e_1} = 0)\,\mathbb{P}(X_{e_3} = 0) \;\Leftrightarrow \\
\log(\mathbb{P}(Y_{P_1} = 0)) &= \log(\mathbb{P}(X_{e_1} = 0)\,\mathbb{P}(X_{e_3} = 0)) \\
&= \log(\mathbb{P}(X_{e_1} = 0)) \;+\; \log(\mathbb{P}(X_{e_3} = 0)) \\
\Leftrightarrow y_1 &= x_1 \;+\; x_3 \qquad (4)
\end{aligned}$$

where $y_i = \log(\mathbb{P}(Y_{P_i} = 0))$ and $x_k = \log(\mathbb{P}(X_{e_k} = 0))$.

▷ Similarly, if we consider the cases where paths P_2 and P_3 are good, we can write:

$$\begin{aligned}
y_2 &= x_2 \;+\; x_3 \qquad (5) \\
y_3 &= x_2 \;+\; x_4 \qquad (6)
\end{aligned}$$

▷ Finally, if we consider the case where paths P_2 and P_3 are both good, we can write:

$$\begin{aligned}
&\mathbb{P}(Y_{P_2} = 0,\; Y_{P_3} = 0) \;= \\
&\qquad \mathbb{P}(X_{e_2} = 0)\,\mathbb{P}(X_{e_3} = 0)\,\mathbb{P}(X_{e_4} = 0) \\
&\qquad\quad \Leftrightarrow\; y_{23} \;=\; x_2 \;+\; x_3 \;+\; x_4 \qquad (7)
\end{aligned}$$

where $y_{ij} = \log(\mathbb{P}(Y_{P_i} = 0,\; Y_{P_j} = 0))$.

▷ Equations 4, 5, 6, 7 form a system of 4 linearly independent equations, from which we can compute $x_{1..4}$ and, consequently, $\mathbb{P}(X_{e_{1..4}} = 0)$ and $\mathbb{P}(X_{e_{1..4}} = 1)$.

There are two things to note about the above example. First, we used not only single-path observations ($y_{1..3}$), but also two-path observations (y_{23}), which allowed us to form 4 linear equations, exactly as many as we needed to solve for our unknowns. Second, to form our fourth equation, we used paths P_2 and P_3. What is special about these two paths is that they do not involve any correlated links. If, instead, we had used paths P_1 and P_2, our fourth equation would have been:

$$\mathbb{P}(Y_{P_1} = 0,\ Y_{P_2} = 0) \ = \ \mathbb{P}(X_{e_1} = 0,\ X_{e_2} = 0)\, \mathbb{P}(X_{e_3} = 0)$$
$$\Leftrightarrow y_{12} \ = \ x_{12} \ + \ x_3 \qquad (8)$$

which introduces a new unknown, $x_{12} = \log(\ \mathbb{P}(X_{e_1} = 0,$ $X_{e_2} = 0\ )$. So, by considering paths and combinations of paths that do not involve any correlated links, we can form equations that do not introduce new unknowns beyond x_k for all links.

Our algorithm, then, consists of the following steps:

▷ Identify all paths P_i that do not involve correlated links and form a set of $N_1 \leq |\mathcal{P}|$ linearly independent equations

$$y_i = \sum_{k \text{ s.t. } e_k \in P_i} x_k \qquad (9)$$

▷ Identify all pairs of paths that do not involve correlated links, and form a set of N_2 linearly independent equations

$$y_{ij} = \sum_{\substack{k \text{ s.t. } e_k \in P_i \\ \text{or } e_k \in P_j}} x_k \qquad (10)$$

▷ If $N_1 + N_2 = |\mathcal{E}|$, we compute x_k for all links $e_k \in \mathcal{E}$ by solving the system of $|\mathcal{E}|$ equations that we have formed. If $N_1 + N_2 < |\mathcal{E}|$, there are multiple solutions that satisfy our system of equations; we pick the one that minimizes the L1 norm error.

In all topologies we experimented with, by considering paths and pairs of paths that did not involve correlated links, we were able to gather close to $|\mathcal{E}|$ linearly independent equations; thanks to L1 norm minimization, the error introduced by the (few) missing equations was negligible.

5. EVALUATION

Simulator. We built a simulator, in which the network is represented as a graph, with vertices corresponding to nodes and edges corresponding to links. In the beginning of each experiment, we determine which links belong to each correlation set, the probability of congestion of each link, and the joint probability of congestion of each set of correlated links.

An experiment consists of multiple rounds. In each round, we take the following actions:

▷ For each link, we determine whether it will be congested in this round or not, such that we respect the individual and joint probabilities of congestion determined in the beginning of the experiment.

▷ To each link, we assign a packet-loss rate according to the loss model from [13] (also similar to the models from [11, 16]), which assigns packet-loss rates between 0 and 0.01 to good links and between 0.01 and 1 to congested links.

▷ We simulate the scenario in which a given number of packets are sent along each path, hence, along each link. For each packet that arrives at a given link, we flip a coin to determine whether it will be dropped or not, such that we respect the packet-loss rate of the link determined in the previous step.

▷ We measure the packet-loss rate of each path as the fraction of packets sent along the path that were lost. If the packet-loss rate of a path is above the congestion threshold $t_p = 1-(1-t_l)^d$, we identify the path as congested; $t_l = 0.01$, as proposed in [10].

Topologies. We experimented with two types of topologies.

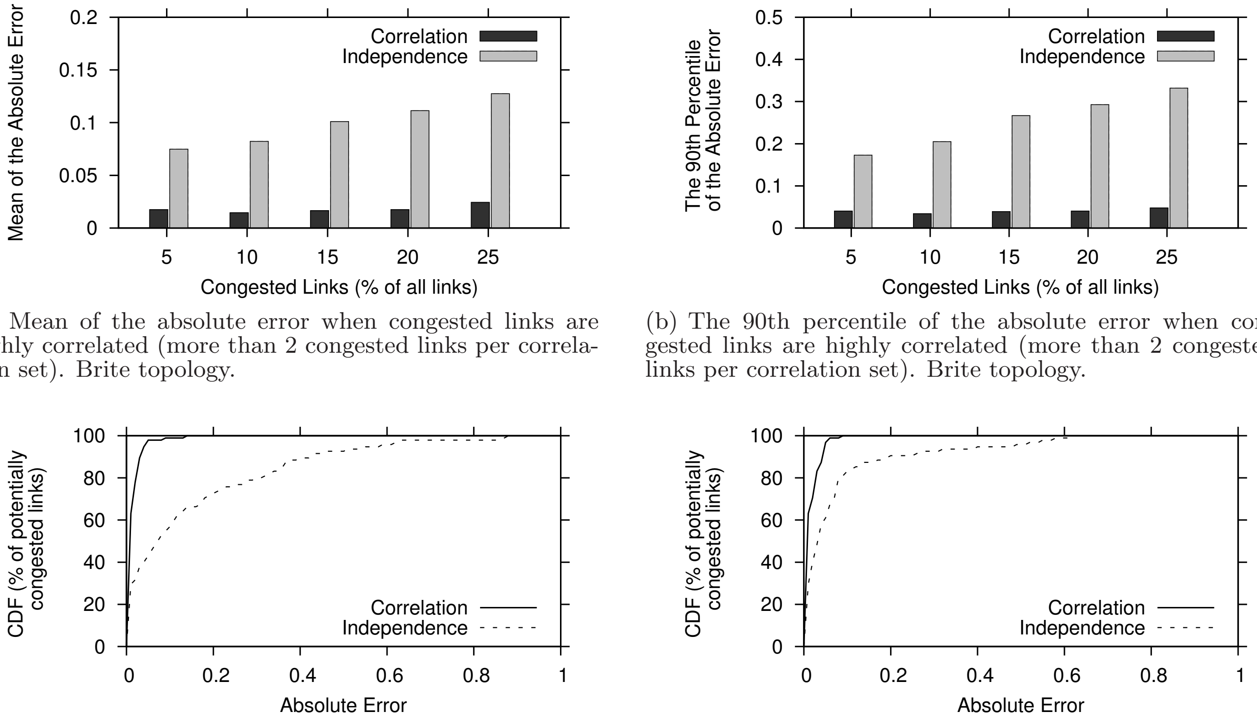
(i) **Brite topologies**: We used the Brite topology generator [1] to obtain pairs of AS-level and router-level topologies. From each pair, we used the AS-level topology to derive a network graph for our simulator and the router-level topology to determine the degree of correlation between each pair of links in the network graph; the point was to simulate scenarios where each correlation set corresponds to an administrative domain. More specifically, we mapped each link in the AS-level topology to a sequence of links in the router-level topology. We assumed that any two links in the router-level topology are uncorrelated; and that two links in the AS-level topology are correlated if and only if they share at least one link in the underlying router-level topology. In the beginning of the experiment, we assigned a probability of congestion to each link in the router-level topology, then derived the probability of congestion of each AS-level link and each set of correlated AS-level links accordingly. We show results for topologies consisting of 1500 paths.

(ii) **PlanetLab topologies**: To obtain each of these topologies, we got hold of several PlanetLab nodes and ran traceroute between them to identify the sequence of routers on each path; we discarded all paths with incomplete traceroute results. We assigned links to correlation sets, such that each correlation set consisted of a contiguous cluster of links; the point was to simulate scenarios where each correlation set corresponds to a local-area network or an administrative domain. We show results for topologies consisting of roughly 2000 links and 1500 paths.

Simulated Algorithms. We compare the algorithm presented in Section 4 (we will call this "correlation algorithm") against the algorithm presented in [12], which assumes that all links are independent (we will call this "independence algorithm"). To the best of our knowledge, this is the only tomographic algorithm that computes the congestion probabilities of links. The fact that our algorithm performs better is not surprising, since we consider scenarios where links are correlated. The point of the comparison is to show that, when links are correlated, ignoring this correlation leads to significant error, even when congestion and correlation are limited.

Metrics. To evaluate the performance of each algorithm, we look at the absolute error between the actual congestion probability of a link and its congestion probability as computed by the algorithm. For instance, if the actual congestion probability of a link is 0.5, but the algorithm thinks it is 0.1, then the absolute error is 0.4.

We use three ways to illustrate the performance of each algorithm for a given experiment: (i) We plot the cumulative distribution function (CDF) of the absolute error for all the *potentially congested links*, i.e., all links that participate in at least one congested path. For a perfect algorithm, this CDF would be a single point at $x = 0$, $y = 100\%$, i.e., the algorithm would compute the congestion probability of



(a) Mean of the absolute error when congested links are highly correlated (more than 2 congested links per correlation set). Brite topology.

(b) The 90th percentile of the absolute error when congested links are highly correlated (more than 2 congested links per correlation set). Brite topology.

(c) CDF of the absolute error when 10% of the links are congested. Congested links are highly correlated (more than 2 congested links per correlation set). Brite topology.

(d) CDF of the absolute error when 10% of the links are congested. Congested links are loosely correlated (up to 2 congested links per correlation set). Brite topology.

Figure 3: Performance of the two algorithms under ideal conditions, when different fractions of the links are congested.

all the potentially congested links with an absolute error of 0. In general, the earlier the CDF hits the $y = 100\%$ line, the better the performance of the corresponding algorithm. (ii) For a more compact illustration, we show the 90th percentile of the absolute error for all the potentially congested links, i.e., the absolute error that corresponds to a value of $y = 90\%$ of the CDF. For instance, if the 90th percentile of the absolute error is 0.1, this means that the corresponding algorithm computed the congestion probability of 90% of the potentially congested links with an absolute error below 0.1. (iii) We show the mean of the absolute error for all the potentially congested links.

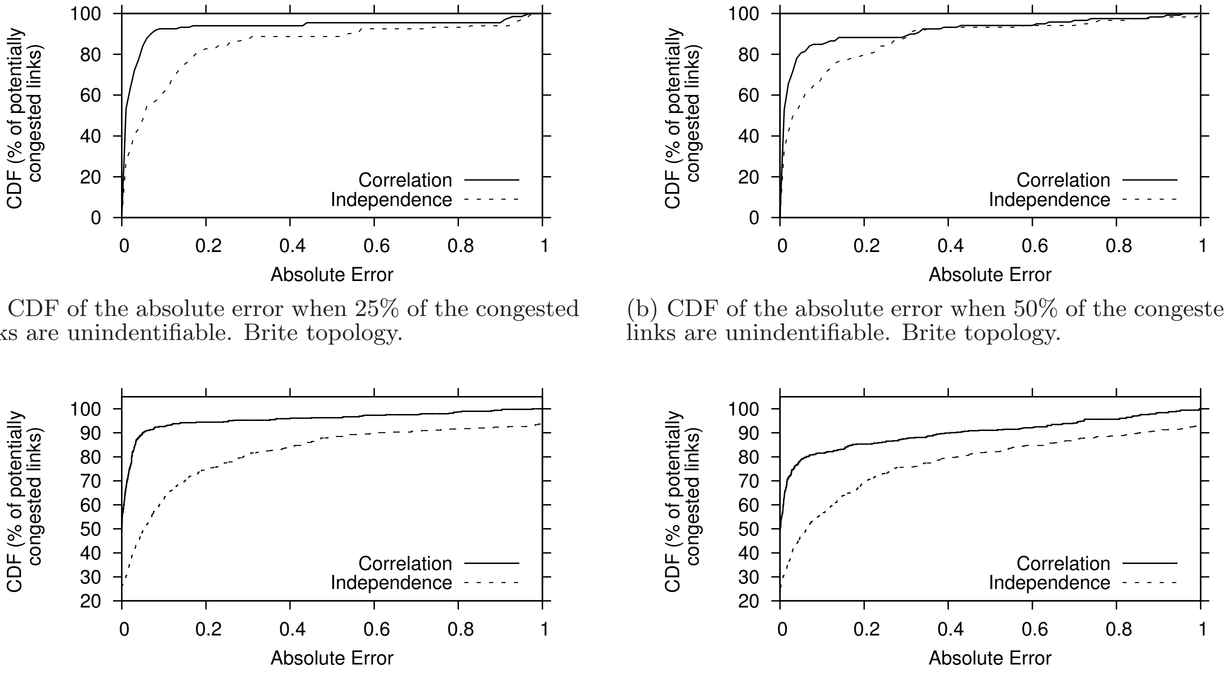
Performance Under Ideal Conditions. We first look at the performance of the two algorithms when the congestion probabilities of all links are identifiable (i.e., Assumption 4 holds) and there are no unknown correlation patterns in the network (i.e., links from different correlation sets are never correlated). Figure 3 shows the results for a Brite topology and various congestion scenarios.

First, we observe that our algorithm performs well even when congested links are highly correlated. As the percentage of congested links in the network increases from 5% to 25%, the mean of the absolute error stays below 0.03 (Figure 3(a)), while the 90th percentile of the absolute error stays below 0.1 (Figure 3(b)). For the independence algorithm, the mean and the 90th percentile of the absolute error increase with the percentage of congested links, because more congestion in the network implies that more correlated links are congested. When 10% of the links are congested, our algorithm computes the congestion probability of 95% of the links with an error below 0.1; the independence algorithm computes the congestion probability of only 50% of the links with this error (Figure 3(c)).

Second, we observe that taking correlation into account matters, even when the congested links are (very) loosely correlated. For instance, when 10% of the links are congested, and there are only up to 2 congested links per correlation set, our algorithm computes the congestion probability of 95% of the links with an error below 0.1; the independence algorithm computes the congestion probability of only 80% of the links with this error (Figure 3(d)).

Unidentifiable Links. Next, we look at the performance of the two algorithms when Assumption 4 does not hold, i.e., there exist correlation subsets whose congestion probability is unidentifiable; for brevity, we refer to links that belong to such subsets as "unidentifiable links." Figure 4 shows the results for a Brite and a PlanetLab topology, when 10% of the links in the network are congested.

We observe that our algorithm performs well (and better than the independence algorithm), even when *half* of the congested links are unidentifiable. This is true both for the Brite (Figure 4(b)) and the PlanetLab topology (Fig. 4(d)). Interestingly, in certain cases, we outperform the independence algorithm *even with respect to the unidentifiable links* (we do not show the corresponding graphs for lack of space). This may seem counter-intuitive at first—why



(a) CDF of the absolute error when 25% of the congested links are unindentifiable. Brite topology.

(b) CDF of the absolute error when 50% of the congested links are unindentifiable. Brite topology.

(c) CDF of the absolute error when 25% of the congested links are unindentifiable. PlanetLab topology.

(d) CDF of the absolute error when 50% of the congested links are unindentifiable. PlanetLab topology.

Figure 4: Performance of the two algorithms, when different fractions of congested links are unidentifiable. In all cases, 10% of the links are congested.

should the independence algorithm be any worse in characterizing unidentifiable links? The reason is that, in tomographic algorithms, mistakes "propagate," i.e., mischaracterizing one link leads to a cascade of link mischaracterizations; this effect is stronger when problematic links are concentrated in the same few paths. Our algorithm characterizes the identifiable links accurately, which means that it makes fewer mistakes, hence suffers less from this cascading effect, which improves its accuracy with the rest of the links as well.

Unknown Correlation Patterns. Finally, we look at the performance of the two algorithms when Assumption 4 does not hold *and* there are unknown correlation patterns in the network. In particular, we consider the scenario where a worm has infected a large number of end-hosts and periodically orders them to flood a set of otherwise uncorrelated links; as a result, these links become correlated, i.e., they get congested at about the same time. Since there is no practical way for an operator to know of this correlation pattern, we assume that it is unknown, i.e., our algorithm treats the targeted links as uncorrelated; for brevity, we refer to these links that are incorrectly labeled as uncorrelated with one another as "mislabeled." Figure 5 shows the results for a Brite and a PlanetLab topology, when 10% of the links in the network are congested.

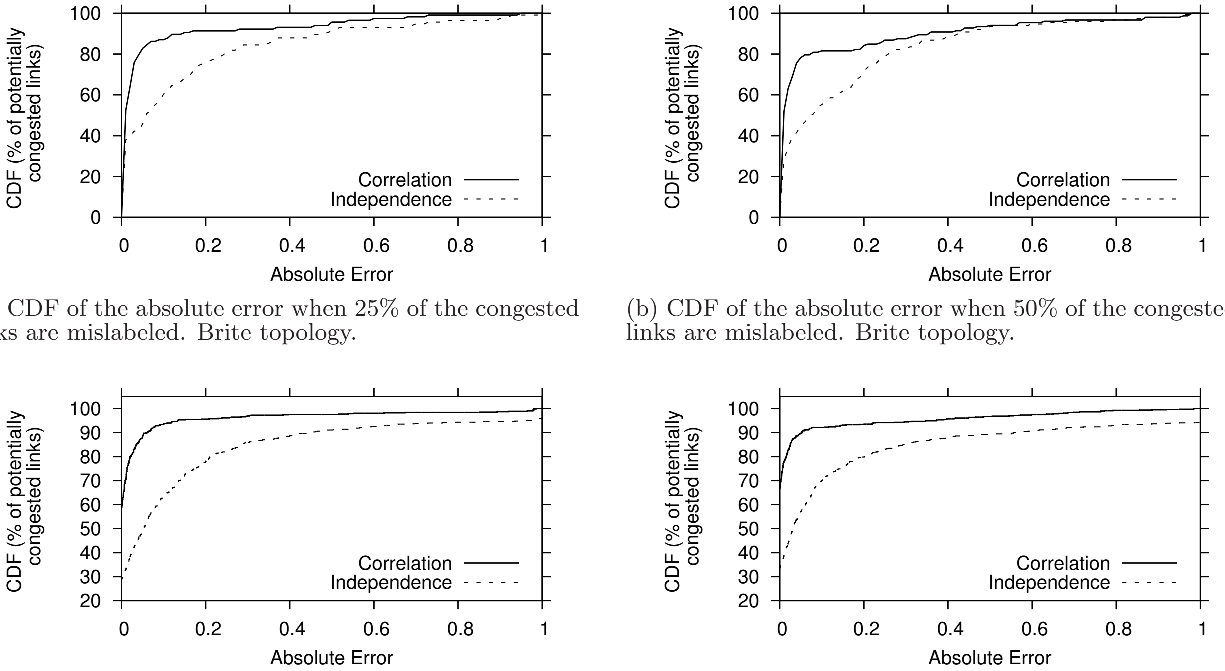
We observe that our algorithm still performs well (and better than the independence algorithm), even when *half* of the congested links participate in unknown correlation patterns (Figures 5(b) and 5(d)). Moreover, we significantly outperform the independence algorithm, even with respect to the links that are affected by the unknown correlation patterns (graphs omitted for lack of space). The reason is that we ignore one correlation pattern, whereas the independence algorithm ignores *all* correlation patterns in the network, which is compounded by the cascading effect mentioned above.

Ongoing Work: PlanetLab Tomographer. We are in the process of building a network tomographer that runs on PlanetLab nodes and infers the congestion probabilities of the links between them (we measure its accuracy through the indirect validation method proposed in [13]). Our tomographer uses traceroute to identify the sequence of routers between each pair of PlanetLab nodes, then tries to map each router to an AS number. Our plan is to run our tomographer (i) assuming that all links are uncorrelated and (ii) assuming that all links in the same AS are correlated, and compare the results; such a comparison would provide evidence regarding the effect of link correlation on network tomography in practice.

6. RELATED WORK

Network performance tomography, which infers the characteristics of links from end-to-end path measurements, is an ill-posed inverse problem well studied in the last decade. A number of methods have been proposed and validated, which differ on the assumptions made about the network and on the characteristics of links they provide.

In order to infer the characteristics of links, the initial methods rely on temporal correlation, either by sending multicast packets (which are perfectly correlated) [5, 4, 2, 3], or



(a) CDF of the absolute error when 25% of the congested links are mislabeled. Brite topology.

(b) CDF of the absolute error when 50% of the congested links are mislabeled. Brite topology.

(c) CDF of the absolute error when 25% of the congested links are mislabeled. PlanetLab topology.

(d) CDF of the absolute error when 50% of the congested links are mislabeled. PlanetLab topology.

Figure 5: Performance of the two algorithms, when different fractions of the links are mislabeled, i.e., participate in unknown correlation patterns. In all cases, 10% of the links are congested.

emulating multicast by sending unicast back to back packets (which are strongly correlated on shared links) [8, 9, 17]. All moments of link loss rates and delays (except first moments) are statistically identifiable in a multicast tree topology [7]. However, multicast is not widely deployed, and groups of unicast packets require substantial development and administrative costs, hence, it is not easy to rely on temporal correlations.

The set of methods that followed [13, 10, 19, 15], use only unicast end-to-end measurements for the simpler goal of identifying the congested links (i.e. identifying if the link loss rate or delay exceeds some threshold, instead of computing their actual value). These “Boolean” network tomographic methods use additional information or assumptions. For example, the methods in [13, 10] identify the congested links by finding the smallest set of links whose congestion can explain the observed measurements. These methods essentially use three assumptions: (i) network links are independent, (ii) links are equally likely to be congested, and (iii) the number of congested links is small.

Assumption (ii) holds for homogeneous networks, but the Internet is a heterogeneous network, and some links such as access links or peering links are often more congested than core links. Fortunately, the congestion probabilities of links can be identified from end-to-end measurements if the links are independent (i.e. if Assumption (i) holds) [12]. These probabilities can be learned from multiple network measurements, and used to locate the congested links with higher accuracy [12]. In addition, they correct the bias towards the (usually non congested) core links, since these links are shared by many more paths than access links, and therefore, by Assumption (iii) they are the smallest number of links that explains the end-to-end performance.

However, all the previous methods strongly rely on Assumption (i), that is, links are independent. This paper shows that one can lift this assumption in part, allow for some local link correlation, and still identify correctly the congestion probabilities of links. Our goal is not to determine the link correlations, we want to eliminate them as to identify the first-order marginal distribution of congested link probabilities.

Another network tomography problem that has received a lot of attention is traffic matrix estimation, which infers the volumes of end-to-end flows from link measurements [18, 6]. Here, the unknown variables are the flow volumes, and are assumed to follow a certain parametric distribution. Flow correlations have been studied by Singhal and Michailidis [14], who showed that under some classes of dependencies, the order n moments of the flow volumes are identifiable from link measurements for $n \geq 2$. Our approach differs in three aspects. First, we are interested in the “dual” problem of inferring link properties given the end-to-end measurements, whereas in [18, 6, 14] link measurements are known, but end-to-end traffic counts need to be inferred. Second, we are interested in Boolean variables, and we do not need a parametrization of the continuous distribution of the variables at stake. The different nature of the relation between the unknown quantities of interest and the measurements restrain us to extend the theoretical results in [14]. Finally, we prove identifiability of the first order moments of the quantities of interest, while the authors in [14] prove identifiability of order $n \geq 2$ moments.

7. CONCLUSION

We considered the problem of identifying the probability that each link in a network is congested from end-to-end path measurements. The key characteristic that sets our work apart from related work is that it takes into account link correlations. In particular, we considered the model where we know which links are most likely to be correlated (e.g., links from the same local-area network or the same administrative domain), however, we do not know the exact nature of the correlation (i.e., we do not assume knowledge of any correlation coefficients). We formally proved that, under certain well defined conditions, it is feasible to identify the probability that each link is congested from end-to-end path measurements, even in the presence of link correlations. We also presented an algorithm that computes these probabilities. We showed through simulations that our algorithm is accurate in a variety of congestion scenarios, even when we do not know all the correlation patterns in the network. Moreover, by comparing with a similar algorithm that does not take link correlation into account, we showed that considering link correlation matters, even when there is a low level of congestion in the network and congested links are only loosely correlated.

Acknowledgments

This work was supported by grant ManCom 2110 of the Hasler Foundation, Bern, Switzerland, and by an Ambizone grant from the Swiss National Science Foundation. We would like to thank the anonymous reviewers for their constructive feedback.

8. REFERENCES

[1] Boston University Representative Internet Topology Generator. `http://www.cs.bu.edu/brite/`.

[2] A. Adams, T. Bu, T. Friedman, J. Horowitz, D. Towstey, R. Caceres, N. Duffield, F. L. Presti, S. B. Moon, and V. Paxson. The Use of End-to-end Multicast Measurements for Characterizing Internal Network Behavior. *IEEE Communications Magazine*, May 2000.

[3] V. Arya, N. Duffield, and D. Veitch. Temporal Delay Tomography. In *Proceedings of the IEEE INFOCOM Conference*, 2008.

[4] T. Bu, N. Duffield, F. L. Presti, and D. Towsley. Network Tomography on General Topologies. In *Proceedings of the ACM SIGMETRICS Conference*, 2002.

[5] R. Caceres, N. G. Duffield, J. Horowitz, and D. Towsley. Multicast-based Inference of Network-Internal Loss Characteristics. *IEEE Transactions on Information Theory*, 45:2462–2480, 1999.

[6] J. Cao, D. Davis, S. V. Wiel, and B. Yu. Time-Varying Network Tomography: Router Link Data. *Journal of the American Statistical Association*, 95(452):1063–1075, Dec. 2000.

[7] A. Chen, J. Cao, and T. Bu. Network Tomography: Identifiability and Fourier Domain Estimation. In *Proceedings of the IEEE INFOCOM Conference*, 2007.

[8] M. Coates and R. Nowak. Network Loss Inference Using Unicast End-to-End Measurement. In *Proceedings of the ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, 2000.

[9] N. Duffield, F. L. Presti, V. Paxson, and D. Towsley. Inferring Link Loss Using Striped Unicast Probes. In *Proceedings of the IEEE INFOCOM Conference*, 2001.

[10] N. G. Duffield. Network Tomography of Binary Network Performance Characteristics. *IEEE Transactions on Information Theory*, 52(12):5373–5388, December 2006.

[11] H. X. Nguyen and P. Thiran. Network Loss Inference with Second Order Statistics of End-to-End Flows. In *Proceedings of the IEEE Internet Measurement Conference (IMC)*, 2007.

[12] H. X. Nguyen and P. Thiran. The Boolean Solution to the Congested IP Link Location Problem: Theory and Practice. In *Proceedings of the IEEE INFOCOM Conference*, 2007.

[13] V. N. Padmanabhan, L. Qiu, and H. J. Wang. Server-based Inference of Internet Performance. In *Proceedings of the IEEE INFOCOM Conference*, 2003.

[14] H. Singhal and G. Michailidis. Identifiability of Flow Distributions from Link Measurements with Applications to Computer Networks. *Inverse Problems*, 23:1821–1850, 2007.

[15] J. Sommers, P. Barford, N. Duffield, and A. Ron. Accurate and Efficient SLA Compliance Monitoring. In *Proceedings of the ACM SIGCOMM Conference*, 2007.

[16] H. H. Song, L. Qiu, and Y. Zhang. NetQuest: A Flexible Framework for Large-Scale Network Measurement. In *Proceedings of the ACM SIGMETRICS Conference*, 2006.

[17] Y. Tsang, M. Coates, and R. Nowak. Passive Network Tomography Using the EM Algorithms. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2001.

[18] Y. Vardi. Network Tomography: Estimating Source-Destination Traffic Intensities. *Journal of the American Statistical Association*, 91:365–377, 1996.

[19] Y. Zhao, Y. Chen, and D. Bindel. Toward Unbiased End-to-End Network Diagnosis. In *Proceedings of the ACM SIGCOMM Conference*, 2006.

APPENDIX

A. PROOF OF THEOREM 1

A.1 Definitions and Notation

We introduce three more symbols. All our symbols are summarized in Table 1.

Network States. We denote by $\mathcal{S}_n$ a particular value of $\mathcal{S}$ and by $S^p_n \subseteq C_p$ the corresponding value of S^p (i.e., the set of congested links in correlation set C_p when $\mathcal{S} = \mathcal{S}_n$). We can write:

$$\mathcal{S}_n \equiv \bigcup_{p=1..|\mathcal{C}|} S^p_n. \tag{11}$$

During any particular snapshot, each correlation set (as well as the entire network) can only be in one state, hence, any two states $S^p_n, S^p_m, n \neq m$, (as well as $\mathcal{S}_n, \mathcal{S}_m, n \neq m$,) are mutually exclusive.

Ordering of Path Sets. Consider two correlation subsets $A, B \in \tilde{\mathcal{C}}$. We define the precedence property as:

$$A \prec B \quad \equiv \quad |\psi(A)| < |\psi(B)| \tag{12}$$

that is, the edges in B are traversed by more paths than the edges in A. Using this definition, we can determine a partial ordering $\mathcal{T}$ of all the correlation subsets in $\tilde{\mathcal{C}}$, i.e., order them by the number of paths that traverse them.

$\mathcal{G}$	the network graph
$\mathcal{V}$	the set of all nodes
$\mathcal{E}$	the set of all links
$\mathcal{P}$	the set of all paths
$e_k \in \mathcal{E}$	a link
$P_i \in \mathcal{P}$	a path
$e_k \in P_i$	link e_k is traversed by path P_i
$\psi(A)$	the paths "covered" by the set of links A
$\lvert\psi(A)\rvert$	the number of paths in $\psi(A)$
X_{e_k}	(r.v.) the state of link e_k
Y_{P_i}	(r.v.) the state of path P_i
$\mathcal{C}$	partition of $\mathcal{E}$ into correlation sets
$C_p \in \mathcal{C}$	a correlation set
$\tilde{\mathcal{C}}$	the set of all correlation subsets
$A, B \in \tilde{\mathcal{C}}$	two correlation subsets
S^p	(r.v.) the set of congested links in C_p
$\mathcal{S}$	(r.v.) the set of all congested links
$\psi(\mathcal{S})$	(r.v.) the set of all congested paths
α_A	congestion factor of $A \in \tilde{\mathcal{C}}$
$\mathcal{S}_n$	a value of the random set $\mathcal{S}$
S_n^p	a value of the random set S^p
$A \prec B$	$\lvert\psi(A)\rvert < \lvert\psi(B)\rvert$

Table 1: List of defined symbols. "r.v." = "random variable."

A.2 Some Basic Probabilities

Network State Probability. We will first express the probability that the network is in state $\mathcal{S}_n$. We will use Eq. 11 and the fact that S^p and $S^q, p \neq q$, are independent.

$$\mathbb{P}(\ \mathcal{S} \ = \ \mathcal{S}_n\ ) = \mathbb{P}\left(\bigcap_{p=1..|\mathcal{C}|} (\ S^p = S_n^p\ )\right)$$
$$= \prod_{p=1..|\mathcal{C}|} \mathbb{P}(\ S^p = S_n^p\ ). \qquad (13)$$

All Paths Are Good. Next, we will express the probability that all paths in $\mathcal{P}$ are good. We will use Assumption 2, in particular, its implication that, if all paths in $\mathcal{P}$ are good, then necessarily all links in $\mathcal{E}$ are good.

$$\mathbb{P}(\ \psi(\mathcal{S}) = \emptyset\ ) = \mathbb{P}(\ \mathcal{S} = \emptyset\ )$$
$$= \prod_{p=1..|\mathcal{C}|} \mathbb{P}(\ S^p = \emptyset\ ). \qquad (14)$$

Some Paths Are Congested. Next, we will express the probability of event $\psi(\mathcal{S}) = \psi(A)$, i.e., that the paths covered by correlation subset A are the only congested paths in the network.

$$\mathbb{P}(\ \psi(\mathcal{S}) = \psi(A)\ ) = \sum_{\substack{n \text{ s.t.} \\ \psi(\mathcal{S}_n)=\psi(A)}} \mathbb{P}(\ \mathcal{S} = \mathcal{S}_n\ ). \qquad (15)$$

Eq. 15 expresses the fact that, when the paths in $\psi(A)$ are the only congested paths, it must be true that a set of links that cover exactly these paths (for which $\psi(\mathcal{S}_n) = \psi(A)$) are the only congested links. One such set of links is A, but there may be other sets that consist of multiple correlation subsets.

We will now develop Eq. 15 further by considering the following: Since $A \in \tilde{\mathcal{C}}$, there exists one correlation set $C_q \in \mathcal{C}$, which contains A. In the following, we denote by q the index of the correlation set C_q such that $A \subseteq C_q$. We partition the possible network states in two sets: the states where $A = S_n^q$ (the only congested links in C_q are the links in A) and the states where $A \neq S_n^q$:

$$\mathbb{P}(\ \psi(\mathcal{S}) = \psi(A)\ ) = \sum_{\substack{n \text{ s.t. } S_n^q=A, \\ \psi(\mathcal{S}_n)=\psi(A)}} \mathbb{P}(\ \mathcal{S} = \mathcal{S}_n\ )$$
$$+ \sum_{\substack{n \text{ s.t. } S_n^q\neq A, \\ \psi(\mathcal{S}_n)=\psi(A)}} \mathbb{P}(\ \mathcal{S} = \mathcal{S}_n\ ). \qquad (16)$$

Finally, if we combine Eq. 16 with Eq. 13, we get:

$$\mathbb{P}(\ \psi(\mathcal{S}) = \psi(A)\ ) =$$
$$\mathbb{P}(\ S^q = A\ ) \sum_{\substack{n \text{ s.t. } S_n^q=A, \\ \psi(\mathcal{S}_n)=\psi(A)}} \left(\prod_{\substack{p=1..|\mathcal{C}|, \\ p\neq q}} \mathbb{P}(\ S^p = S_n^p\ )\right)$$
$$+ \sum_{\substack{n \text{ s.t. } S_n^q\neq A, \\ \psi(\mathcal{S}_n)=\psi(A)}} \left(\prod_{p=1..|\mathcal{C}|} \mathbb{P}(\ S^p = S_n^p\ )\right). \qquad (17)$$

Illustration. Consider Figure 1(a), correlation subset $A = \{e_1, e_2\}$, and the event $(\psi(\mathcal{S}) = \psi(A) = \{P_1, P_2, P_3\})$, i.e., that all paths are congested. This means that the network is in one of the following states:

$\mathcal{S}_n$	S_n^1	$\cup$	S_n^2	$\cup$	S_n^3
$\mathcal{S}_1$	$\{e_1, e_2\}$	$\cup$	$\emptyset$	$\cup$	$\emptyset$
$\mathcal{S}_2$	$\{e_1, e_2\}$	$\cup$	$\{e_3\}$	$\cup$	$\emptyset$
$\mathcal{S}_3$	$\{e_1, e_2\}$	$\cup$	$\emptyset$	$\cup$	$\{e_4\}$
$\mathcal{S}_4$	$\{e_1, e_2\}$	$\cup$	$\{e_3\}$	$\cup$	$\{e_4\}$
$\mathcal{S}_5$	$\emptyset$	$\cup$	$\{e_3\}$	$\cup$	$\{e_4\}$
$\mathcal{S}_6$	$\{e_1\}$	$\cup$	$\{e_3\}$	$\cup$	$\{e_4\}$
$\mathcal{S}_7$	$\{e_2\}$	$\cup$	$\{e_3\}$	$\cup$	$\{e_4\}$
$\mathcal{S}_8$	$\{e_2\}$	$\cup$	$\{e_3\}$	$\cup$	$\emptyset$

In this particular case, $A \subseteq C_1$, i.e., $q = 1$. For the first four states, $S_n^1 = A$, whereas for the rest, $S_n^1 \neq A$. Hence, if we apply Equations 16 and 17, we get:

$$\mathbb{P}(\ \psi(\mathcal{S}) = \psi(\{e_1, e_2\})\ ) =$$
$$\sum_{n=1}^{4} \mathbb{P}(\ \mathcal{S} = \mathcal{S}_n\ ) + \sum_{n=5}^{8}\ \mathcal{S} = \mathcal{S}_n\ ) =$$
$$\begin{array}{lll}
\mathbb{P}(\ S^1 = \{e_1, e_2\}\ ) & \mathbb{P}(\ S^2 = \emptyset\ ) & \mathbb{P}(\ S^3 = \emptyset\ ) + \\
\mathbb{P}(\ S^1 = \{e_1, e_2\}\ ) & \mathbb{P}(\ S^2 = \{e_3\}\ ) & \mathbb{P}(\ S^3 = \emptyset\ ) + \\
\mathbb{P}(\ S^1 = \{e_1, e_2\}\ ) & \mathbb{P}(\ S^2 = \emptyset\ ) & \mathbb{P}(\ S^3 = \{e_4\}\ ) + \\
\mathbb{P}(\ S^1 = \{e_1, e_2\}\ ) & \mathbb{P}(\ S^2 = \{e_3\}\ ) & \mathbb{P}(\ S^3 = \{e_4\})\ ) + \\
\mathbb{P}(\ S^1 = \emptyset\ ) & \mathbb{P}(\ S^2 = \{e_3\}\ ) & \mathbb{P}(\ S^3 = \{e_4\}\ ) + \\
\mathbb{P}(\ S^1 = \{e_1\}\ ) & \mathbb{P}(\ S^2 = \{e_3\}\ ) & \mathbb{P}(\ S^3 = \{e_4\}\ ) + \\
\mathbb{P}(\ S^1 = \{e_2\}\ ) & \mathbb{P}(\ S^2 = \{e_3\}\ ) & \mathbb{P}(\ S^3 = \{e_4\}\ ) + \\
\mathbb{P}(\ S^1 = \{e_2\}\ ) & \mathbb{P}(\ S^2 = \{e_3\}\ ) & \mathbb{P}(\ S^3 = \emptyset\ ).
\end{array}$$

A.3 Proof

If we divide Eq. 17 by Eq. 14, we obtain:

$$\frac{\mathbb{P}(\ \psi(\mathcal{S}) = \psi(A)\ )}{\mathbb{P}(\ \psi(\mathcal{S}) = \emptyset\ )} = \alpha_A\, \Gamma_A + \Gamma_{\bar{A}} \tag{18}$$

where

$$\Gamma_A \ = \sum_{\substack{n \text{ s.t. } S_n^q = A,\\ \psi(\mathcal{S}_n)=\psi(A)}} \left(\prod_{\substack{p=1..|\mathcal{C}|,\\ p\neq q}} \alpha_{S_n^p} \right)$$

and

$$\Gamma_{\bar{A}} \ = \sum_{\substack{n \text{ s.t. } S_n^q \neq A,\\ \psi(\mathcal{S}_n)=\psi(A)}} \left(\prod_{p=1..|\mathcal{C}|} \alpha_{S_n^p} \right).$$

Lemma 1. *The terms Γ_A and $\Gamma_{\bar{A}}$ depend on congestion factors $\alpha_{S_n^p}$, where n and p are such that $S_n^p = \emptyset$ or $S_n^p \prec A$.*

Proof. We know that Γ_A and $\Gamma_{\bar{A}}$ depend on congestion factors $\alpha_{S_n^p}$, where n and p are such that $S_n^p \neq A$. This is because: Γ_A depends on congestion factors $\alpha_{S_n^p}$, where n is such that $S_n^q = A$ and $p \neq q$. $\Gamma_{\bar{A}}$ depends on congestion factors $\alpha_{S_n^p}$, where n is such that $S_n^q \neq A$.

Moreover, Γ_A and $\Gamma_{\bar{A}}$ depend on congestion factors $\alpha_{S_n^p}$, where n is such that $\psi(\mathcal{S}_n) = \psi(A)$. Since $\psi(S_n^p) \subseteq \psi(\mathcal{S}_n)$, it implies that $\psi(S_n^p) \subseteq \psi(A)$. We distinguish two cases:

1. $\psi(S_n^p) = \psi(A)$. Correlation subsets S_n^p and A cover exactly the same paths. From Assumption 4, it follows that $S_n^p = A$, which contradicts the fact that $S_n^p \neq A$.
2. $\psi(S_n^p) \subset \psi(A)$. Correlation subset S_n^p covers fewer paths than A, i.e., $|\psi(S_n^p)| < |\psi(A)|$. Thus, either $S_n^p = \emptyset$ or, by definition, $S_n^p \prec A$.

□

Lemma 2. *The congestion factors α_A are identifiable, for all $A \in \tilde{\mathcal{C}}$.*

Proof. We prove the lemma by induction on the partial ordering $\mathcal{T}$.

Initial Step. Let A be the first element from the partial ordering $\mathcal{T}$ defined by Eq 12. We will prove that we can compute α_A from Eq. 18.

Γ_A and $\Gamma_{\bar{A}}$ consist of terms $\alpha_{S_n^p}$, where n is such that $\psi(\mathcal{S}_n) = \psi(A)$ (call this Condition 1). Moreover, from Lemma 1, we know that n and p are such that $S_n^p = \emptyset$ or $S_n^p \prec A$; since A is the first element in $\mathcal{T}$, it cannot be that $S_n^p \prec A$, hence, $S_n^p = \emptyset$ (call this Condition 2). The only network state that satisfies Conditions 1 and 2 is $\mathcal{S}_n = S_n^q = A$.

Hence, we have:

$$\begin{aligned}\Gamma_A &= \sum_{\substack{n \text{ s.t. } S_n^q = A\\ \mathcal{S}_n = A}} \left(\prod_{\substack{p=1..|\mathcal{C}|,\\ p\neq q}} \alpha_{S_n^p} \right)\\ &= \prod_{\substack{p=1..|\mathcal{C}|,\\ p\neq q}} \frac{\mathbb{P}(\ S^p = \emptyset\ )}{\mathbb{P}(\ S^p = \emptyset\ )} = 1\end{aligned}$$

and

$$\Gamma_{\bar{A}} = \sum_{\substack{n \text{ s.t. } S_n^q \neq A\\ \mathcal{S}_n = A}} \left(\prod_{p=1..|\mathcal{C}|} \alpha_{S_n^p} \right) = 0.$$

Thus, we can compute α_A from Eq. 18, where the term on the left hand side is obtained through end-to-end measurements.

Induction Step. We assume that we know α_B, for all $B \prec A$. We will prove that we can compute α_A.

Let us recast Eq. 18 as

$$\alpha_A = \frac{\frac{\mathbb{P}(\ \psi(\mathcal{S})=\psi(A)\ )}{\mathbb{P}(\ \psi(\mathcal{S})=\emptyset\ )} - \Gamma_{\bar{A}}}{\Gamma_A}.$$

Note that the denominator Γ_A is never 0, because for any correlation subset A, there always exists at least one state $\mathcal{S}_n$ such that $\psi(\mathcal{S}_n) = \psi(A)$ and $S_n^q = A$, which is $\mathcal{S}_n = A$.

According to Lemma 1, Γ_A and $\Gamma_{\bar{A}}$ depend on congestion factors $\alpha_{S_n^p}$, where n and p are such that either $S_n^p = \emptyset$ or $S_n^p \prec A$. If $S_n^p = \emptyset$, then by definition, $\alpha_{S_n^p} = \frac{\mathbb{P}(\ S^p=\emptyset\ )}{\mathbb{P}(\ S^p=\emptyset\ )} = 1$. If $S_n^p \prec A$ with $S_n^p \neq \emptyset$, from the induction hypothesis, we know $\alpha_{S_n^p}$. Thus, we can compute α_A from Eq. 18, where the term on the left is obtained through end-to-end measurements. □

Lemma 3. *If the congestion factors α_A are known for all $A \subseteq C_p$, then the probability $\mathbb{P}(\ X_{e_k} = 1\ )$ is identifiable for all $e_k \in C_p$.*

Proof. As

$$\begin{aligned}\mathbb{P}(\ S^p = \emptyset\ ) &= 1 - \mathbb{P}(\ S^p \neq \emptyset\ )\\ &= 1 - \sum_{A\subseteq C_p, A\neq\emptyset} \mathbb{P}(\ S^p = A\ )\\ &= 1 - \sum_{A\subseteq C_p, A\neq\emptyset} \alpha_A\, \mathbb{P}(\ S^p = \emptyset\ )\end{aligned}$$

we have that

$$\mathbb{P}(\ S^p = \emptyset\ ) = \frac{1}{1 + \sum_{A\subseteq C_p, A\neq\emptyset} \alpha_A}.$$

Since α_A is known for all $A \in C_p$, we can compute $\mathbb{P}(\ S^p = \emptyset\ )$ from the above equation.

Furthermore, we can compute $\mathbb{P}(\ S^p = A\ )$ for all $A \subseteq C_p, A \neq \emptyset$:

$$\mathbb{P}(\ S^p = A\ ) = \alpha_A \cdot \mathbb{P}(\ S^p = \emptyset\ )$$

and $\mathbb{P}(\ X_{e_k} = 1\ )$ for all $e_k \in C_p$:

$$\mathbb{P}(\ X_{e_k} = 1\ ) = \sum_{\substack{A\subseteq C_p\\ e_k\in A}} \mathbb{P}(\ S^p = A).$$

□

We have proved that the congestion factors α_A are identifiable for all correlation subsets $A \in \tilde{\mathcal{C}}$ (Lemma 2). We have also proved that, if the congestion factors α_A are known for all $A \subseteq C_p$, the probability that all the links in A are congested is identifiable for all $A \subseteq C_p$ (Lemma 3), which proves Theorem 1.