A Network Coding Approach to Loss Tomography

Pegah Sattari, *Student Member, IEEE*, Athina Markopoulou, *Senior Member, IEEE*, Christina Fragouli, *Member, IEEE*, and Minas Gjoka

***Abstract*—Network tomography aims at inferring internal network characteristics based on measurements at the edge of the network. In loss tomography, in particular, the characteristic of interest is the loss rate of individual links and multicast and/or unicast end-to-end probes are typically used. Independently, recent advances in network coding have shown that there are advantages from allowing intermediate nodes to process and combine, in addition to just forward, packets. In this paper, we study the problem of loss tomography in networks with network coding capabilities. We design a framework for estimating link loss rates, which leverages network coding capabilities, and we show that it improves several aspects of tomography, including the identifiability of links, the trade-off between estimation accuracy and bandwidth efficiency, and the complexity of probe path selection. We discuss the cases of inferring link loss rates in a tree topology and in a general topology. In the latter case, the benefits of our approach are even more pronounced compared to standard techniques but we also face novel challenges, such as dealing with cycles and multiple paths between sources and receivers. Overall, this work makes the connection between active network tomography and network coding.**

***Index Terms*—Link loss inference, network coding, network tomography.**

I. Introduction

DISTRIBUTED Internet applications often need to know information about the characteristics of the network. For example, an overlay or peer-to-peer network may want to detect and recover from failures or degraded performance of the underlying Internet infrastructure. A company with several geographically distributed campuses may want to know the behavior of one or several Internet service providers (ISPs) connecting the campuses, in order to optimize traffic engineering decisions and achieve the best end-to-end performance. To achieve this high-level goal, it is necessary for the nodes participating in the application or overlay to monitor Internet paths, assess and predict their behavior, and eventually make efficient use of them by taking appropriate control and traffic engineering decisions both at the network and at the application layers. Therefore, accurate monitoring at minimum overhead and complexity is of crucial importance in order to provide the input needed to take such informed decisions. However, there is currently no incentive for ISPs to provide detailed information about their internal operation and performance or to collaborate with other ISPs for this purpose. As a result, distributed applications usually rely on their own end-to-end measurements between nodes they have control over, in order to infer performance characteristics of the network.

Over the past decade, a significant research effort has been devoted to a class of monitoring problems that aim at inferring internal network characteristics using measurements at the edge [1]. This class of problems is commonly referred to as *tomography* due to its analogy to medical tomography. In this work, we are particularly interested in loss tomography, *i.e.*, inferring the loss probabilities (or loss rates) of individual links using active end-to-end measurements [2]–[6]. The topology is assumed known and sequences of probes are sent and collected between a set of sources and a set of receivers at the network edge. Link-level parameters, in this case loss rates of links, are then inferred by the observations at the receivers. The bandwidth efficiency of these methods can be measured by the number of probes needed to estimate the loss rates of interest within a desired accuracy. Despite its significance and the research effort invested, loss tomography remains a hard problem for a number of reasons, including complexity (of optimal probe routing and of estimation), bandwidth overhead, and identifiability (the fundamental fact that tomography is an inverse problem and we cannot directly observe the parameters of interest). Moreover, there are some practical limitations such as the lack of cooperation of ISPs, the need for synchronization of sources in some schemes, etc.

Recently, a new paradigm to routing information has emerged with the advent of network coding [7]–[9]. The main idea in network coding is that, if we allow intermediate nodes to not only forward but also combine packets, we can obtain significant benefits in terms of throughput, delay, and robustness of distributed algorithms. Our work is based on the observation that, in networks equipped with network coding capabilities, we can leverage these capabilities to significantly improve several aspects of loss tomography. For example, with network coding, we can combine probes from different paths into one, thus reducing the bandwidth needed to cover a general graph and also increasing the information per packet. Furthermore, the problem of optimal probe routing, which is known to be NP-hard, can be solved with linear complexity when network coding is used.

Manuscript received February 01, 2009; revised February 15, 2012; accepted August 08, 2012. Date of publication December 28, 2012; date of current version February 12, 2013. This work was supported by the following grants: National Science Foundation CAREER Award 0747110, the Air Force Office of Scientific Research (AFOSR) Multi-University Research Initiative under Grant FA9550-09-0643, the AFOSR under Grant FA9550-10-1-030, the Swiss National Science Foundation under Award PP00P2-128639, and the European Research Council under Grant ERC-2009-StG-240317.

P. Sattari was with the Department of Electrical Engineering and Computer Science, University of California, Irvine, CA 92697 USA. She is now with Jeda Networks, Newport Beach, CA 92660 (e-mail: psattari@uci.edu).

A. Markopoulou and M. Gjoka are with the Department of Electrical Engineering and Computer Science, University of California, Irvine, CA 92697 USA (e-mail: athina@uci.edu; mgjoka@uci.edu).

C. Fragouli is with the School of Computer and Communication Sciences, EPFL, CH-1015 Lausanne, Switzerland (e-mail: christina.fragouli@epfl.ch).

Communicated by R. A. Berry, Associate Editor for Communication Networks.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2012.2236916

This paper proposes a framework for loss tomography (including mechanisms for probe routing, probe and code design, estimation, and identifiability guarantees) in networks that already have network coding capabilities. Such capabilities do not exist yet on the Internet today, but are available in wireless mesh networks, peer-to-peer and overlay networks, and we expect them to appear in more environments as network coding becomes more widely adopted. We show that, in those settings, our network coding-based approach improves the following aspects of the loss tomography problem: how many links of the network we can infer (identifiability); the tradeoff between how well we can infer link loss rates (estimation accuracy) and how many probes we need in order to do so (bandwidth efficiency); how to select sources and receivers and how to route probes between them (optimal probe routing). Overall, this is a novel application of network coding techniques to a practical networking problem, and it opens a promising research direction.

The structure of this paper is as follows. Section II discusses related work. Section III states the problem and summarizes the challenges and main results. Section IV presents a motivating example and provides the conditions of identifiability. Sections V and VI present in detail the framework and mechanisms in the cases of trees and general topologies, respectively. Section VII concludes this paper.

II. Related Work

Network Tomography: The term network tomography typically refers to a family of problems that aim at inferring internal network characteristics from measurements at the edge of the network. Internal characteristics of interest may include link-level parameters (such as loss and delay metrics) or the network topology. Another type of tomography problem aims at inferring path-level traffic intensity (*e.g.*, traffic matrices) from link-level measurements [10]. Our paper focuses on inferring the loss rates of internal links using active end-to-end measurements and assuming that the topology is known. Therefore, it is related to the literature on loss tomography, part of which is discussed in the following.

Caceres *et al.* considered a single multicast tree (MT) with a known topology and inferred the link loss rates from the receivers' observations [2]. In particular, they developed a low-complexity algorithm to compute the maximum likelihood estimator (MLE), by taking into account the dependences introduced by the tree hierarchy to factorize the likelihood function and eventually compute the MLE in a recursive way. Throughout this paper, we refer to the MLE for an MT, developed in [2], as MINC, and we build on it. Bu *et al.* used multiple MTs to cover a general topology and proposed an EM algorithm for link loss rate estimation [3]. Follow-up approaches have been developed for unicast probes [5], [6], joint inference of topology and link loss rates [4], and adaptive tomography and delay inference [11]. The aforementioned list of references is not comprehensive. Good surveys of network tomography can be found in [1] and [12].

Active Versus Passive Tomography: Tomography can be based either on active (generating probe traffic) or on passive (monitoring traffic flows and sampling existing traffic) measurements. Passive approaches have been most commonly used for estimating path-level information, in particular, origin-destination traffic matrices, from data collected at various nodes of the network [10]. This approach and problem statement are well suited for the needs of a network provider. For the problem of inferring link loss rates, active probes are typically used, and information about individual packets received or lost is analyzed at the edge of the network. This approach is better suited for end users that do not have access to the network. However, there are also papers that study link loss inference by using existing traffic flows to sample the state of the network [13], [14]. Once measurements have been collected following either of the two methods, statistical inference techniques are applied to determine network characteristics that are not directly observed.

The passive approach has the advantage that it does not impose additional burden on the network and that it measures the actual loss experienced by real traffic. However, it must also ensure that the characteristics of the traffic (*e.g.*, TCP) do not bias the sample. In the active approach, one has more control over designing the probes, which can thus be optimized for efficient estimation. The downside is that we inject measurement traffic that may increase the load of the network, may be treated differently than regular traffic, or may even be dropped, *e.g.*, due to security concerns.

Network Coding and Inference: An extensive body of work on network coding [9], [15] has emerged after the seminal work of Ahlswede *et al.* [7] and Li *et al.* [8]. The main idea in network coding is that, if we allow intermediate nodes to not only forward but also combine packets, we can realize significant benefits in terms of throughput, delay, and robustness of distributed algorithms. Within this large body of work, closer to ours are a few papers that leverage the headers of network coded packets for passive inference of properties of a network. In [16], Ho *et al.* showed how information contained in network codes can be used for passive inference of possible locations of link failures or losses. In [17], Sharma *et al.* considered random intrasession network coding and showed that nodes can passively infer their upstream network topology, based on the headers of the received coded packets they observe (which play essentially the role of probes). The main idea is that the transfer matrix (*i.e.*, the linear transform from the sender to the receiver) is distinct for different networks, with high probability. All possible transfer matrices are enumerated, and matched to the observed input/output, and a large finite field is used to ensure that all topologies remain distinguishable. An extended version of this work to erroneous networks is provided by Yao *et al.* in [18], where different (ergodic or adversarial) failures lead to different transfer functions. The approach in [17] and [18] has the advantage of keeping the measurement bandwidth low (not higher than the transmission of coefficients, which is anyway required for data transfer with network coding) and the disadvantage of high complexity. In [19], Jafarisiavoshani *et al.* considered peer-to-peer systems and used subspace nesting structures to passively identify local bottlenecks. Similar to these papers, we leverage network coding operations for inference; in contrast to these papers, which use the headers of network-coded packets for passive inference of topology, we use the contents of active probes for inference of link loss rates.

Our Work: We make the connection between active network tomography and network coding capabilities. In [20], we introduced the basic idea of leveraging network coding capabilities to improve network monitoring. In [21], we studied link loss estimation in tree topologies. In [22], we extended the approach to general graphs. In [23], we built on MINC [2], and we provided the MLEs of the loss rates for all links simultaneously, in multiple-source tree topologies with multicast and network coding; similarly to MINC, we presented an efficient algorithm for computing the MLEs, we proved the correctness, and we analyzed the rate of convergence. This paper combines ideas from these preliminary conference papers into a common framework, and extends them by a more in-depth analysis of identifiability, routing, estimation, and code design.

Our approach is active in that probes are sent/received from/to the edge of the network and observations at the receivers are used for statistical inference. Intermediate nodes forward packets using unicast, multicast, and simple coding operations. However, the operations at the intermediate nodes need to be set up once, fixed for all experiments, and be known for inference. Therefore, our approach requires more support from the network than traditional tomography, for the benefit of more accurate/efficient estimation. Our methods may also be applicable to passive tomography, where instead of sending specialized probes, one can view the coding coefficients on a network coded packet as the "probe," thus overloading them with both communication and tomographic goals, as is the case in [17] and [18]. In this paper, we focus exclusively on the tomographic goals by taking an active approach *i.e.*, sending, collecting, and analyzing specialized probes for tomography.

III. Problem Statement

A. Model and Definitions

1) Network and Monitoring Scheme: We consider a network represented as a graph $G = (V, E)$, where V is the set of nodes and E is the set of edges corresponding to *logical links*.[1] We use the notation $e = AB$ for the link e connecting vertex A to vertex B. We assume that G has no self-loops and that there is a loss rate associated with every edge in G.[2] The topology $G = (V, E)$ is assumed to be known.

We assume that packet loss on a link $e \in E$ is i.i.d Bernoulli with probability $0 \leq \overline{\alpha}_e < 1$, where $\overline{\alpha}_e = 1 - \alpha_e$, and α_e is the success probability of link e. Losses are assumed to be independent across links. Let $\alpha = (\alpha_e)_{e \in E}$ be the vector of the link success probabilities.[3] In loss tomography, we are interested in estimating all or a subset of the parameters in α. We use additional notation for the case of tree topologies, as we explain in Section V-B1.

[1]A logical link results from combining several consecutive physical links into a single link. This results in a graph G where every intermediate vertex has degree at least three, and in-degree and out-degree at least one. This is a standard assumption in the tomography literature, which is imposed for identifiability purposes, as discussed after Definition 2.

[2]In general, the loss rates in the two directions of an edge can be different, as it is the case on the Internet due to different congestion levels.

[3]Note that the notation α refers to the vector of all success probabilities, and α_e refers to the success probability of an individual edge e.

A set S of $|S| = M$ source nodes in the periphery of the network can inject probe packets, while a set R of $|R| = N$ receivers can collect such packets. Several problem variations in the choice of sources and receivers are possible, and we will discuss the following in this paper: 1) the set of sources and the set of receivers are given and fixed; 2) a set of nodes that can act as either sources or receivers is given (and we can select among them); 3) we are allowed to select any node to act as a source or a receiver. We assume that intermediate nodes are equipped with unicast, multicast, and network coding capabilities. Probe packets are routed and coded inside the network following specific paths and according to specified coding operations. We assume that the packets incur zero transmission, propagation, and processing delay as they travel through the network. The routes selected and the operations the intermediate nodes perform are part of the design of the tomography scheme: they are chosen once at setup time and are kept the same throughout all experiments; all operations of intermediate nodes are known during estimation. For the theoretical results of this paper, we focus on *synchronized acyclic networks with zero delay*;[4] for cyclic networks, we convert them to acyclic networks by a proper choice of routing and sources/receivers.

In general, a probe packet is a vector of M symbols, with each symbol being in a finite field F_q. This includes as special cases: scalar network coding (for $M = 1$), operations over binary vectors (for $q = 2$), and more generally, vector network coding (for $M > 1$).[5] In one *experiment*, we send probes from all sources and we collect probes at the receivers: each source $S_i \in S$ injects one probe packet x_i in the network, and each receiver $R_j \in R$ receives one probe X_j. The observations at all receivers R is a vector $X_{(R)} = (X_1, X_2, \ldots X_N)$ in the space $\Omega \subseteq (F_{q^M})^N$. For a given set of link success probabilities $\alpha = (\alpha_e)_{e \in E}$, the probability distribution of all observations $X_{(R)}$ will be denoted by P_α. The probability mass function for a single observation $x \in \Omega$ is $p(x; \alpha) = P_\alpha(X_{(R)} = x)$.

To estimate the success rates of links, we perform a sequence of n independent experiments. Let $n(x)$ denote the number of probes for which the observation $x \in \Omega$ is obtained, where $\sum_{x \in \Omega} n(x) = n$. The probability of n independent observations $x^1, \ldots, x^n$ (each $x^t = (x_k^t)_{k \in R}$) is

$$p(x^1, \ldots, x^n; \alpha) = \prod_{t=1}^{n} p(x^t; \alpha) = \prod_{x \in \Omega} p(x; \alpha)^{n(x)}. \quad (1)$$

It is convenient to work with the log-likelihood function, which calculates the logarithm of this probability

$$\mathcal{L}(\alpha) = \log p(x^1, \ldots, x^n; \alpha) = \sum_{x \in \Omega} n(x) \log p(x; \alpha). \quad (2)$$

[4]Note that the link delays will only affect where the probe packets would meet in the network; they will not affect our general model.

[5]What is important is that a probe can take one of the q^M possible values. We note, however, that there is an equivalence between operations with elements in a finite field and operations with vectors of appropriate length. For example, in [24], the multicast scenario was considered, and scalar network coding over a finite field of size 2^M was used equivalently to vector network coding over the space of binary vectors of length M. Thinking in terms of one of the aforementioned special cases is appropriate in special topologies, as we will see, *e.g.*, in tree and reverse tree topologies, where scalars and binary vectors are used, respectively.

We make two assumptions, which are both realistic in practice and standard in the tomography literature.

1) We perform sufficient measurements so that each observation $x \in \Omega$ at the receivers occurs at least once, *i.e.*, $n(x) > 0$. This ensures that no term in the likelihood function becomes a constant (due to a zero exponent). Note that the final equality in (1) and (2) is valid due to this assumption.
2) The probability of loss $\overline{\alpha}_i$ on a link i is not 1, *i.e.*, $\overline{\alpha}_i \in [0, 1)$. This ensures that the log-likelihood function is well defined and differentiable.

The goal is to use the observations at the receivers, the knowledge of the network topology, and the knowledge of the routing/coding scheme to estimate the success rates of internal links of interest. We may be interested in estimating the success rate on a subset of links, or on all the links.

Definition 1: A *monitoring scheme* for a given graph G refers to a set of M source nodes, a set of N receivers, a set of paths that connect the sources to the receivers, the probe packets that sources send, and the operations that intermediate nodes perform on these packets.

We use the notion of link identifiability as it was defined in [2, Th. 3, Condition (i)]:

Definition 2: A link e is called *identifiable* under a given monitoring scheme iff: $\alpha, \alpha' \in (0, 1]^{|E|}$ and $P_\alpha = P_{\alpha'}$ implies $\alpha_e = \alpha'_e$.

To illustrate the concept, consider two consecutive links $e_1 = AB$ and $e_2 = BC$ in a row, where node B has degree 2, and is neither a source nor a receiver. These links are not identifiable, as maximizing the log-likelihood function would only allow us to identify the value of the product $\alpha_{e_1}\alpha_{e_2}$ and, thus, would lead to an infinite number of solutions. This is because, it is not possible to distinguish whether a packet gets dropped on link e_1 or e_2. Note, however, that the case of having two links in a row is ruled out by our assumption of working on a graph with logical links (all vertices in the graph have degree three or greater). Another case that e_1 and e_2 are not identifiable, which is possible even on a graph with logical links, is when both links belong to every path used from any source to any receiver.

Identifiability is not only a property of the network topology, but also depends on the monitoring scheme. One of the main goals of the monitoring scheme design is to maximize the number of identifiable links. However, our definition of identifiability does not depend on the estimator employed. Essentially, identifiability depends on the probability distribution P_α and on whether this uniquely determines α.

Estimation: The MLE $\breve{\alpha}$ identifies the parameters $(\alpha_e)_{e\in E}$ that maximize the probability of the observations $\mathcal{L}(\alpha)$

$$\breve{\alpha} = \text{argmax}_{\alpha\in(0,1]^{|E|}} \mathcal{L}(\alpha). \tag{3}$$

Candidates for the MLE are the solutions $\hat{\alpha}$ of the *likelihood equation*

$$\frac{\partial \mathcal{L}}{\partial \alpha_e}(\alpha) = 0, \quad e \in E. \tag{4}$$

We can compute the MLE for tree networks as we see in Section V-B. However, it becomes computationally hard for large networks; this creates the need for faster algorithms that provide good approximate performance in practice.

To measure the per link estimation accuracy, we use the mean-squared error (MSE): $\text{MSE} = E(|\alpha_e - \hat{\alpha}_e|^2)$. In order to measure the estimation performance on all links $e \in E$, we need a metric that summarizes all links. We use an entropy measure ENT that captures the residual uncertainty. Since we expect the scaled estimation errors to be asymptotically Gaussian (similar to the case in [2]), we define the quality of the estimation across all links as

$$\text{ENT} = \sum_{e\in E} \log\left(E[\hat{\alpha}_e - \alpha_e]^2\right) \tag{5}$$

which is a shifted version of the entropy of independent Gaussian random variables with the given variances [25]. If the entire error covariance matrix $\mathcal{R}$ is available, then we can compute the metric as $\text{ENT} = \log\det\mathcal{R}$, which captures also the correlations among the errors on different links. The metric ENT defined previously captures only the diagonal elements of $\mathcal{R}$, *i.e.*, the MSE for each link independently of the others.

In some cases, we approximate the error covariance matrix $\mathcal{R}$ using the Fisher information matrix $\mathcal{I}$. Under mild regularity conditions (see, for example, [26, Ch. 7]), the scaled asymptotic covariance matrix of the optimal estimator is lower bounded by the Cramer–Rao bound $\mathcal{I}^{-1}$. The Fisher information matrix $\mathcal{I}$ is a square matrix with element $\mathcal{I}_{p,q}$ defined as

$$\mathcal{I}_{p,q}(\alpha) = -E\left[\frac{\partial}{\partial\alpha_p}\log p(X_{(R)};\alpha)\frac{\partial}{\partial\alpha_q}\log p(X_{(R)};\alpha)\right] \tag{6}$$

where α_p, α_q are the success probabilities of two links. In particular, under the regularity conditions, the MLE is asymptotically efficient, *i.e.*, it asymptotically, in sample size, achieves this lower bound.

B. Subproblems

Given a certain network topology, a monitoring scheme for loss tomography can be designed by solving the following subproblems.

1) *Identifiability:* For each link $e \in E$, derive conditions that the scheme should satisfy so that the edge is identifiable. Whether the goal is to maximize the number of identifiable edges, or to measure the link success rate on a particular set of edges, the identifiability conditions will guide the routing and code design choices.
2) *Routing:* Select the sources and receivers of probe packets, the paths through which probes are routed, and the nodes where they will be linearly combined.[6] The design goals include minimizing the utilized bandwidth, and improving the estimation accuracy, while respecting the required identifiability conditions.
3) *Probe and Code Design:* Select the contents of the probes sent by the sources and the operations performed at intermediate nodes. The goal is to use the simplest operations

[6]Depending on the practical constraints, such flexibility may or may not be available. If one cannot choose the source/receiver nodes and/or routing, as it is the case in most of the tomography literature, then this step can be skipped. If one can choose some of these parameters, then this can lead to further optimization of identifiability and estimation accuracy.

and the smallest finite field, while ensuring that the identifiability conditions are met.

4) *Estimation Algorithm:* This is the algorithm that processes the collected probes at the receivers and estimates the link loss rates. The objective is low complexity with good estimation performance. There is clearly a tradeoff between the estimation error and the measurement bandwidth.

We note that these steps are *not* independent from each other. In fact, the design of routing, probe, and code design needs to be done with identifiability and estimation in mind.

C. Main Results

In this paper, we propose a monitoring scheme for loss tomography in networks that have multicast and network coding capabilities. In Sections V and VI, we present our design for the cases of trees and general topologies, respectively. We evaluate all our schemes through extensive simulation results. In the following, we preview the main results, in each subproblem.

1) *Identifiability:* 1) We provide simple necessary and sufficient conditions for *identifying* the loss rate of a single link. In (logical) tree topologies, all links are identifiable, using a very simple monitoring scheme.[7] In general topologies, where identifiability depends on the routing and code design as well, these conditions still apply. 2) We also prove a structural property, which we call *reversibility*: if a link is identifiable under a given monitoring scheme, it remains identifiable if we reverse the directionality of all paths and exchange the role of sources and receivers (which we call the *dual configuration*).
2) *Routing:* 1) For a given set of sources and receivers over an arbitrary topology, the problem of selecting a routing that meets the identifiability conditions while minimizing the employed bandwidth is NP-hard. We prove that, when network coding is used, this problem can be solved in polynomial time. 2) Moreover, we demonstrate, via simulation, that the choice of sources and receivers affects the estimation accuracy. 3) Finally, we present heuristic orientation algorithms for general graphs, designed to achieve identifiability, small number of receivers, and high estimation accuracy.
3) *Probe and Code Design:* 1) In trees, we show that binary vectors sent by the sources and deterministic code design with XOR operations at the intermediate nodes are sufficient. 2) In general graphs, we need to use operations over higher finite fields. We provide bounds on the required alphabet size, and we propose and evaluate deterministic code design.
4) *Loss Estimation:* 1) In a tree topology (under mild conditions on the selection of sources and receivers), we develop a low-complexity method for computing the MLE of the loss rates *for all links simultaneously*. Our algorithm builds on and extends MINC (the well-known ML estimator [2] for an MT) to multiple-source multiple-destination tree topologies (with multicast at branching points and network coding at joining points). We describe the algorithm, prove its correctness, and analyze its rate of convergence. 2) A key property that we formulate, prove, and extensively use in this work is *reversibility*, *i.e.*, the fact that the MLE's for a configuration and its dual (defined as the same topology, but with the role of sources and receivers reversed) have the same functional form. For example, the MLE for a *reverse multicast tree* (RMT) (with several sources and one receiver) has the same functional form as MINC for an MT (with the role of the source and the receivers reversed); we refer to the MLE for the RMT as RMINC. 3) For topologies other than trees, no efficient MLE algorithm is known for estimating the loss rates of all links simultaneously. Therefore, we propose a number of heuristic algorithms, including belief propagation (BP) and subtree decomposition algorithms, and we evaluate their performance through simulation. 4) We provide a simple algorithm for computing the MLE of *a single link* at a time in *any* topology. This is particularly useful in practice because 1) a few bottleneck links are typically congested, thus of interest; and 2) the method is applicable to *any* topology, even if it is not of the type (1) above.

The use of network coding at intermediate nodes, in addition to unicast and multicast, offers several benefits for loss tomography: it increases the number of identifiable links; it improves the tradeoff between number of probes and estimation accuracy; and it reduces the complexity of selecting probe paths for minimum cost monitoring of a general graph from NP-hard to linear. The approach gracefully generalizes from trees to general topologies (*e.g.*, having the same identifiability conditions, using the same estimation algorithm, and avoiding the use of overlapping trees or paths), where its advantages are amplified.

IV. Motivating Example

In this section, we present a motivating example to demonstrate the benefits of network coding in identifying the link loss rates; we derive the conditions of identifiability for a single link; and we discuss the identifiability of all links in the network.

Example 1: Consider the five-link topology depicted in Fig. 1. Nodes A and B send probes and nodes E and F receive them. Every link can drop a packet according to an i.i.d. Bernoulli distribution, with probability $\overline{\alpha}_e$, independently of other links. We are interested in estimating the success probabilities of all links, namely α_{AC}, α_{BC}, α_{CD}, α_{DE}, and α_{DF}.

The traditional multicast-based tomography approach would use two MTs rooted at nodes A and B and ending at E and F. This approach is depicted in Fig. 1(a) and (b). At each experiment, source A sends packet x_1 and source B sends packet x_2. The receivers E and F infer the link loss rates by keeping track of how many times they receive packets x_1 and x_2. Note that, due to the overlap of the two trees, for each experiment, links CD, DE, and DF are used twice, leading to inefficient bandwidth usage. Moreover, from this set of experiments, we cannot calculate α_{CD}, and thus, edge CD is not identifiable. Indeed, by observing the outcomes of experiments on each MT, we cannot distinguish whether packet x_1 is dropped on edge AC or CD; similarly, we cannot distinguish whether packet x_2 is dropped on edges BC or CD. (Note that if we restricted ourselves to

[7]This scheme is described in Section V-B1: it selects some leaf nodes as sources, and the remaining leaf nodes as receivers; the sources send simple binary vectors, and the intermediate nodes do simple XOR operations or multicast.

TABLE I

LIST OF TEN POSSIBLE OBSERVED OUTCOMES, THE STATE OF THE LINKS THAT LEAD TO A PARTICULAR OUTCOME, THE PROBABILITY OF OBSERVING THIS OUTCOME, AND THE NUMBER OF TIMES WE OBSERVE THIS OUTCOME IN A SEQUENCE OF N INDEPENDENT EXPERIMENTS. TEN LEFTMOST COLUMNS REFER TO THE FIVE-LINK TOPOLOGY IN FIG. 1(C). THEY SHOW THE POSSIBLE PAIRS OF PROBES COLLECTED (*I.E.,* THE OBSERVATIONS $x \in \Omega$) AT THE RECEIVERS E, F, THEIR PROBABILITIES P_α, AND THE NUMBER OF TIMES n_i EACH OBSERVATION OCCURRED. THESE OBSERVATIONS DEPEND ON THE COMBINATION OF LOSS (0) AND SUCCESS (1) ON THE FIVE LINKS, WHICH HAPPEN W.P. α. THE REMAINING RIGHTMOST COLUMNS SHOW HOW THE SAME PROBES CAN BE INTERPRETED AS OBSERVATIONS AT THE RECEIVER(S) OF THE REDUCED TOPOLOGIES, NAMELY THE MT AND THE RMT (AS WE DESCRIBE IN SECTION V-B3), AND THEIR CORRESPONDING PROBABILITIES

#	Is link working (1) or not (0)?					**Original (5-link) Tree** actual probes received at:		Prob.	#times	**Reduced Multicast Tree** observations			**Reduced Reverse Multicast Tree**	
	AC	BC	CD	DE	DF	E	F	P_α		E	F	P_α^m	EF	P_α^r
1	Multiple possible events					-	-	p_0	n_0	0	0	p_0	$[0,0]$	p_0
2	1	0	1	1	0	x_1	-	p_1	n_1				$[1,0]$	$p_1+p_4+p_7$
3	0	1	1	1	0	x_2	-	p_2	n_2	1	0	$p_1+p_2+p_3$	$[0,1]$	$p_2+p_5+p_8$
4	1	1	1	1	0	$x_1 \oplus x_2$	-	p_3	n_3				$[1,1]$	$p_3+p_6+p_9$
5	1	0	1	0	1	-	x_1	p_4	n_4				$[1,0]$	$p_1+p_4+p_7$
6	0	1	1	0	1	-	x_2	p_5	n_5	0	1	$p_4+p_5+p_6$	$[0,1]$	$p_2+p_5+p_8$
7	1	1	1	0	1	-	$x_1 \oplus x_2$	p_6	n_6				$[1,1]$	$p_3+p_6+p_9$
8	1	0	1	1	1	x_1	x_1	p_7	n_7				$[1,0]$	$p_1+p_4+p_7$
9	0	1	1	1	1	x_2	x_2	p_8	n_8	1	1	$p_7+p_8+p_9$	$[0,1]$	$p_2+p_5+p_8$
10	1	1	1	1	1	$x_1 \oplus x_2$	$x_1 \oplus x_2$	p_9	n_9				$[1,1]$	$p_3+p_6+p_9$

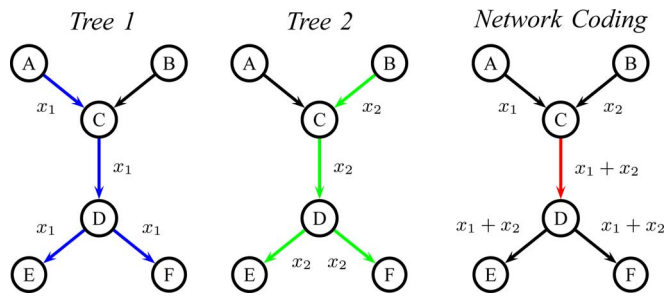


Fig. 1. Link loss monitoring for the basic five-link topology. Nodes A and B are sources, and nodes E and F are receivers. Using multicast-based tomography, the topology can be covered using two MTs 1 and 2. Alternatively, the topology can be covered using coded packets, if node C can add (XOR) incoming packets.

unicast only, four unicast probes from A, B to E, F would be needed to cover all five links. Not only would the problems of identifiability and overlap of probe paths still be present, but they would be further amplified.)

If network coding capabilities are available, they can help alleviate these problems. Assume that the intermediate node C can combine incoming packets before forwarding them to outgoing links. Node A sends to C a probe packet with payload that contains the binary string $x_1 = [1\ 0]$. Similarly, node B sends probe packet $x_2 = [0\ 1]$ to node C. If node C receives only x_1 or only x_2, then it just forwards the received packet to node D; if C receives both packets x_1 and x_2, then it creates a new packet, with payload their linear combination $x_3 = [1\ 1]$, and forwards it to node D; more generally, $x_3 = x_1 \oplus x_2$, where $\oplus$ is the bitwise XOR operation. Node D multicasts the incoming packet x_3 to both outgoing links DE and DF. The flow of packets in this experiment is shown in Fig. 1(c). In every experiment, probe packets (x_1, x_2) are sent from A, B, and may or may not reach E, F, depending on the state of the links. Observe that with the network coding approach, link CD becomes identifiable. Moreover, we have avoided the overlap of probes on link CD during each experiment.

Table I lists the ten possible observed outcomes, the state of the links that leads to a particular outcome, the probability $p_i,\ i = 0, \ldots, 9$ of observing this outcome, and the number of times $n_i,\ i = 0, \ldots, 9$ we observe this outcome in a sequence of n independent experiments. The probability of observing an outcome p_i can be computed from the success probabilities $\alpha = (\alpha_{AC}, \alpha_{BC}, \alpha_{CD}, \alpha_{DE}, \alpha_{DF})$ of the five links. For example, for outcomes 1–4

$$\begin{aligned} p_0 &= 1 - p_1 \cdots - p_9 = 1 - (1-\overline{\alpha}_{AC}\overline{\alpha}_{BC})\alpha_{CD}(1-\overline{\alpha}_{DE}\overline{\alpha}_{DF}) \\ p_1 &= \alpha_{AC}\overline{\alpha}_{BC}\alpha_{CD}\alpha_{DE}\overline{\alpha}_{DF} \\ p_2 &= \overline{\alpha}_{AC}\alpha_{BC}\alpha_{CD}\alpha_{DE}\overline{\alpha}_{DF} \\ p_3 &= \alpha_{AC}\alpha_{BC}\alpha_{CD}\alpha_{DE}\overline{\alpha}_{DF} \\ &\ldots \end{aligned} \tag{7}$$

and we can write similar expressions for the probabilities of the remaining observations. Thus, we can explicitly write down the probability distribution of the observations P_α.

In a sequence of $n = \sum_{i=0}^{i=9} n_i$ independent experiments, the frequency of each event i is $\hat{p}_i = \frac{n_i}{n}$. After sending n independent probes, the log-likelihood function of the observations given the set of parameters (α_e) is $\mathcal{L}(\alpha_{AC}, \alpha_{BC}, \alpha_{CD}, \alpha_{DE}, \alpha_{DF}) = \sum_{i=0}^{i=9} n_i \log p_i(\alpha)$. The MLE would compute the α's that maximize $\mathcal{L}(\alpha)$. □

In general, we may be interested in estimating one of the α variables, some of them, or all five of them. In the next section, we discuss a single link, namely link CD. Note that the remaining four links can depict the equivalent paths connecting CD to the sources and receivers. In Section IV-B, we discuss the identifiability of all links.

A. Identifiability of One Link

Let us focus on a single link CD with success probability α_{CD}. Consider Fig. 2, which generalizes the motivating example of the previous section. Note that links other than CD can be viewed as summarizing paths: *e.g.*, AC could correspond to a path from A to C, possibly consisting of the concatenation of several links.

For a given choice of sources and receivers and a coding scheme described in Section V-B1 (which is extremely simple: just pick any leaf or leaves as sources and the remaining leaves as receivers; sources send binary vectors; intermediate nodes

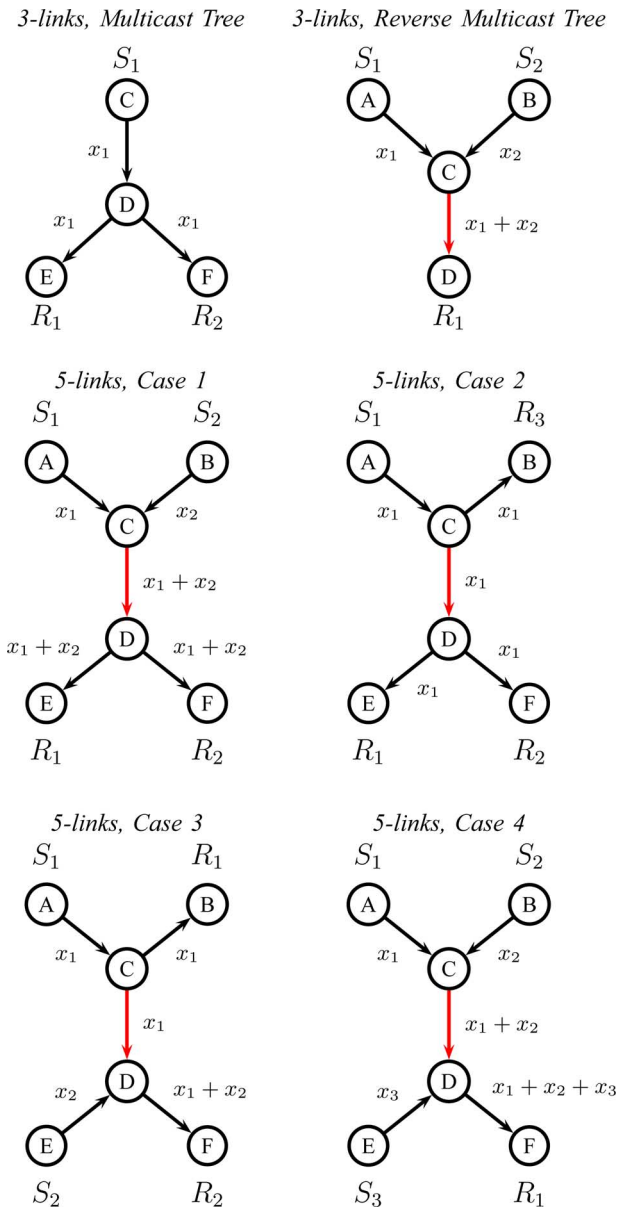


Fig. 2. Configurations (*i.e.*, combinations of) Conditions 1) and 2) that allow us to identify the success rate of a single link (CD). Recall that links, other than *CD*, can correspond to paths with the same loss probability. The top of the figure shows a three-link topology where *C* is a source (of an MT) or *D* is a receiver (of an RMT). The trivial case that *C* is a source and *D* is a receiver corresponds to a single-link topology and is omitted here. The bottom of the figure shows a five-link topology and four configurations (choices of sources and receivers), where neither C nor D are edge nodes and packets are sent and received at the edge nodes A, B, E, and F. Case 1 is our familiar motivating example; Case 2 is similar to a single MT rooted at A; Case 3 uses sources A and E and linear combinations whenever the two flows meet; Case 4 does the same thing for sources A, B, and E, and is equivalent to an inverse MT (with sink at F).

simply code using bitwise XOR or multicast), we want to translate the conditions for identifiability of link CD in Definition 2 to graph properties of the network. Our intuition is that a link CD is identifiable if C is a source, a coding point or a branching point, and D is a receiver, a coding point or a branching point. These are the structures depicted in Fig. 2, where we want to identify the link success rate associated with edge CD, and interpret the remaining edges as corresponding to paths. The top two cases of Fig. 2 depict the simple cases where node C is a source, or node D is a receiver; the four bottom cases depict the cases where C and D are coding or branching points.

TABLE II
IDENTIFIABLE LINKS IN THE FOUR CASES (DIFFERENT CHOICES OF SOURCES AND RECEIVERS, FOR THE SAME FIVE-LINK TOPOLOGY) DEPICTED AT THE BOTTOM OF FIG. 2

Case	Network Coding	Multicast Probes
1	all links	DE, DF
2	all links	all links
3	all links	AC, CB
4	all links	no links

To formalize this intuition, consider the following two conditions:

1) *Condition 1:* At least one of the following holds.
   a) $C \in S$.
   b) There exist two edge-disjoint paths (X_1, C) and (X_2, C) that do not employ edge CD, with distinct $X_1, X_2 \in S$.
   c) There exist two paths (X_1, C) and (C, X_2) that do not employ edge CD, with $X_1 \in S$, $X_2 \in R$.
2) *Condition 2*: At least one of the following holds.
   a) $D \in R$.
   b) There exist two edge-disjoint paths (D, X_1) and (D, X_2) that do not employ edge CD, with distinct $X_1, X_2 \in R$.
   c) There exist two paths (X_1, D) and (D, X_2) that do not employ edge CD, with $X_1 \in S$, $X_2 \in R$.

Theorem 4.1: For a given choice of sources and receivers and for the simple coding scheme described above, link CD is identifiable if and only if both Conditions 1 and 2 hold.

The proof is provided in Appendix A.1.

B. Identifiability of all Links

In fact, we can identify *all* links at the same time. It is sufficient to ensure that each link is identifiable, according to the conditions of Theorem 4.1. This is true in all directed trees, where each leaf node is either a source or a receiver, and each intermediate node satisfies the following mild conditions: 1) it has degree at least three (which is true in all logical topologies); 2) it has in-degree at least one (otherwise, the node should be a source); and 3) it has out degree at least one (otherwise, the node should be a receiver).

Example 2: Table II lists which links are identifiable in the four bottom cases of Fig. 2, if we use our approach versus if we use multicast tomography. All four configurations depict the same basic five-link topology, but they differ in the choice of sources and receivers. Our approach is able to identify all links for any sets of sources and receivers. This is not always the case for the multicast tomography. □

V. TREE TOPOLOGIES

In this section, we consider tree topologies, and we describe our design choices in the four subproblems: we have already discussed identifiability in the previous section. Next, we describe routing in Section V-A, probe and code design in Section V-B1 (operation of sources and intermediate nodes), and estimation algorithms in Sections V-B–V-D.

A. Routing, Selection of Sources and Receivers

Routing in trees is well defined: there exists a single path that connects a source to a receiver, through which probes flow. For a tree with L leaf nodes, some leaves act as sources S and the remaining leaves act as receivers $R = L \setminus S$. Intermediate nodes simply combine (XOR) the probes coming on all incoming links and forward (multicast) to all their outgoing links. This section looks at situations where we may have some freedom in the choice of the nodes that act as sources and receivers. If such flexibility is not available (as it is assumed in most tomography work), this step can be skipped. We study the effect of the selection of sources and receivers on estimation accuracy and we come up with empirical guidelines for source selection, obtained through a number of examples and simulation scenarios.

In Example 2, we saw that, with network coding, all links are identifiable, while if we use two MTs, they are not. In Appendix A.2, we revisit the basic five-link topology of Fig. 2 and we show that, even though with network coding links are identifiable for all four cases, the estimation accuracy differs depending on the number of sources and their relative positions in the tree. This idea also applies to larger topologies. For example, in [27], we consider a nine-link tree and we run simulations for different number and location of sources and we summarize the intuition obtained.

Link loss tomography is essentially a parameter estimation problem, and different choices of sources and receivers lead to different estimators. That is, for a fixed number of probes, each topology leads to a different estimation accuracy; put differently, to achieve the same MSE, we may need a different number of probes for each topology. In general, the optimal selection of the number and location of sources depends on the network topology, the values of link loss rates, and possibly the number of employed probes. This is currently an open problem.

B. Maximum Likelihood (ML) Estimation of all Link Loss Rates

In this section, we focus on tree topologies and we develop an efficient MLE to estimate all link loss rates from the observations at the receivers. In the special case where the topology is an MT, *i.e.*, probes are sent between one source and several receivers, an efficient ML estimator (MINC) has been designed in the pioneering paper [2]. We build on MINC, and we extend it to multiple-source multiple-receiver trees, where multicast is used at all branching points and network coding is used at all joining points [23]. We propose Algorithm 1 in Section V-B4, which provides an efficient way to compute the MLE of *all links at the same time*.

A key property that we formulate, prove, and extensively use in this section is *reversibility*, as discussed in Section III-C, and as we describe in detail in Section V-B2. In Section V-C, we also describe how to efficiently compute the MLE for *a single link* at a time (in both trees and general topologies). In Section V-D, we describe heuristic estimation algorithms, some of which apply to general topologies as well.

1) Model and Framework: We first describe the model of tree networks for which we derive the MLE.

Logical Tree: We consider a tree topology, like the one depicted in Fig. 3, $G = (V, E)$ consisting of the set V of nodes and the set E of directed links. M leaf nodes, shown on top of the tree, act as sources of probe packets. The remaining N leaves, shown at the bottom of the tree, act as receivers. As typically assumed in tomography problems (as described in Section III), this is a "logical" tree topology, *i.e.*, every intermediate node has degree at least three. An intermediate node is either a coding point (with multiple incoming links and one outgoing link) or a branching point (with one incoming link and multiple outgoing links). For each node j, we denote the set of its parents (nodes with a link outgoing to j) by $f(j)$ and the set of its children (nodes with a link coming from j) by $d(j)$. The source nodes $S = \{S_1, \ldots, S_M\}$ have no parent and the receiver nodes $R = \{R_1, \ldots, R_N\}$ have no children. G, S, R are considered known and fixed throughout the experiments.

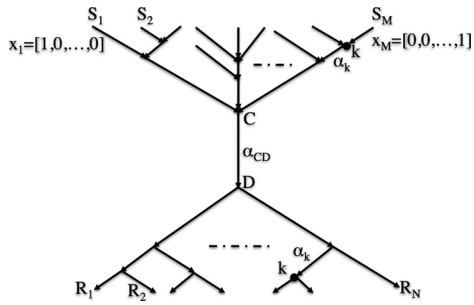


Fig. 3. Tree topology with multiple sources and multiple receivers. All sources are located at the top M leaves, and all receivers are located at the bottom N leaves. Multicast is used in all branching points and network coding is used in all joining/coding points. All coding points are located above all branching points. (This is a mild assumption that can be enforced if we are allowed to appropriately pick the sources and receivers.) For this tree topology, we have designed an algorithm that efficiently computes the MLE for *all links simultaneously*.

In this section, we focus on the tree topology shown in Fig. 3, which has the property that all coding points are located above all branching points. This is actually a mild assumption: starting from an undirected tree, if one is allowed to choose the sources among the leaf nodes, then one can always ensure this property.[8] Note that this tree model includes all cases in Fig. 2 (except for Case 3 in the five-link topology, which is treated separately in Section V-C).

Operation of Sources: Each source S_i sends a probe packet x_i, which is a vector of length M in the form of:

$$x_i = [\overbrace{0, \ldots, 0, \underbrace{1}_{i}, 0, \ldots, 0}^{M}], \quad i = 1, 2, \ldots, M.$$

Operation of Intermediate Nodes. Each coding point (bitwise) XORs all packets it receives from its parents, and forwards the result to its child.[9] This very simple design effectively keeps the presence of each source orthogonal from every other source. This ensures versatility, in the sense that no matter which probe packets get XOR-ed, they will not cancel each other out. For

[8]Once the sources are properly chosen, the rest of the leaves are receivers; the direction of the links is uniquely defined along the paths from the sources to the receivers; and intermediate nodes perform either coding or multicast, as uniquely dictated by the direction of their incoming and outgoing edges.

[9]We assume that the network is delay free and all packet arrivals at a coding point are synchronized. Link delays only affect where the probe packets would meet.

most practical purposes, this simple probe design is sufficient: a single IP packet can be up to 1500 B (including the headers) and thus, can accommodate roughly 12 000 probe sources (bits). In large networks, one can also spatially reuse probe packets by allocating the same probe packet to all sources whose packets do not meet. Finally, each branching point multicasts the packet it receives from its parent to all its children.

One can see that there will be a node after which $x_1 + x_2 + \cdots + x_M$ flows thought the network. We denote this node by C. Node C is the last coding point in the tree. Node C has P parents $f(C)_1, \ldots, f(C)_P$, and only one child, which we denote by node D. Node D multicasts the packet it receives from node C to all its Q children $d(D)_1, \ldots, d(D)_Q$.

We use the notation that $k < k'$, $k, k' \in V$ when k is a descendant of k', and that $k > k'$ when k is an ancestor of k'. Every node $k > C$ has multiple parents and only one child, while every node $k < D$ has one parent and multiple children. We are going to treat these two sets of nodes differently in the rest of Section V-B. We name any link of the tree that is above node C by its starting point, and we name any link that is below node D by its endpoint. In other words, link k denotes a link between nodes (k, j) if $k > C$ and $j > C$, while link k denotes a link between nodes (j, k) if $j < D$ and $k < D$.

Loss Model: As described in Section III, we model the loss rate of individual links by an i.i.d. Bernoulli process, independent across links. In particular, we use the following notation.

1) A packet that traverses a link k above node C is lost with probability $\overline{\alpha}_k = 1 - \alpha_k$ and arrives at node j with probability α_k.
2) A packet that traverses a link k below node D is lost with probability $\overline{\alpha}_k = 1 - \alpha_k$ and arrives at node k with probability α_k.
3) Finally, we denote the loss rate of link CD by $\overline{\alpha}_{CD}$.

In general, we use the notation $\overline{\alpha} = 1 - \alpha$ for any quantity $0 < \alpha < 1$.

Let X_k denote the packet observed at node k, and let $X = (X_k)$, $k \in V$ denote the set of all X_k's. X_k is a binary vector of length M. Its ith element $(X_k)_i$ represents the probe packet of source i: $(X_k)_i = 1$ indicates that the probe packet of source i reaches node k, and 0 that it does not. For the sources, $X_{S_i} = x_i$; thus, $(X_{S_i})_i = 1$ and $(X_{S_i})_{i'} = 0$, $\forall i' \neq i$. For any node $k \geq C$, if $(X_j)_i = 1$ for j a parent of k, $(X_k)_i = 1$ with probability α_j, and $(X_k)_i = 0$ with probability $\overline{\alpha}_j$, independently for all the parents of k. For any node $k \leq D$, if $X_k = [0, 0, \ldots, 0]$ (the all-zero vector), then $X_j = [0, 0, \ldots, 0]$, for the children j of k (and hence for all descendants of k). If $X_k \neq [0, 0, \ldots, 0]$, then for j a child of k, $X_j = X_k$ with probability α_j, and $X_j = [0, 0, \ldots, 0]$ with probability $\overline{\alpha}_j$, independently for all the children of k.

Data, Likelihood, and Inference: As described in Section III-A, in each experiment, one probe is dispatched from each source. The outcome of a single experiment is a record of whether or not each source probe was received at each receiver, which is the set of vectors X_k observed at receiver $k \in R$. It is denoted by $X_{(R)} = (X_k)_{k \in R}$ and is an element of the space $\Omega \subseteq \{[\ldots, 0, 1, \ldots]\}^N$ of all such outcomes. For a given set of link probabilities $\alpha = (\alpha_k)_{k \in V \setminus \{C,D\}} \cup \alpha_{CD}$, the distribution of the outcomes $X_{(R)}$ on Ω will be denoted by P_α. The probability mass function for a single outcome $x \in \Omega$ is $p(x; \alpha) = P_\alpha(X_{(R)} = x)$.

We perform n experiments. The probability of n independent observations $x^1, \ldots, x^n$ (each $x^t = (x_k^t)_{k \in R}$) is given by (1). Our task is to estimate α using ML, from the data $(n(x))_{x \in \Omega}$. We work with the log-likelihood function $\mathcal{L}(\alpha)$ given in (2). The MLE of the loss rates $\breve{\alpha}$ is the α that maximizes $\mathcal{L}(\alpha)$, as given by (3).

2) Likelihood Equation and its Solution: Candidates for the MLE are solutions $\hat{\alpha}$ of the *likelihood equation*:

$$\frac{\partial \mathcal{L}}{\partial \alpha_k}(\alpha) = 0, \quad k \in V. \tag{8}$$

We need to define some additional variables to compute the MLEs. For each node $k \geq D$, let $\Omega^r(k)$ be the set of outcomes $x \in \Omega$ such that $(x_a)_j \neq 0$ for at least one source $j \in S$ that is an ancestor of k and for any arbitrary set of receivers $\{a\} \subset R$. Let $\gamma_k^r = \Gamma_k^r(\alpha) = P_\alpha[\Omega^r(k)]$; an estimate of γ_k^r can be computed from:

$$\hat{\gamma}_k^r = \sum_{x \in \Omega^r(k)} \hat{p}(x), \quad \text{where} \quad \hat{p}(x) = \frac{n(x)}{n} \tag{9}$$

is the observed proportion of experiments with outcome x. γ_k^r shows the probability of the set of outcomes $\Omega^r(k)$ in which link k has definitely worked. Note that link k may have worked for some other outcomes as well, but they are not included in $\Omega^r(k)$. Also note that γ_k^r can be directly estimated from the observations at the receivers.

For each node $k \leq C$, we define $\Omega^m(k)$ to be the set of outcomes $x \in \Omega$ such that $x_j \neq [0, 0, \ldots, 0]$ for at least one receiver $j \in R$ which is a descendant of k. Let $\gamma_k^m = \Gamma_k^m(\alpha) = P_\alpha[\Omega^m(k)]$; an estimate of γ_k^m is:

$$\hat{\gamma}_k^m = \sum_{x \in \Omega^m(k)} \hat{p}(x) \tag{10}$$

where γ_k^m is the probability of the outcomes $\Omega^m(k)$ in which link k has definitely worked; and it can be directly estimated from the observations at the receivers. Our goal is to compute $\hat{\alpha}$ from $\hat{\gamma} = (\hat{\gamma}_k^r \cup \hat{\gamma}_k^m)_{k \in V}$.

Special Case (i): MT (MINC): If $M = 1$, the general model turns into an MT with a single source, which is the case considered in [2]. We represent the source node by $0 \in V$. Each node j other than the source node has one parent $f(j)$ and a set $d(j)$ of children. We denote the link loss rates by $\overline{\alpha}_k$, where k is the endpoint. We simply assume that $\alpha_0 = 1$.

The outcome of each experiment is $X_{(R)} = (X_k)_{k \in R}$, where each X_k is a single binary value (instead of a binary vector of length M in the general case), corresponding to whether the source probe is observed at each receiver $k \in R$ or not. The state space of the observations $X_{(R)}$ is $\Omega = \{0, 1\}^N$. We say that a link k is at level $l^m(k)$ if there is a chain of $l^m(k)$ ancestors $k < f(k) < f^2(k) \cdots < f^{l^m(k)}(k) = 0$ leading back to the source.

Only $\Omega^m(k)$ is used for each node k in the MT; it is the set of outcomes $x \in \Omega$ where $x_j = 1$ for at least one receiver $j \in R$ that is a descendant of k. The definition of γ_k^m is like before.

The MLE for the MT has been computed in [2]: Let $A_k^m = \prod_{i=0}^{l^m(k)} \alpha_{f^i(k)}$ show the probability that the path from the source to node k works, which we denote by $P(Y_{0\to k}=1)$. Its estimate $\hat{A}_k^m$ can be computed as follows. For the source node, $\hat{A}_0^m = 1$, for the leaf nodes $k \in R$, $\hat{A}_k^m = \hat{\gamma}_k^m$, and for all other nodes $k \in V\backslash\{0,R\}$, $\hat{A}_k^m$ is the unique solution in $(0,1]$ of:

$$1 - \frac{\hat{\gamma}_k^m}{\hat{A}_k^m} = \prod_{j\in d(k)} (1 - \frac{\hat{\gamma}_j^m}{\hat{A}_k^m}) \tag{11}$$

$\hat{\alpha}_k$ can then be computed from $\hat{\gamma}_k^m$, *i.e.*, $\hat{\alpha} = \Gamma^{m-1}(\hat{\gamma^m})$, as follows:

$$\hat{\alpha}_k = \frac{\hat{A}_k^m}{\hat{A}_{f(k)}^m}, \quad k \in V\backslash\{0\} \quad (\hat{\alpha}_0 = 1). \tag{12}$$

We refer to (12) as MINC in the rest of this paper.

Note. Equation (11) is obtained from the following relations, after some computations in [2], which we repeat here for completeness. Let $\beta_k^m = P[\Omega^m(k)|X_{f(k)}=1]$ denote the conditional probability of $\Omega^m(k)$ given that $f(k)$ has observed something. Failure can be due to either $\overline{\alpha}_k$ (failure of link k), or all paths toward the destinations failing. Therefore, the β_k^m obey the following recursion:

$$\overline{\beta}_k^m = \overline{\alpha}_k + \alpha_k \prod_{j\in d(k)} \overline{\beta}_j^m, \quad k \in V\backslash R \tag{13}$$

$$\beta_k^m = \alpha_k, \quad k \in R. \tag{14}$$

Equation (11) then follows from the following relation between α and γ^m:

$$\gamma_k^m = \beta_k^m \prod_{i=1}^{l^m(k)} \alpha_{f^i(k)}. \tag{15}$$

Special Case (ii): RMT (RMINC): If $N = 1$, the general model turns into an RMT with a single receiver, which we denote by $0 \in V$. Each node j other than 0 has one child $d(j)$, and a set $f(j)$ of parents. We denote link loss rates by $\overline{\alpha}_k$, where k is the starting point. We assume that $\alpha_0 = 1$.

The outcome of each experiment X_R is a binary vector of length M. Each of its elements $(X_R)_i$ represents whether the probe packet of source i is observed at the receiver or not. The state space of the observations X_R is $\Omega = \{0,1\}^M$. We say that a link k is at level $l^r(k)$ if there is a chain of $l^r(k)$ descendants $k > d(k) > d^2(k) \cdots > d^{l^r(k)}(k) = 0$ leading down to the receiver.

Only $\Omega^r(k)$ is used for each node k in the RMT; it is the set of outcomes $x \in \Omega$ where $x_j = 1$ for at least one source $j \in S$ that is an ancestor of k. The definition of γ_k^r is like before.

The MLE for the RMT is similar to the MT. Let $A_k^r = \prod_{i=0}^{l^r(k)} \alpha_{d^i(k)}$ show the probability that the path from node k to the receiver node works, which we denote by $P(Y_{k\to 0} = 1)$. Its estimate $\hat{A}_k^r$ can be computed as follows. For the receiver node, $\hat{A}_0^r = 1$, for the source nodes $k \in S$, $\hat{A}_k^r = \hat{\gamma}_k^r$, and for all other nodes $k \in V\backslash\{S,0\}$, $\hat{A}_k^r$ is the unique solution in $(0,1]$ of:

$$1 - \frac{\hat{\gamma}_k^r}{\hat{A}_k^r} = \prod_{j\in f(k)} (1 - \frac{\hat{\gamma}_j^r}{\hat{A}_k^r}). \tag{16}$$

We can then compute $\hat{\alpha}_k$ from $\hat{\gamma}_k^r$, *i.e.*, $\hat{\alpha} = \Gamma^{r-1}(\hat{\gamma^r})$, as follows:

$$\hat{\alpha}_k = \frac{\hat{A}_k^r}{\hat{A}_{d(k)}^r}, \quad k \in V\backslash\{0\} \quad (\hat{\alpha}_0 = 1). \tag{17}$$

We refer to (17) as RMINC in the rest of the paper.

Note. Equation (16) results from the following relations. Let $\beta_k^r = P[\Omega^r(k)|Y_{d(k)\to 0} = 1]$ denote the conditional probability of $\Omega^r(k)$ given that the path from $d(k)$ to the receiver works. We have that:

$$\overline{\beta}_k^r = \overline{\alpha}_k + \alpha_k \prod_{j\in f(k)} \overline{\beta}_j^r, \quad k \in V\backslash S \tag{18}$$

$$\beta_k^r = \alpha_k, \quad k \in S \tag{19}$$

$$\gamma_k^r = \beta_k^r \prod_{i=1}^{l^r(k)} \alpha_{d^i(k)}. \tag{20}$$

Comparison of MINC and RMINC: The reader will notice that the MLE for the MT and the RMT has the same functional form. This is a special case of the more general "reversibility" property, first observed in [22]. Indeed, there is a 1–1 correspondence between the observable outcomes in the two cases; furthermore, the corresponding outcomes have the same probability, as a function of α_k's, thus leading to the same MLE. In the following, we describe the reversibility property in more detail.

Reversibility—A Structural Property: Consider a tree topology $G = (V,E)$ with L leaf nodes, some of which act as sources S and the remaining ones, $R = L \setminus S$, act as receivers of probes. Routing from S to R is given (*e.g.*, determined in the routing subproblem) and defines a direction on every link $e \in E$, along which probes flow.

Definition 3: We call the triplet (G,S,R) a *configuration*.

We define as dual the configuration that results from reversing the orientation of all links in the network, and from having the sources S become receivers, while the receivers R act as sources. More formally, we have the following.

Definition 4: Consider the original configuration (G,S,R). Consider the graph $G^d = (V,E^d)$ that has the same nodes but reversed edges, *i.e.*, $e = (i,j) \in E$ iff $e^d = (j,i) \in E^d$, and success rate $\alpha_e^d = \alpha_e$, associated with every edge $e^d \in E^d$. Select sources $S^d = R$ and receivers $R^d = S$. We call the (G^d, S^d, R^d) the *dual configuration* of (G,S,R).

For example, an MT is the dual configuration of an RMT (Cases 2 and 4 in Fig. 2). In Appendix B, we show that the dual configurations of Fig. 25(a) and (b) result in the same MSE bound. In fact, a closer look reveals that not only the values but also the functional forms of these two ML estimators coincide. The following theorem generalizes this notion to general trees.

Theorem 5.1: Consider a configuration (G,S,R) with observations at the receivers Ω, and probability distribution $P_\alpha = \{p(x;\alpha),\ x \in \Omega\}$. Consider its dual configuration (G^d,R,S), with observations Ω^d and probability distribution P_α^d. Then, there is a bijection between outcomes and their probabilities in the original $(x \in \Omega,\ p(x;\alpha))$ and in the dual configuration $(x^d \in \Omega^d,\ p(x^d;\alpha))$.

Proof: Let $G = (V,E)$ be the original tree graph, and G^d its dual. In every experiment, there exist $2^{|E|}$ possible error

events, depending on which subset of the links fails. Observing the outcomes at the receivers corresponds to observing unions of events that occur with the corresponding probability (*e.g.*, as in the example of Table I). We show that for each observable outcome, which occurs with probability p in G, there exists exactly one observable outcome that occurs with the same probability in G^d and vice versa. This establishes a bijection.

With every edge e of G, we can associate a set of sources $S(e) \subset V$ that flow through this edge, and a set of receivers $R(e) \subset V$ that observe the flow through e. Our main observation is that the pair $\{S(e),\ R(e)\}$ uniquely identifies e, *i.e.*, no other edge has the same pair. In the dual configuration G^d, edge e is uniquely identified by the pair $\{R(e),\ S(e)\}$. If in G, edge e fails while all other edges do not, the receivers $R(e)$ will not receive the contribution in the probe packets of the sources $S(e)$. If in G^d, edge e fails while all other edges do not, the receivers $S(e)$ will not receive the contribution in the probe packets of the sources $R(e)$. Thus, there is a one-to-one mapping between these events. Using this equivalence, an observable outcome consisting of a union of events can be mapped to an observable outcome in the reverse tree. ■

Corollary 5.2: The MLEs for a configuration and its dual have the same functional form.

Proof: The bijection established previously implies that a configuration and its dual have the same set of observable outcomes, with the same probabilities. Therefore, they have the same likelihood function and, thus, the same MLE. ■

We note that this corollary establishes reversibility only for the ML estimation. The performance of suboptimal algorithms may differ when applied to a configuration and its dual.

A note on directional networks: It is also important to note that the notion of dual configurations does *not* assume that the loss rates in both directions of a link are the same. Reversibility means that the two ML estimators for a configuration and its dual are described by the same function. However, the loss parameters we try to estimate (using the same estimator function) in the two directions may have different values.

3) ML Estimation of Loss Rates: We now present how to "reduce" the original tree to an MT and to an RMT, and how to estimate α_{CD}. These intermediate results are then used in the MLE algorithm in Section V-B4.

Reduction to an MT (m): If we take the upper part of the original tree in Fig. 3 and consider it as an aggregate link, we obtain the reduced MT in Fig. 4(a). The aggregate link agg^m summarizes the operation of all links above node C and link CD. Node D receives a packet if at least one path from the sources to node C works and link CD works. In other words, the success probability of the aggregate link α^m_{agg} depends on the paths from the sources to node C, and also link CD.

More formally, we map the outcomes $x \in \Omega$ of the original tree to the outcomes x^m of the MT, as follows. Each x is a set of N binary vectors, each of length M, while each x^m is a single binary vector of length N. Any outcome x^m is obtained by taking a set of outcomes $\{x\}$, in all of which the same receivers have observed all-zero vectors[10] and the same receivers

[10]Note that if a receiver does not receive any packet, then this is treated as an all-zero vector.

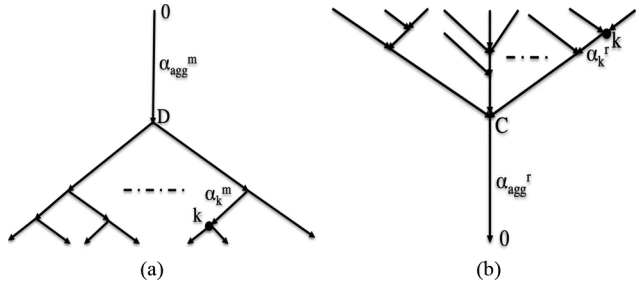


Fig. 4. Reducing the tree topology in Fig. 3 to an MT and to an RMT. (a) Reduced MT. (b) Reduced RMT.

have observed nonzero vectors, and by replacing each nonzero vector (that may contain any of the source probes $x_1, \ldots, x_M$) by value 1, and each all-zero vector by value 0, *i.e.*,

$$\sum_{x_{R_t} \neq [0,0,\ldots,0], x_{R_{t'}} = [0,0,\ldots,0]} n(x) = n^m(x^m)$$
$$x^m_{R_t} = 1, x^m_{R_{t'}} = 0,\ t, t' \in \{1, \ldots, N\}, t \neq t'. \quad (21)$$

If the original tree has link success rates α and an associated probability distribution of outcomes P_α, then the MT is defined with parameters α^m and associated probability distribution P^m_α, such that:

$$\alpha^m_k = \alpha_k,\ k < D, \quad \alpha^m_{\text{agg}} = \alpha_{CD}(1 - \prod_{i=1}^{P} \overline{\beta}^r_{f(C)_i}). \quad (22)$$

P^m_α can be directly calculated from P_α, since each event in P^m_α is the union of a disjoint subset of events in P_α and has probability equal to the sum of probabilities of those events in P_α (such as the five-link example in Table I).

Reduction to an RMT (r): Similarly, if we consider the lower part of the original tree in Fig. 3 as an aggregate link, we obtain the reduced RMT in Fig. 4(b), with parameters α^r and associated probability distribution P^r_α, such that:

$$\alpha^r_k = \alpha_k,\ k > C, \quad \alpha^r_{\text{agg}} = \alpha_{CD}(1 - \prod_{j=1}^{Q} \overline{\beta}^m_{d(D)_j}). \quad (23)$$

The Relation Between the Two Reduced Trees:

Lemma 5.3: We have that $\hat{\gamma}^r_C = \hat{\gamma}^m_D = 1 - \hat{p}([0, 0, \ldots, 0])$.

The proof directly results from the definitions of γ^m_D in the reduced MT and γ^r_C in the reduced RMT.

Estimating α_{CD}: The MLE of α_{CD} can be obtained from:

$$\hat{\alpha}_{CD} = \frac{\hat{A}^r_C \cdot \hat{A}^m_D}{\hat{\gamma}^r_C} = \frac{\hat{A}^r_C \cdot \hat{A}^m_D}{\hat{\gamma}^m_D}. \quad (24)$$

The proof can be found in Appendix A.2.

4) Analysis of the MLE: In this section, we propose the MLE algorithm, we discuss its complexity, and we illustrate our results through the example tree topology in Fig. 1(c).

MLE Algorithm: Algorithm 1 computes the MLE of all link loss rates in the tree topology of Fig. 3; it proceeds in the following steps: 1) it computes $\hat{\alpha}_k$ for any link k below node D from the reduced MT using (12); 2) it computes $\hat{\alpha}_k$ for any link

k above node C from the reduced RMT using (17); and 3) it computes $\hat{\alpha}_{CD}$ from (24). These are indeed the MLEs of the link loss rates, $\hat{\alpha}$, for the tree of Fig. 3.

Algorithm 1 Computing the MLE of all Link Loss Rates in the Original Tree Topology of **Fig. 3**

1: **for all** links k, where $k < D$ **do**

2: Reduce the original tree to an MT. Use MINC **[2] (12)** to compute the MLEs $\hat{\alpha}_k^m$ and $\hat{\alpha}_{\text{agg}}^m$.

3: Let $\hat{\alpha}_k = \hat{\alpha}_k^m$.

4: **end for**

5: **for all** links k, where $k > C$ **do**

6: Reduce the original tree to an RMT. Use RMINC **(17)** to compute the MLEs $\hat{\alpha}_k^r$ and $\hat{\alpha}_{\text{agg}}^r$.

7: Let $\hat{\alpha}_k = \hat{\alpha}_k^r$.

8: **end for**

9: Use **(24)** to compute the MLE $\hat{\alpha}_{CD}$.

Theorem 5.4: The estimates computed by Algorithm 1 are the MLEs of the link loss rates in the original tree topology in Fig. 3.

The proof of Theorem 5.4 relies on the following two lemmas, whose proofs are provided in Appendix A.3. (Theorem 5.4 is then proved in Appendix A.4.)

Lemma 5.5: The solutions of the likelihood equations of the original tree and the reduced MT are related via: 1) $\hat{\alpha}_k = \hat{\alpha}_k^m$, $k < D$; and 2) $\hat{\alpha}_{CD} = \hat{\alpha}_{\text{agg}}^m / (1 - \prod_{i=1}^{P} \overline{\beta}_{f(C)_i}^r)$.

Lemma 5.6: The solutions of the likelihood equations of the original tree and the reduced RMT are related via: 1) $\hat{\alpha}_k = \hat{\alpha}_k^r$, $k > C$; and 2) $\hat{\alpha}_{CD} = \hat{\alpha}_{\text{agg}}^r / (1 - \prod_{j=1}^{Q} \overline{\beta}_{d(D)_j}^m)$. We note that the likelihood functions of the original tree and the reduced multicast (or reverse multicast) tree are different. What the aforementioned lemmas establish is that these likelihood functions are maximized for the same values of their common variables.

Complexity: Algorithm 1 is very efficient. In the first two steps, it calls MINC and RMINC. MINC (and thus RMINC) is known to be efficient by exploiting the hierarchy of the tree topology to factorize the probability distribution and recursively compute the estimates. The computation at each node is at worst proportional to the depth of the tree [2]. The last step $\hat{\alpha}_{CD}$ uses the estimates $\hat{A}_k, \hat{\gamma}_k$ already computed in the first two steps.

Rate of Convergence of the MLE: We can provide the rate of convergence of $\hat{\alpha}$ to the true value α. The Fisher information matrix at α based on $X_{(R)}$ is obtained from $\mathcal{I}_{jk}(\alpha) = -E\frac{\partial^2 \mathcal{L}}{\partial \alpha_j \partial \alpha_k}(\alpha)$[2]. We have the following.

Theorem 5.7: $\mathcal{I}(\alpha)$ is nonsingular, and as $n \to \infty$, $\sqrt{n}(\hat{\alpha} - \alpha)$ converges in distribution to $\mathcal{N}(0, \mathcal{I}^{-1}(\alpha))$.

The proof follows from the asymptotic properties of the MLEs [2], [28]. Therefore, asymptotically for large n, with probability $1-\delta$ (for $1-\delta$ confidence interval), $\hat{\alpha}_k$ lies between the points:[11]

$$\alpha_k \pm z_{\delta/2}\sqrt{\frac{\mathcal{I}_{kk}^{-1}(\alpha)}{n}}. \tag{25}$$

Example 3: We now illustrate our results by revisiting the example five-link tree topology in Fig. 1(c). Note that here, following the notation described in Section V-B1, we use the notation α_A, α_B, α_E, and α_F, for the four edge links in Fig. 1(c), instead of α_{AC}, α_{BC}, α_{DE}, and α_{DF}, respectively, which were used in Example 1.

MLE: The two source nodes A and B send probe packets $x_1 = [1, 0]$ and $x_2 = [0, 1]$, respectively. The space Ω consists of ten possible outcomes shown in Table I. Table I also shows the corresponding outcomes for the reduced MT and the reduced RMT. From (9) and (10), we have that:

$$\begin{aligned}
\hat{\gamma}_A^r &= \hat{p}_1 + \hat{p}_3 + \hat{p}_4 + \hat{p}_6 + \hat{p}_7 + \hat{p}_9 \\
\hat{\gamma}_B^r &= \hat{p}_2 + \hat{p}_3 + \hat{p}_5 + \hat{p}_6 + \hat{p}_8 + \hat{p}_9 \\
\hat{\gamma}_C^r = \hat{\gamma}_D^m &= \hat{p}_1 + \hat{p}_2 + \hat{p}_3 + \hat{p}_4 + \hat{p}_5 + \hat{p}_6 + \hat{p}_7 + \hat{p}_8 + \hat{p}_9 = 1 - \hat{p}_0 \\
\hat{\gamma}_E^m &= \hat{p}_1 + \hat{p}_2 + \hat{p}_3 + \hat{p}_7 + \hat{p}_8 + \hat{p}_9 \\
\hat{\gamma}_F^m &= \hat{p}_4 + \hat{p}_5 + \hat{p}_6 + \hat{p}_7 + \hat{p}_8 + \hat{p}_9.
\end{aligned}$$

We then solve (11) for $\hat{A}_k^m$ and (16) for $\hat{A}_k^r$, and then we find $\hat{\alpha}_A$ and $\hat{\alpha}_B$ from (17), $\hat{\alpha}_E$ and $\hat{\alpha}_F$ from (12), and $\hat{\alpha}_{CD}$ from (24), as follows:

$$\hat{\alpha}_A = \frac{\hat{\gamma}_A^r + \hat{\gamma}_B^r - \hat{\gamma}_C^r}{\hat{\gamma}_B^r} \quad , \quad \hat{\alpha}_B = \frac{\hat{\gamma}_A^r + \hat{\gamma}_B^r - \hat{\gamma}_C^r}{\hat{\gamma}_A^r} \tag{26}$$

$$\hat{\alpha}_E = \frac{\hat{\gamma}_E^m + \hat{\gamma}_F^m - \hat{\gamma}_D^m}{\hat{\gamma}_F^m} \quad , \quad \hat{\alpha}_F = \frac{\hat{\gamma}_E^m + \hat{\gamma}_F^m - \hat{\gamma}_D^m}{\hat{\gamma}_E^m} \tag{27}$$

$$\hat{\alpha}_{CD} = \frac{\hat{\gamma}_A^r \hat{\gamma}_B^r \hat{\gamma}_E^m \hat{\gamma}_F^m}{\hat{\gamma}_D^m (\hat{\gamma}_A^r + \hat{\gamma}_B^r - \hat{\gamma}_C^r)(\hat{\gamma}_E^m + \hat{\gamma}_F^m - \hat{\gamma}_D^m)}. \tag{28}$$

Confidence Intervals. Fig. 5 shows $\mathcal{I}^{-1}(\alpha)$ for the confidence intervals in (25). We note that the confidence intervals for parameters $\hat{\alpha}$ can be obtained by inserting (26)–(28) into Fig. 5. □

C. MLE of a Single Link

Section V-B provides a computationally efficient way to estimate all link loss rates at the same time, under the mild assumption that the tree is of the form depicted in Fig. 3. If one is allowed to pick the sources and the receivers in the tree, then one can ensure that this mild assumption holds.

However, there are practical scenarios where one might not want to or might not be able to use this scheme. First, if we are not allowed to choose the sources, *e.g.*, due to practical constraints, it is possible that the monitoring scheme does not have the desired property of Fig. 3, *i.e.*, all coding points may not be above all branching points. An example is Case 3 in the five-link topology of Fig. 2: all links are still identifiable, but the assumption does not hold and the MLE provided in the previous section

[11] $z_{\delta/2}$ denotes the number that cuts off an area $\delta/2$ in the right tail of the standard normal distribution.

$$\mathcal{I}^{-1}(\alpha)=\begin{pmatrix} \frac{\alpha_A\overline{\alpha}_A}{\alpha_B\alpha_{CD}(\alpha_E+\alpha_F-\alpha_E\alpha_F)} & \frac{\overline{\alpha}_A\overline{\alpha}_B}{\alpha_{CD}(\alpha_E+\alpha_F-\alpha_E\alpha_F)} & \frac{-\overline{\alpha}_A\overline{\alpha}_B}{\alpha_B(\alpha_E+\alpha_F-\alpha_E\alpha_F)} & 0 & 0 \\ \frac{\alpha_A\overline{\alpha}_B}{\alpha_{CD}(\alpha_E+\alpha_F-\alpha_E\alpha_F)} & \frac{\alpha_B\overline{\alpha}_B}{\alpha_A\alpha_{CD}(\alpha_E+\alpha_F-\alpha_E\alpha_F)} & \frac{-\overline{\alpha}_A\overline{\alpha}_B}{\alpha_A(\alpha_E+\alpha_F-\alpha_E\alpha_F)} & 0 & 0 \\ \frac{-\overline{\alpha}_A\alpha_B}{\alpha_B(\alpha_E+\alpha_F-\alpha_E\alpha_F)} & \frac{-\overline{\alpha}_A\alpha_B}{\alpha_A(\alpha_E+\alpha_F-\alpha_E\alpha_F)} & \mathcal{I}_{33}^{-1}(\alpha) & \frac{-\overline{\alpha}_E\overline{\alpha}_F}{\alpha_F(\alpha_A+\alpha_B-\alpha_A\alpha_B)} & \frac{-\overline{\alpha}_E\overline{\alpha}_F}{\alpha_E(\alpha_A+\alpha_B-\alpha_A\alpha_B)} \\ 0 & 0 & \frac{-\overline{\alpha}_E\alpha_F}{\alpha_F(\alpha_A+\alpha_B-\alpha_A\alpha_B)} & \frac{\alpha_E\overline{\alpha}_E}{\alpha_{CD}\alpha_F(\alpha_A+\alpha_B-\alpha_A\alpha_B)} & \frac{\overline{\alpha}_E\overline{\alpha}_F}{\alpha_{CD}\alpha_E(\alpha_A+\alpha_B-\alpha_A\alpha_B)} \\ 0 & 0 & \frac{-\overline{\alpha}_E\alpha_F}{\alpha_E(\alpha_A+\alpha_B-\alpha_A\alpha_B)} & \frac{\overline{\alpha}_E\overline{\alpha}_F}{\alpha_{CD}(\alpha_A+\alpha_B-\alpha_A\alpha_B)} & \frac{\alpha_F\overline{\alpha}_F}{\alpha_{CD}\alpha_E(\alpha_A+\alpha_B-\alpha_A\alpha_B)} \end{pmatrix}$$

$$\mathcal{I}_{33}^{-1}(\alpha)=\frac{1}{\alpha_A\alpha_B\alpha_E\alpha_F(-\alpha_A\overline{\alpha}_B-\alpha_B)(-\alpha_E\overline{\alpha}_F-\alpha_F)}(-\alpha_{CD}(-\alpha_B\overline{\alpha}_B\alpha_E\alpha_F-\alpha_A^2\overline{\alpha}_B\alpha_E(-1+\alpha_B(2+\alpha_{CD}(-\alpha_E\overline{\alpha}_F-\alpha_F)))\alpha_F$$
$$+\alpha_A(-\alpha_E\alpha_F+\alpha_B^2\alpha_E\alpha_F(-3+\alpha_{CD}(\alpha_E+\alpha_F-\alpha_E\alpha_F))+\alpha_B(-\alpha_F\overline{\alpha}_F+\alpha_E(-1+7\alpha_F-3\alpha_F^2)+\alpha_E^2(1-3\alpha_F+2\alpha_F^2)))))$$

Fig. 5. Inverse of the Fisher information matrix governing the confidence intervals for models in (25). Here, the order of the coordinates is $\alpha_A, \alpha_B, \alpha_{CD}, \alpha_E, \alpha_F$.

does not apply. Second, we are often not even interested in estimating the loss rates for all links; it is common that only one or a few bottleneck/congested links are of interest. In general topologies, focusing on a few, as opposed to all, links has the side benefit that we may not need to deal with cycles, if they do not appear in the paths that go through the links of interest.

In all these cases, we propose that one estimates the loss rate of one link at a time. Recall the discussion in Section IV-A. The conditions for identifiability of a link (say link CD in Theorem 4.1) still apply, while the other four links AC, BC, DE, and DF in the five-link topology can be interpreted as paths from/to the sources/receivers; *i.e.*, we do not care about the individual link loss rates on these paths. Depending on the constraints on the selection of sources, any of the four cases in the five-link topology of Fig. 2 may be possible. We note that Table I and Algorithm 1 correspond to Case 1 in the five-link topology. Tables for the other three cases are provided in Appendix B.2.

In fact, similar MLE algorithms can be provided for all other three cases. For example, MINC and RMINC can be used for Cases 2 and 4 directly. Only Case 3 needs to be estimated similarly to Case 1 using reductions and Table VI. For Case 3, the reduced MT will consist of AC, CB, CD, and DF. We use MINC on this tree to infer the loss rates $\overline{\alpha}_{AC}$ and $\overline{\alpha}_{CB}$. The reduced RMT will consist of AC, CD, ED, and DF. We use RMINC on this tree to infer the loss rates $\overline{\alpha}_{ED}$ and $\overline{\alpha}_{DF}$. We can then replace these results in the likelihood function and find $\overline{\alpha}_{CD}$ by maximizing it. In general, an algorithm similar to Algorithm 1 can be developed to compute the MLE for the single link of interest: we first compute MINC on the reduced MT, then RMINC on the reduced RMT, and then we estimate link CD using a similar procedure as in Appendix A.2.

1) Remarks: Note that even when we focus on estimating a single link, the brute force approach appears to be computationally demanding even though it involves only five variables. Therefore, the efficient computation of the MLE for a single link is an important contribution on its own.

D. Heuristic Approaches for Loss Estimation

Beyond tree topologies, there is no known computationally efficient algorithm to compute the MLE of *all* link loss rates. In this section, we propose three heuristic estimation algorithms and evaluate their performance through simulation. The first two (subtree decomposition and MINC-like heuristic, in Sections V-D-I and V-D-II, respectively) are specific to trees, while the third one (BP, in Section V-D-III) applies also to general graphs.

1) Subtree Decomposition: Algorithm 2 partitions the tree into multicast subtrees separated by coding points. Each coding point virtually acts as a receiver for incoming flows and as a source for outgoing flows. As a result, each subtree will either have a coding point as its source, or will have at least one coding point as a receiver. In each subtree, we can then use the ML estimator (MINC) proposed in [2].

Algorithm 2 Subtree Decomposition Algorithm

Consider a tree G, with sources S and receivers R. Each source sends one probe packet. Each receiver receives at most one probe packet.

1) Determine the coding points. These partition G into $|\mathcal{T}| \leq 2M-1$ subtrees.
2) For each of the $|\mathcal{T}|$ subtrees:
   — If the MT is rooted at a coding point:
   * if any of the descendant receivers receives a probe, use this experiment as a measurement on the subtree.
   * otherwise, w.p. p assume no node in R received a probe packet, and w.p. $(1-p)$ ignore the experiment.

   — If the MT is rooted at a source S_i:
   Consider each coding point C that acts as a receiver:
   *if no descendant receivers $C(R)$ observed a probe, assume, w.p. p, that C received a packet, and w.p. $(1-p)$, that it did not.
   * otherwise
   1 if at least one of $C(R)$ observed a linear combination of x_i, deduce that C received x_i.

Note that we can only observe packets received at the edge of the network, but not at the coding points. However, we can still infer that information from the observations at the receivers downstream from the coding point. The fact that we infer observations of the coding points from the observations of the leaves is what makes this algorithm suboptimal, while MINC in each partition is optimal.

We introduce the probability p in order to account for the fact that if none of the receivers in $C(R)$ receives a packet, this might be attributed to two distinct events: either the coding point C itself did not receive a packet, or C did receive a packet, which got subsequently lost in the descendent edges. For example, in

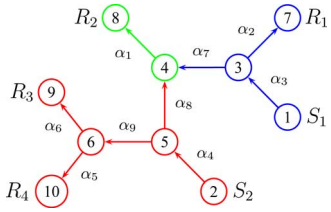


Fig. 6. Network topology with nine links. The link orientation depicted corresponds to nodes 1 and 2 acting as sources of probes.

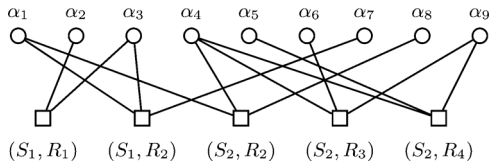


Fig. 7. Bipartite graph corresponding to the nine-link example tree in Fig. 6. It indicates which edges belong to which observable paths.

Fig. 6, consider the tree rooted at S_1; if R_2 receives x_1 or $x_1 + x_2$, we deduce that x_1 was received at node 4. If R_2 receives x_2, we deduce that x_1 was not received at node 4. If R_2 does not receive a probe packet, then, with probability $1 - p$, we assume that node 4 did not receive a probe packet. Ideally, p should match the probability that C correctly received a probe packet. This depends on the graph structure and on the loss probabilities downstream of C, and possibly prior information we may have about the link loss rates.

2) MINC-Like Heuristic: For every multicast node, we can use the MINC algorithm described in [2]. For every coding point, we can use RMINC described in Section V-B2.

Similarly to the subtree decomposition, we infer which probes have been received by an interior node i from observations at the downstream receivers. In particular, if at least one receiver downstream of i has received a probe with any content (the probe is from at least one source and potentially contains the XOR of probes from multiple sources), then we can infer that i received the packet. This can be used to compute the probability γ_i, in the terminology of MINC [2]. If no downstream receiver got any probe, we decide w.p. p whether the node i received a probe or not, exactly the same as in the subtree decomposition. The reductions shown in Fig. 23 use similar arguments and can serve as examples.

Different from the subtree decomposition, which estimates the α's locally in each subtree, we use the mapping from γ's to α's provided in MINC [2] to estimate the α's in the entire graph. This heuristic is optimal for multicast and reverse multicast configurations, and for configurations that are concatenations of the two, but suboptimal for any other configuration.

3) BP: We propose to use a BP approach, similar to what was proposed in [29]. Unlike the previous two heuristics, which are specific to tree topologies, the BP approach also applies to general graphs. The first step in the BP approach is to create the factor graph corresponding to our estimation problem. Fig. 7 shows the factor graph corresponding to the nine-link tree shown in Fig. 6. This is a bipartite graph: on one side, there are the links (variable nodes), whose loss rates we want to estimate; on the other side, there are the paths (function nodes) that are observed by each received probe. An edge exists in the factor graph between a link and a path, if the link belongs to this path in the original graph. Note that in tree topologies, there exists exactly one path for every source–receiver pair, while this is not the case in general graphs. Once the factor graph is created from the original graph, each received probe triggers message passing and results in an estimate of link success probabilities; these estimates from different probes are then combined using standard methods [29]. The result is an estimate $(\hat{\alpha}_e)$ of the actual success probability (α_e) of every link $e \in E$.

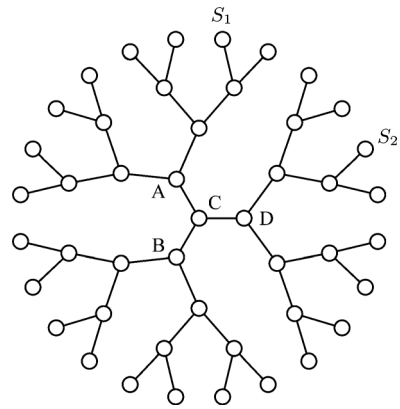


Fig. 8. Tree with 45 links used for simulating the suboptimal estimators.

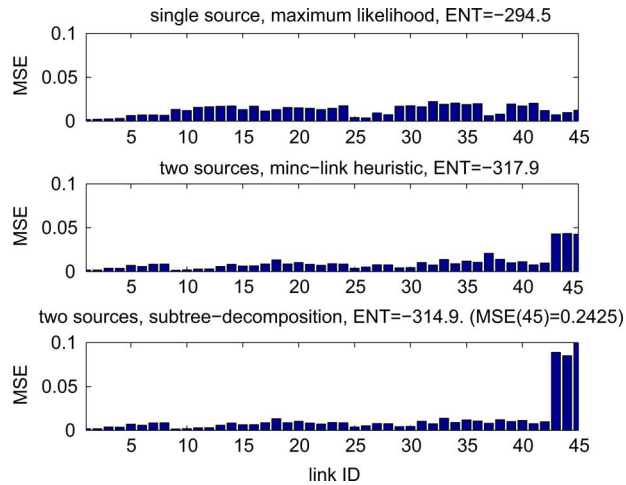


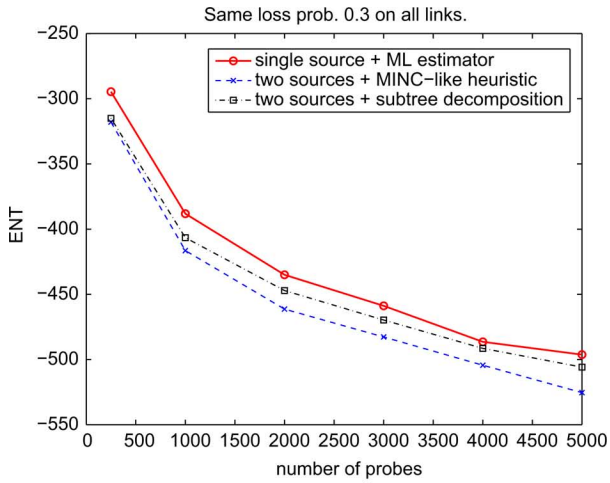
Fig. 9. Comparison of one multicast source + MLE versus two sources + network coding + suboptimal estimation (subtree decomposition and MINC-like heuristic). We show the MSE for each link in the 45-link topology.

E. Simulation Results

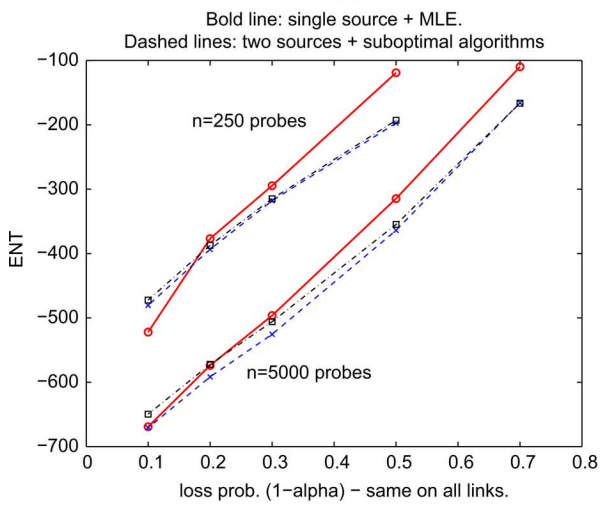
In this section, we evaluate the heuristic estimators via simulation and we compare them to each other as well as to multicast-based tomography. The main finding is that using more than one source helps: using multiple sources and network coding (even with suboptimal estimation) outperforms a single MT (even with optimal estimation), thus demonstrating the usefulness of our approach.[12]

Consider the 45-link topology shown in Fig. 8, where all links have the same success rate α. We will estimate α and compare

[12]Note that using more than one multicast sources, without network coding, would traditionally require to combine the observations from the two trees in a suboptimal way [3], thus further degrading the performance; that is why we skip the comparison and compare only against a single MT and optimal estimation, which has the best performance among the baselines.



(a)



(b)

Fig. 10. Comparison of one source with MLE, to two sources with suboptimal estimation (MINC-like and subtree estimation algorithms) for the 45-link tree. The comparison summarizes the error ENT over all links. (a) ENT versus number of probes. (b) ENT versus loss probability (same on all links).

different methods in terms of their estimation accuracy. First, we did simulations for $\alpha = 0.7$, a large number of probes, and repeated for many experiments. We looked at the MSE at each link. The results are shown in Fig. 9 for the following three algorithms:.

1) a single multicast source S_1 and ML estimation (top plot).
2) two sources S_1, S_2, network coding at the middle node C, and the MINC-like heuristic (middle plot).
3) the same two sources and coding point, with the subtree estimation algorithm (bottom plot).

Notice that in the case of two sources, the 45-link topology is partitioned into three subtrees: one rooted at A (where probe x_1 flows), another rooted at D (where x_2 flows), and a third one rooted at B (where $x_1 + x_2$ flows).

One can make several observations from this graph. First, using two sources and network coding, even with suboptimal estimators, performs better than using a single multicast source and an ML estimator. Indeed, the residual entropy (which is the metric that summarizes the MSE across all 45 links) is lower for two sources with the MINC-like (ENT $= -317.9$) and for the subtree-decomposition (ENT $= -314.9$) heuristics, than

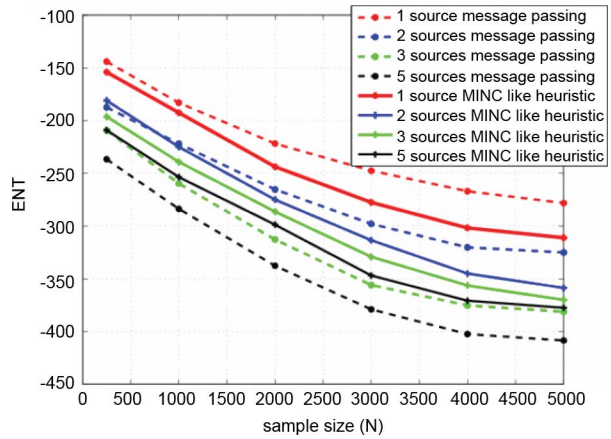


Fig. 11. Estimation error for two suboptimal algorithms (BP and MINC-like) for the 45-link tree. ENT versus number of probes.

it is for the single source MLE (ENT $= -294.5$). This illustrates the benefit of using multiple sources. Second, notice that the MSE for individual links is smaller in the lower two graphs than in the top graph, for all links except for links 43, 44, 45, for which it is significantly higher. This is no coincidence: links 43, 44, 45 are the middle ones (CA, CB, CD in Fig. 8). This is due to the fact that we cannot directly observe the packets received at the coding point C and we have to infer them from observations at the leaves of the subtree rooted at B. The performance of the heuristics could further improve by using the following tweak: we could estimate what probes are received at C, using observations from leaves not only in the subtree rooted at B, but also from the subtrees rooted at A and D.

The aforementioned simulations were for a single value of $\alpha = 0.7$. We then exhaustively considered several values of α (same on all links) and n (the number of probes). The results are shown in Fig. 10. We can see that, even with suboptimal estimation, using two sources consistently outperforms a single multicast source, even with MLE estimation. This is apparent in Fig. 10, where the ENT metric for the single source (drawn in bold lines) is consistently above the other two algorithms.[13]

In Fig. 11, we compare the MINC-like and the BP algorithms over the 45-link network, in terms of the ENT measure, and as a function of the number of probes n. Both algorithms yield better performance (lower ENT values) as the number of sources increases from one to five. The MINC-like algorithm performs better for the MT, in which case it coincides with the ML estimator, as well as for the two source tree. However, BP offers significantly better performance for the case of three and five sources. This trend can be explained by looking at the number of cycles in the factor graph. A cycle is created in the factor graph of a network configuration when 1) two different paths have more than one link in common and 2) a set of m paths, say W_m, covers a set E_m of m links, with each of the paths in W_m containing at least two links in E_m. As the factor graph becomes more and more cyclic, the performance of the sum–product algorithm degrades.

[13]Two observations on the ENT metric: First, the differences in the value of ENT are significant, although this is not visually obvious; recall that ENT is defined by taking the sum of the log of the MSE's. Second, ENT can be < 0; it is the differential entropy that matters.

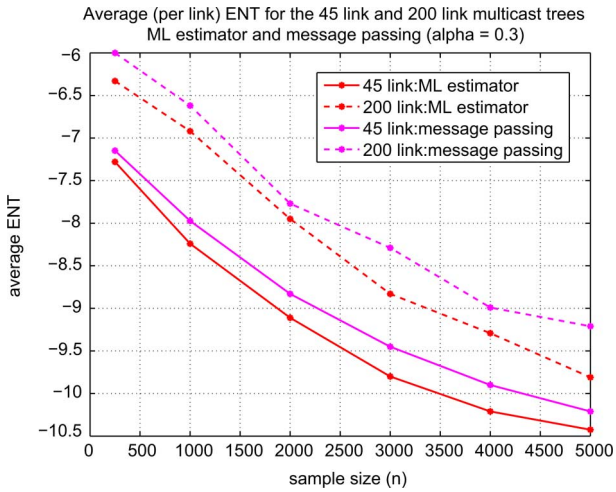


Fig. 12. Comparing BP to MLE for the 45-link and 200-link trees. ENT_{av} is ENT divided by the number of links.

Finally, in Fig. 12, we compare the performance of BP to ML estimation using a single source. We considered two trees: the 45-link and another, randomly generated 200-link tree. Because ENT captures the error over all links, and the two topologies have different numbers of links, we use ENT_{av} (defined as the ENT value divided by the number of network links) for a fair comparison of the two topologies. ENT_{av} for the 45-link tree is better (lower) than that of the 200-link tree for a given number of probes. We see that the BP algorithm closely follows the optimal ML estimator, for the range of number of probes and for both trees considered.

VI. General Topologies

In this section, we extend our approach from trees to general topologies. The difference in the second case is the presence of cycles, which poses two challenges: 1) probes may meet more than once and 2) probes may be trapped in loops. To deal with these challenges, in this section, we propose 1) an orientation algorithm for undirected graphs and 2) probe coding schemes, whose design is more involved than in trees.

The approach followed by prior work on tomography over general networks was to cover the graph with several multicast [3] and/or unicast probes [4], [6]. This approach faces several challenges. 1) The selection of multicast/unicast probes so as to minimize the total bandwidth (cost) is an NP-complete problem. 2) Having several probes from different source–destination paths cross the same link leads to bandwidth waste (especially close to sources or receivers). 3) Finding an optimal and/or practical method to combine the observations from different multicast/unicast paths is a nontrivial problem, addressed in a suboptimal way [3].

In contrast, using network coding allows us to measure all links with a single probe per link and brings the following benefits. 1) It makes the selection of routes so as to minimize the cost of linear complexity. 2) It eliminates the waste of bandwidth by having each link traversed by exactly one probe per experiment; furthermore, each network coded probe brings more information, as it observes several paths at the same time. 3) It does not need to combine observations from different experiments for estimation (as all links in the network are probed exactly once in one pass/experiment).

Because of the aforementioned features, the benefits of the network coding approach compared to traditional tomographic approaches are even more pronounced in general topologies than they were in tree topologies.

In this section, we describe the framework for link loss tomography in general graphs. In particular, we address the four subproblems mentioned in Section III-B: 1) identifiability of links; 2) how to select the routing; 3) how to perform the code design; and 4) what estimation algorithms to use. We evaluate our approach through extensive simulations on two realistic topologies: a small research network (Abilene), used to illustrate the ideas; and a large commercial ISP topology (Exodus), used to evaluate the performance in large graphs.

A. Identifiability

The identifiability of an edge given a fixed monitoring scheme follows from Theorem 4.1 in Section IV-B. CD is the edge we would like to identify, and we interpret the edges AC, BC, DE, and DF as paths that connect CD to sources and destinations. In particular, we are able to identify the link loss rate of edge CD from the probes collected at the receivers, if we can reconstruct the table associated with one of the cases in Fig. 2 (all tables are provided for completeness in Appendix B.1).

In a general topology, it is desirable to be able to know the state of all paths $\{\mathcal{P}\}$ that connect the sources to all receivers, at the end of each experiment. Let $\mathcal{P}(e)$ denote the set of paths that are routed from a source to a receiver, and employ an edge e. We refer to *path identifiability* as the ability to uniquely map each possible observation (received probes at all receivers) to the state of the paths $\{\mathcal{P}\}$, *i.e.*, which paths operated and which failed during the experiment. For a formal definition, see (29) and the related discussion. From the state of the paths, we can tell which links worked (w.p. 1) and which likely failed (with the associated probability). Moreover, knowing the state of the paths is particularly well suited for running the BP algorithm that we use for estimation of general graphs: indeed, message passing in the BP algorithm is triggered by giving the state of the paths as input. Therefore, we will attempt to make the maximum number of path states distinguishable, by appropriate selection of coding. The following example indicates how the selection of a coding scheme can allow more or less path states to be distinguishable at a receiver.

Example 4: Consider the network and edge orientation shown in Fig. 13; this is based on a real backbone topology (Abilene [30]), as will be discussed in detail in a later section. Node 1 acts as a source and node 9 as a receiver; assume that all intermediate nodes are only allowed to do XOR operations.

Note that paths P_3 and P_1 overlap twice: on edge E_2, and later on edge E_9. If all links in both paths function, the XOR operations "cancel" each other out, resulting in exactly the same observation with both paths being disrupted. More specifically, the following two events become indistinguishable: 1) all edges function: node 5 receives packet x_2 through edge E_7 and packet $x_2 + x_3$ through edge E_6, and sends packet x_3 through edge E_9 to the receiver; 2) edges E_4 and E_7 fail, while all other edges

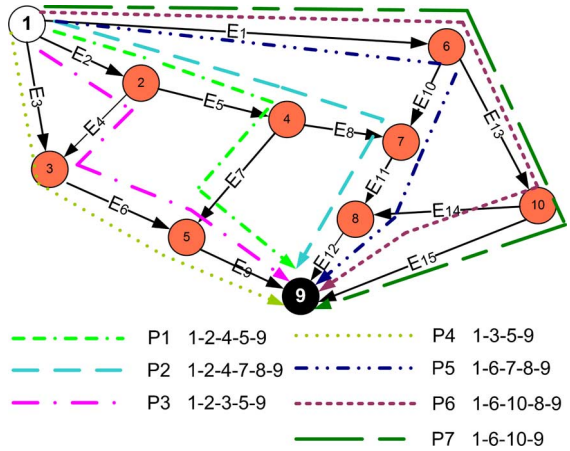


Fig. 13. Example of a general topology (Abilene). For one source (node 1), we show the orientation of edges, the resulting receiver (node 9), and the possible paths from the source to the receiver ($P_1, \ldots P_7$).

function: node 5 only receives packet x_3 from its incoming links, and again sends packet x_3 through edge E_9 to the receiver. On the other hand, if we allow coding operations over a larger alphabet, as in Example 6, these two events result in observing the distinct packets 1) $3x_2 + x_3$ and 2) x_3 at the receiver. □

B. Routing

First, we discuss the case where we want to estimate the success rate associated with a specific subset of links, and we express the corresponding optimization problem as a linear programming (LP) that can be solved in polynomial time. Then, we examine the practical special case where we are interested in measuring all links, and which will be the main focus of the rest of the section.

1) Minimum Cost Routing: Consider an arbitrary network topology, a given set S of nodes that can act as sources, a given set R of nodes that can act as receivers, and a set I of edges whose link success rates we want to estimate. Our goal is to estimate the success probability for all links in I at the minimum bandwidth cost. That is, we assume that a cost $C(e)$ is associated with each edge e that is proportional to the flow through the edge. We are interested in identifying the success rate α_e of edge $e \in I$. Let ρ be the rate of probes crossing that edge, in a manner consistent with the identifiability conditions for edge e.

Remarks: We note that the flow-based formulation of this problem does not rely on any major assumption. The accuracy of estimation depends only on the number of probes and not on the rate of the probe flows. The rates determine how quickly those *n* packets will be collected. For example, for smaller rates, it will take longer to collect the *n* packets. We also note that having flows coded together in an edge does not reduce the estimation accuracy. In fact, a coded packet observes more than one path, thus increasing the estimation accuracy versus bandwidth tradeoff.

The minimum cost routing problem was shown to be NP-hard, when performing tomography with MTs [31]. Indeed, the problem of even finding a single minimum cost Steiner tree is NP-hard. In contrast, we show here that if we use network coding, we can find the minimum cost routing in polynomial time. In the case of network coding, to ensure identifiability, we want to route flows so that the conditions in Theorem 4.1 are satisfied. We will consider the flow interpretation of paths in Theorem 4.1, *i.e.*, we will think of each path as a flow of fixed rate ρ. To ensure minimum cost, we want these flows to use the minimum resources possible.

In the following, we provide an LP formulation that allows us to solve the minimum cost cover problem in polynomial time, provided that we allow intermediate nodes to combine probes. We assume that there are no capacity constraints on the edges of the network, *i.e.*, we can utilize each edge as much as we want. This is a realistic assumption, since the rate ρ at which we send probe packets would be chosen to be a very small fraction of the network capacity, and nowhere close to consuming the whole capacity.

Intuition. Following an approach similar to [32], we introduce conceptual flows that can share a link without contending for the link capacity. We associate with each edge $e_i \in I$ one such conceptual flow f^i. We want each f^i to bring probe packets to link $e_i = u_i v_i \in I$, in a manner consistent with the conditions of Theorem 4.1 for edge e_i. We allow conceptual flows corresponding to different edges e_i to share edges of the graph without contention, and will measure through a total flow f the utilization of edges by probe packets. We use the condition $f^i \leq f$ to express the fact that each packet in f might be the linear combination of several packets of conceptual flows.

Notation. Let $C : E \rightarrow R^+$ be our cost function that associates a nonnegative cost $C(e)$ with each edge e. We are interested in minimizing the total cost $\sum_e C(e)f(e)$, where $f(e)$ is the flow through edge e. We also denote by $f_{\text{in}}(v)/f_{\text{out}}(v)$ the total incoming/outgoing flow of vertex v and with $f_{\text{in}}(e)/f_{\text{out}}(e)$ the total incoming/outgoing flow to edge e. The same notation but with the superscript i, *e.g.*, $f^i_{\text{in}}(u)$ has the same meaning but specifically for conceptual flow f^i. We connect all nodes in $S = \{S_i\}$ to a common source node $\mathcal{S}$ through a set of infinite-capacity and zero-cost edges $E_S = \{\mathcal{S}S_i\}$. Similarly, we connect the nodes in $R = \{R_i\}$ to a common node $\mathcal{R}$ using an infinite-capacity and zero-cost set of edges $E_R = \{R_i\mathcal{R}\}$.

We summarize the LP program for minimum cost routing in the following:

$$\min \textstyle\sum_e C(e)f(e)$$
$$f(e) \leq \rho \quad \forall e \in E - E_S - E_R$$
$$f(e) = \rho \quad \forall e \in I$$
$$\text{Each conceptual flow } f^i \text{ corresponding to } e_i = u_i v_i \text{ satisfies the constraints}$$
$$f^i(e) \leq f(e) \quad \forall e \in E - e_i$$
$$f^i(e) \geq 0 \quad \forall e \in E$$
$$f^i_{\text{in}}(\mathcal{S}) = 0$$
$$f^i_{\text{out}}(\mathcal{R}) = 0$$
$$f^i_{\text{in}}(u) = f^i_{\text{out}}(u) \quad \forall u \in V - \{\mathcal{S}, \mathcal{R}, u_i, v_i\}$$
$$f^i_{\text{in}}(u_i) \geq \rho \quad /*\text{conceptual flow of rate at least } \rho \text{ gets into } (u_i, v_i)*/$$
$$f^i_{\text{in}}(u_i) + f^i_{\text{out}}(u_i) \geq 3\rho$$
$$f^i_{\text{out}}(v_i) \geq \rho \quad /*\text{conceptual flow of rate at least } \rho \text{ gets out of } (u_i, v_i)*/$$
$$f^i_{\text{in}}(v_i) + f^i_{\text{out}}(v_i) \geq 3\rho.$$

The idea is to lower bound the probe rate $f(e)$, in edge e, given the conceptual flows and the condition $f^i(e) \leq f(e)$. Solving this LP will give us a set of flows and paths, for each edge $e = (u_i, v_i)$. To ensure identifiability, we need to additionally select a coding scheme so that the flows arriving and leaving at u_i and v_i utilize distinct packets, *i.e.*, from the observable events at the sink, we can reconstruct for edge e the probability of the events of one of the cases 1–4 in identifiability.

In summary, the minimum cost routing problem, so as to identify the loss rates of a predefined set of edges I, can be solved in linear complexity when network coding is used, while the same problem is NP-hard without network coding.

2) Routing (Including Source Selection and Link Orientation) for Measuring all Links: If we are interested in estimating the success rate of *all* identifiable edges of the graph, as opposed to just a restricted set I as in the previous section, we do not need to solve the above LP. We can simply have each source send a probe and each intermediate node forward a combination of its incoming packets to its outgoing edges. This simple scheme utilizes each edge of the graph exactly once per time slot (set of probes sent by the sources) and, thus, requires the minimum total bandwidth. Moreover, if an edge is identifiable, there exists a coding scheme that allows it to be so. Example 4 and Fig. 13 demonstrate such a situation: the source (node 1) sends one probe per experiment, which gets routed and coded inside the network, crossing each link exactly once, and eventually arriving at the receiver (node 9).

Challenge I: Cycles: One novel challenge we face in general topologies compared to trees is that probes may be trapped in cycles. Indeed, if network nodes simply combine their incoming packets and forward them toward their outgoing links, in a distributed manner and without a global view of the network, then probes may get trapped in a positive feedback loop (cycle) that consumes network resources without aiding the estimation process. The following example illustrates such a situation.

Example 5: Consider again the network shown in Fig. 13, but now assume that the orientation of edges E_4 and E_6 was reversed. Thus, edges E_4, E_5, E_7, and E_6 create a cycle between nodes 2, 4, 5, and 3. The probe packets injected by nodes 3 and 2 would not exit this loop. □

To address this problem, we could potentially equip intermediate nodes with additional functionalities, such as removal of packets that have already visited the same node. This is not practical because it requires keeping state at intermediate nodes; furthermore, such operations would need to be repeated for every set of probes, leading to increased processing and complexity.

We take a different approach: we remove cycles. Starting from an undirected graph $G = (V, E)$, where the degree of each node is either one (leaves) or at least three (intermediate nodes), we impose an orientation on the edges of the graph so as to produce a directed acyclic graph (DAG). Our approach is only possible if we are given some flexibility to choose nodes that can act as sources or receivers of probe packets, among all nodes, or among a set of candidate nodes.

There are many algorithms one can use to produce a DAG. In the following, we propose our own orientation algorithm, Algorithm 3, that in addition to removing cycles also achieves some goals related to our problem. In particular, starting from a set of nodes that act as senders $S \subset V$, Algorithm 3 selects an orientation of the graph and a set of receivers so that 1) the resulting graph is acyclic, 2) a small number of receiver nodes is selected,[14] which is desired for the efficient data collection, and 3) the resulting DAG leads to a factor graph that works well with BP estimation algorithms. Algorithm 3 guarantees identifiability, but is heuristic with respect to criteria 2) and 3); it is important to note, however, that optimizing for criterion 3) is an open research problem (as discussed in Section VI-D).

Algorithm 3 Orientation Algorithm: Given Graph $G = (V, E)$ and Senders $S \subset V$, Find Receivers $R \subset V$ and Orientation $\forall\ e\ \in\ E$ s.t. There are no Cycles and all Edges are Identifiable

1: **for all** undirected edges $e = (s, v_2),\ s \in S$ **do**
2: Set outgoing orientation $s \rightarrow v_2$
3: **end for**
4: $R = \{s \in S \text{ that have incoming oriented edges}\}$
5: $V_1 = S$;
6: $V_2 = \{v_2 \in V - V_1 : s.t.\ \exists\ edge\ (v_1, v_2)\ from\ v_1 \in V_1\}$
7: **while** $V_2 \neq \emptyset$ **do**
8: Identify and exclude receivers: find $r \in V_2$ without unset edges: $R := R \bigcup \{r\}$; $V_2 := V_2 - \{r\}$
9: Find nodes $U_1 \subset V_2$ that have the smallest number of edges with unset orientation.
10: Find nodes $U_2 \subset U_1$ that have the minimum distance from the sources S. Choose one of them: $v^* \in U_2$.
11: Let $E^* = \{(v^*, w) \in E \text{ s.t. } w \in V - V_1\}$
12: **for all** undirected edges $(v^*, w) \in E^*$ **do**
13: set direction to $v^* \rightarrow w$
14: **end for**
15: Update $V_1 := V_1 \bigcup \{v^*\}$
16: Update
$V_2 := \{(\text{none-}V_1) \text{ nodes one edge away from current } V_1\}$
17: **end while**

We now describe Algorithm 3. We sequentially visit the vertices of the graph, starting from the source, and selecting an orientation for all edges of the visited vertex. This orientation can be thought of as imposing a partial order on the vertices of the graph: in a sense, no vertex is visited before all its parent vertices in the final directed graph.

Lines 1–3 attempt to set all links attached to the sources as outgoing. If we allow an arbitrary selection of sources, we may fall into cases where sources contain links to other sources. In this case, one of the sources will also need to act as a receiver, *i.e.*, we allow the set S of sources and the set R of receivers to overlap. In the main part of the algorithm, nodes are divided into three sets.

[14]Given a set of sources, one can always produce an orientation and a set of receivers that comprise a DAG, which is what Algorithm 3 does. Conversely, given a set of receivers, one can always produce an orientation and a set of sources that comprise a DAG. If both the sets of sources and receivers are fixed, a DAG may not always exist, depending on the topology.

1) A set of nodes V_1, which we have already visited and have already assigned orientation to all their attached edges. Originally, $V_1 := S$.
2) A set of nodes V_2, which are one edge away from nodes in V_1 and are the next candidates to be added to V_1.
3) The remaining nodes are either receivers R or just nodes not visited yet $V_3 := V - V_1 - V_2 - R$.

In each step of the algorithm, one node $v^* \in V_2$ is selected, all its edges that do not have an orientation are set to outgoing, and v^* is added to $V_1 := V_1 \bigcup \{v^*\}$. Note that the orientation of the edges going from V_1 to V_2 is already set. However, a node $v \in V_2$ may have additional unset edges; if it does not have unset edges, then it becomes a receiver $R := R \bigcup \{v\}$. We include two heuristic criteria in the choice of $v^* \in V_2$: 1) first, we look at nodes with the smallest number of unset edges; 2) if there are many such nodes, then we look for the node with the shortest distance from the sources S; if there are still many such nodes, we pick one of them at random. The rationale behind criterion 1) is to avoid creating too many receivers. The rationale behind criterion 2) is to create a set of paths from sources to receivers with roughly the same path length. The criteria 1) and 2) are just optimizations that can affect the estimation performance.[15] The algorithm continues until all nodes are assigned to either R or V_1.

Lemma 6.1: Algorithm 3 produces an acyclic orientation.

Proof: At each step, a node is selected and all its edges which do not have a direction are set as outgoing. This sequence of selected nodes constitutes a topological ordering. At any point of the algorithm, there are directed paths from nodes considered earlier to nodes considered later. A cycle would exist if and only if for some nodes v_i and v_j, v_j is selected at step $j > i$ and the direction on the undirected edge (v_i, v_j) is set to $v_i \leftarrow v_j$. This is not possible since if there were an edge (v_i, v_j), it would have been set at the earlier step i at the opposite direction $v_i \rightarrow v_j$. Therefore, the resulting directed graph has no cycle. It is possible, however, that there are nodes with no outgoing edges, which become the receivers. ■

We note that the key point that enables us to create an acyclic orientation graph for an undirected graph is that we allow the receivers to be one of the outputs of the algorithm. Note that a similar algorithm can be formulated for the symmetric problem, where the receivers R are given and the orientation algorithm produces a (reverse) orientation and a set of sources S, s.t. that there are no cycles. However, if both S and R are fixed, there is no orientation algorithm that guarantees the lack of cycles for all graphs.

Lemma 6.2: Algorithm 3 guarantees identifiability of every link in a general undirected graph consisting of logical links (*i.e.*, with degree ≥ 3), and for any choice of sources.

Proof: The proof follows directly from the fact that the degree of each node is greater than or equal to three (assuming logical links only), each edge bringing or removing the same amount of flow. Thus, either the node is a source or a receiver, or the conditions of Theorem 4.1 and Fig. 2 are satisfied. ■

[15]One could use different criteria to rank the candidates v^*, so as to enforce additional desirable properties. Here, we used shortest path from the sources to impose a breath-first progression of the algorithm and paths with roughly the same length. One could also use other criteria to optimize for the alphabet size and/or the complexity and performance of the estimation algorithms.

C. Code Design

Challenge II: Code Design affects Identifiability: Another novel challenge that we face in general topologies compared to trees is that simple XOR operations do not guarantee path identifiability, as we saw in Example 4. We deal with this challenge using linear operations over higher field sizes as the following example illustrates.

Example 6: Let us revisit the general topology shown in Fig. 13 and briefly discussed in Example 4. Node 1 acts as a source: in each experiment, it sends probes x_1, x_2, and x_3 through its outgoing edges E_1, E_2, and E_3, respectively. Nodes 2, 4, 6, 10 simply forward their incoming packets to all their outgoing links. Node 3 performs coding operations as follows: if within a predetermined time window it only receives probe packet x_2, it simply forwards this packet. The same holds if it only receives probe packet x_3. If, however, it receives both packets x_2 and x_3, it linearly combines them to create the packet $x_2 + x_3$ that it then sends through its outgoing edge E_6. Nodes 5, 7, and 8 follow a similar strategy. If all links are functioning, node 5 sends packet $3x_2 + x_3$, node 7 sends packet $x_1 + x_2$, and, finally, node 8 sends packet $3x_1 + x_2$. The receiver node 9 observes, in each experiment, three incoming probe packets. For example, if it only observes the incoming packet x_3, it knows that all paths from the source S have failed, apart from path P_4. Therefore, it infers that no packets were lost on edges E_3, E_6, and E_9. □

More generally, we are interested in practical code design schemes that allow for identifiability of all edges in general topologies. We will achieve this goal by designing for path identifiability, which is a different condition. In particular, we are interested in coding schemes that allow us to identify the maximum number of path states. This can be achieved by mapping the failure of each subset of paths to a distinct probe observed at the receivers. For this to be possible, 1) the alphabet size must be sufficiently large and (ii) the coding coefficients must be carefully assigned to edges.

Recall that receiver nodes only have incoming edges. Let e_{R_j} be an edge adjacent to a receiver R_j and $\mathcal{P}(e_{R_j})$ be the set of paths that connect all source nodes to receiver R_j, and have e_{R_j} as their last edge. We say that a probe coding scheme allows maximum path identifiability if it allows the receiver R_j, by observing the received probes from edge e_{R_j} at a given experiment, to determine which of the $\mathcal{P}(e_{R_j})$ paths have been functioning during this experiment and which have not.

1) Alphabet Size: There is a tradeoff between the field size and path identifiability. On one hand, we want a small field size mainly for low computation (to do linear operations at intermediate nodes) and secondarily for bandwidth efficiency (to use a few bits that can fit in a single probe packet). In practice, the latter is not a major problem, because for each probe, we can allocate as many bits as the maximum IP packet size, which is quite large in the Internet.[16] However, for computation purposes, it is still important that we keep the field size as small as possible. On the other hand, a larger field size makes it easier to achieve path identifiability.

[16]The maximum transmission unit on the Internet is at least 575 bytes (4800 bits), and up to 1500 bytes (12 000 bits), including headers. However, in simulation of realistic topologies, we did not need to use more than 18 bits.

For maximum path identifiability, there is the following loose lower bound on the required alphabet size.

Lemma 6.3: Let $G = (V, E)$ be acyclic and let $\mathcal{P}_m$ denote the maximum number of paths sharing an incoming edge of any receiver R_j, *i.e.*, $\mathcal{P}_m = \max_{e_{R_j}} \mathcal{P}(e_{R_j})$. The alphabet size must be greater than or equal to $\log \mathcal{P}_m$.

Proof: Assume that one of the $\mathcal{P}_m$ paths is functioning, while all the others are not. Since two paths cannot overlap in all edges, there exists a set of edge failures such that this event occurs. For the receiver to determine which of the $\mathcal{P}_m$ paths function and which ones fail, it needs to receive at least $\mathcal{P}_m$ distinct values. Essentially, the field size should be large enough to allow for distinguishing among all possible paths arriving at each receiver. Therefore, we need a field size $q \geq \mathcal{P}_m$. ■

What the aforementioned lemma essentially counts is the number of distinct values that we need to be able to distinguish. This can be achieved using either scalar network coding over a finite field F_q of size q, or vector linear coding with vectors of appropriate length. See, *e.g.*, [24] for an application to the multicast scenario, where scalar network coding over a finite field of size q was treated as equivalent to vector network coding over the space of binary vectors of length $\log q$.

The reader will immediately notice that there is an exponential number of paths and failure patterns. We would like to note that this is not unique to our work, but inherent to tomography problems that try to distinguish between exponentially large number of configurations, *e.g.*, transfer matrices and their failure patterns in the passive tomography [17], [18]. Even in that case, simulations of large topologies, such as Exodus, showed that a moderate field size is sufficient in practice. However, in our case of active tomography, a potentially large alphabet size is needed only if one insists to infer the loss rates on *all links simultaneously*. In practice, one can infer the loss rates on links one by one, by carefully selecting the probes and measuring only the corresponding paths, thus creating the "five-link" motivating example, where XOR operations are sufficient.

2) Code Design: Having a large alphabet size is necessary but not sufficient to guarantee path identifiability. We also need to assign coefficients $\{c_h\}$ so that the failure of every subset of paths leads to a distinct observable outcome (received probe content). Here, we discuss how to select these coefficients.

Consider a particular incoming edge e_{R_j} to a receiver R_j and let m be the number of paths arriving at this edge from source S_i. Consider one specific path h that connects source S_i to R_j via edges $e_{h_1}, e_{h_2}, \ldots, e_{R_j}$. The contribution P_h from path h to the observed probe is what we call a *path monomial*, *i.e.*, the product of coefficients on all edges across the path and of probe $\mathcal{X}_{S_i}$ sent by source S_i:

$$P_h = c_{h_1} \cdot c_{h_2} \cdot \cdots \cdot c_{R_j} \cdot \mathcal{X}_{S_i}.$$

For simplicity, we use P_h to denote both a path and the corresponding path monomial. Note that each path consists of a distinct subset of edges; as a result, no path monomial is a factor of any other path monomial. We can collect all the monomials P_h in a column vector $\vec{P}_{e_{R_j}} = (P_1, P_2, \ldots, P_m)$.

If all paths arriving at edge e_{R_j} are working (no link fails), the received probe at that edge is the summation of the contributions $\vec{P} = (P_1, P_2, \ldots, P_m)$ from all m paths:

$$\text{Probe received through } e_{R_j} \text{ (when no loss)} = P_1 + P_2 + \cdots + P_m.$$

In practice, however, any subset of these m paths may fail due to loss on some links and the received probe becomes the summation of the subset of paths that did not fail. Let $\vec{X} = (x_1, x_2, \ldots, x_m)$ be the vector indicating which paths failed: $x_k = 0$ if path k failed and 1 otherwise. Therefore, the probe received through e_{R_j}, in the case of loss, is:

$$\text{Probe received through } e_{R_j} \text{ (when loss)} = \vec{X} \cdot \vec{P} = \sum_{k=1}^{m} x_k \cdot P_k$$

where $\vec{X}$ is the indicator vector corresponding to the loss pattern, *i.e.*, has entry zero if a path fails, and one otherwise. The vector $\vec{X}$ can take 2^m possible values; let $\vec{X}_k$ denote the kth possible value, $k = 0, \ldots, 2^m - 1$. To guarantee identifiability, no two subsets k, l of failed paths should lead to the same observed probe $\vec{X}_k \cdot \vec{P} \neq \vec{X}_l \cdot \vec{P}$.

Therefore, a successful code design should lead to 2^m distinct probes, one corresponding to a different subset of paths failing. In other words, to guarantee identifiability, the coefficients $\{c_e\}_{e \in E}$ assigned to edges E should be such that $\vec{X}_k \cdot \vec{P} - \vec{X}_l \cdot \vec{P} \neq 0, \ \forall \ k, l = 0, \ldots, 2^m - 1$. We can write all these constraints together as follows, which is essentially the definition of *path identifiability*, mentioned in the beginning of Section VI-A:

$$\prod_{k,l=0,\ldots 2^m-1} (\vec{X}_k \cdot \vec{P}_{e_{R_j}} - \vec{X}_l \cdot \vec{P}_{e_{R_j}}) \neq 0. \tag{29}$$

Since each $P_h = c_{h_1} \cdot c_{h_2} \cdot \cdots \cdot c_{R_j} \cdot \mathcal{X}_{S_i}$ is a monomial, with variables the coding coefficients $\{c_e\}_{e \in E}$, the left-hand side in (29) is a multivariate polynomial $f(c_1, c_2, \ldots, c_{|E|})$ with degree in each variable at most $d \leq 2^m$.

Lemma 6.4: The multivariate polynomial $f(c_1, c_2, \ldots, c_{|E|})$ at the left-hand side of (29) is not identically zero.

Proof: The "grand" polynomial is not identically zero because each factor in the product $(\vec{X}_k \cdot \vec{P}_{e_{R_j}} - \vec{X}_l \cdot \vec{P}_{e_{R_j}})$ is a nonzero polynomial in $\{c_h\}$. Indeed, $\vec{X}_k$ and $\vec{X}_l$ differ in at least one position, say g, corresponding to a monomial P_g. Consider the following assignment for the variables $\{c_h\}$. Assign to all the variables in this monomial a value equal to one. Assign to all other variables $\{c_h\}$ a value of zero. Since no monomial is a factor of any other monomial, this implies that the vector $\vec{P}_{e_{R_j}}$ takes value one at position g, and zero everywhere else. Thus, this assignment results in a nonzero evaluation for the polynomial $(\vec{X}_k \cdot \vec{P}_{e_{R_j}} - \vec{X}_l \cdot \vec{P}_{e_{R_j}})$, and as a result, this cannot be identically zero. ■

Up to now, we have considered paths that employ the same incoming edge. We can repeat exactly the same procedure for all incoming edges, and generate, for each such edge, a polynomial in the variables $\{c_h\}$. Alternatively, we could also find these polynomials by calculating the transfer matrix between the

sources and the specific receiver node using the state-space representation of the network and the algebraic tools developed in [33]. Either way, the code design consists of finding values for the variables $\{c_h\}$ so that the product of all polynomials, f, evaluates to a nonzero value. There are several different ways to find such assignments, extensively studied in the network coding literature, *e.g.*, [34]–[36]. One way to select the coefficients is randomly, and this is the approach we follow in the simulations. In that case, it is well known that we can make the probability that $f(c_1, c_2, \ldots, c_{|E|}) = 0$ arbitrarily small, by selecting the coefficients randomly over a large enough field.[17]

Deterministic Operation: We emphasize that although the coefficients may be selected randomly (at setup time), the operation of intermediate nodes (at run time) is deterministic. At setup time, we select the coefficients and we verify the identifiability conditions, and select new coefficients if needed for the conditions to be met. After the selection is finalized, we learn the coefficients and use the same ones at each time slot. Learning the coefficients is important in order to be able to infer the state of the paths and links.

State Table and Complexity Issues: Once the coefficients are randomly selected, we need to check whether the constraints summarized in (29) are indeed satisfied. If they are satisfied, the code design guarantees identifiability; if they are not satisfied, then we can make another random selection and check again. One could also start from a small field size and increase it after a number of failed trials.

The evaluation of (29) above requires to check an exponential number of constraints, up to 2^m, where m is the number of paths for a triplet (source, receiver, edge at receiver). Because the current orientation algorithm does not exclude any edges in the process of building the DAG, we might end up with a large number of paths depending on the connectivity of the topology and the selection of the sources[18]. This motivated us to look into ways for reducing the number of paths per triplet[19]. Even putting aside the exponential number of paths for a moment, the problem is essentially a subset sum: we receive a symbol at a receiver and we would like to know which combinations of nonfailed paths add up to this number. This is a well-known NP-hard problem.

This being said, we do not expect this to be a source of high complexity in practice for several reasons. First, the algorithm that maps the received symbol to a state of paths can be run offline and the table can be computed and stored. This is a static scenario, since coding coefficients remain the same across scenarios. Therefore, we incur setup complexity once in the beginning, but not during run time. All we need to do every time we receive a symbol is just a table lookup, which is inexpensive $(O(1))$, when implemented using hash tables. Second, this design is only necessary if one wants to infer *all* links at the same time, which may be an overkill in practice. The most typical use of our framework in practice will be for inferring the loss rates of a few congested specific links of interest, in which case we do not need to keep track of the state of all paths, and the size of the table reduces.

[17]From the Schwartz–Zippel Lemma [34], which has been instrumental for network coding [36], we know the following. If $f(c_1, c_2, \ldots, c_{|E|})$ is a nontrivially zero polynomial with degree at most d in each variable, and we choose $\{c_e\}_{e \in E}$ uniformly at random in F_q with $q > d$, then the probability that $f(c_1, c_2, \ldots, c_{|E|}) = 0$ is at most $1 - (1 - \frac{d}{q})^{|E|}$.

[18]For example, for the Abilene topology shown in Fig. 13, with one source, there were at most three paths per (S_i, R_j, e_{R_j}) triplet, but for the larger Exodus topology (described in Section VI-E) with five sources, the average and maximum number of paths per triplet were 9 and 25, respectively (for a specific selection of sources in both topologies).

[19]For example, if we are willing to accept less than 100% path identifiability, we can randomly assign coefficients without checking for identifiability conditions. From the observed probes at the receivers, we then infer the subset of paths that failed by looking up a table which is precomputed by solving a subset sum problem. If we identify one or more subsets of paths that when failing lead to the same observed probe, we can use a heuristic, *i.e.*, pick one of the candidate subsets, their union or intersection. We then feed the state of the paths to the BP estimation algorithm. This is the approach we follow in the simulation section.

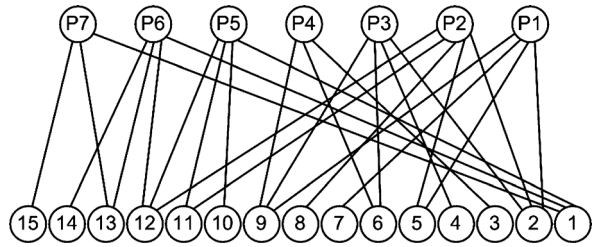


Fig. 14. Factor graph corresponding to the Abilene graph (shown in Fig. 13). It maps the 15 links to the seven observable paths at the single receiver (9). It is used for the BP estimation algorithm.

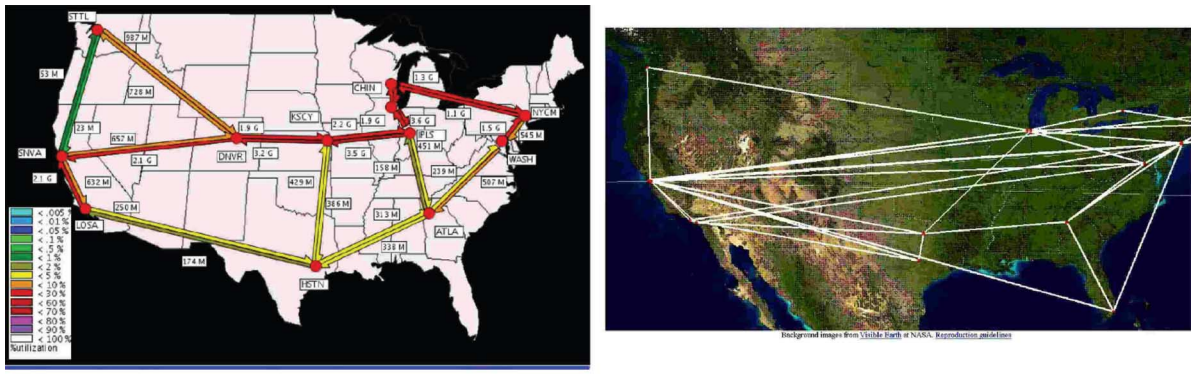
D. Loss Estimation Using BP

For our approach to be useful in practice, we need to employ a low complexity algorithm that allows to quickly estimate the loss rate on every link from all the observations at the receiver. Because MLE is quite involved for general graphs, especially large ones, we use a suboptimal algorithm instead; in particular, we use the BP approach that we also used for trees, see Section V-D3.

There are two steps involved in the algorithm for each round of received probes. First, from the observations, we need to deduce the state of the paths traversed by these probes, as described in Algorithm 4. The second step is to use the BP algorithm, to approximate ML estimation. Once we know which paths worked and which failed in this round, we feed this information into the factor graph, which triggers iterations, and leads to the estimate of the success rate. Similarly to trees, the factor graph is again a bipartite graph, between links and paths containing these links. For example, Fig. 14 shows the bipartite graph corresponding to the Abilene topology of Fig. 13, which we have been discussing in all the examples in this section.

Algorithm 4 Deduce State of the Paths From the Observations

for all $S_i \in Senders$ **do**
  for all $R_j \in Receivers$ **do**
    for all incoming links e_{R_j} **do**
      Map the observed probe to the state of all paths from S_i to R_j coming through link e_{R_j}.
    end for
  end for
end for



(a) (b)

Fig. 15. Topologies used in simulations (a) Left: Abilene Backbone Topology (small research network) (b) Right: Exodus POP Topology (large ISP).

The main difference in the general graphs compared to the trees is that there are multiple (instead of exactly one) paths between a source and a receiver; this has two implications. The first implication is that the design of the coding scheme must allow us to deduce the state of these multiple paths between a source, a receiver, and an incoming edge at the receiver (S_i, R_j, e_{R_j}); this has been extensively discussed in the previous section on code design. The second implication is that there are more cycles in the factor graph of a general graph, which affects the estimation accuracy of the BP algorithm.

In general, the performance of the BP algorithm depends on the properties of the factor graph. Several problems have been identified in the BP literature depending on the existence of cycles, the ratio of factors versus variables (*e.g.*, links per path), and other structural properties (stopping sets, trapping sets, diameter). Fixing such BP-specific problems are outside the scope of this paper and is a research topic on its own. However, we did address two of the aforementioned problems, using existing proposals from the BP literature. First, for performance enhancement in the presence of cycles in the factor graph, we used a modification of the standard BP, similar to what was proposed in the context of error correcting codes [37]. The idea is to combat the overestimation of beliefs by introducing a multiplicative correction factor $a < 1$ for messages passing between variables (links) and factors (paths).[20] Second, we designed the orientation algorithm to traverse the actual topology in a breadth-first manner in order to produce short paths and, thus, small ratio of links per path in the factor graph, which has a good effect on the BP performance. More generally, we note that the properties of the factor graph depend on the orientation algorithm. One could optimize the orientation algorithm to achieve desired properties of the factor graph. In this paper, we have not done modifications other than the two aforementioned because 1) the overall estimation worked well in all the practical cases we tried, and 2) the design of a factor graph for better BP performance is a research topic on its own and outside the scope of this work.

[20]In the same way, we could also use an additive correction factor instead. Making those factors adaptive could give even better results. In the same paper [37], additional modifications of the factor graph (junction tree algorithm, and generalized BP) to deal with cycles have been proposed, which we did not implement in this paper. Other possible modifications of the BP include: [38], a multistage iterative decoding algorithm that combines BP with ordered statistic decoding, and reaches close to the performance of MLE although with a higher complexity than BP; and [39], which uses a probabilistic schedule for message passing between variable nodes and check nodes in the factor graph instead of simple message flooding at every iteration.

E. Simulation Results

We now present extensive simulation results over two realistic topologies.

1) Network Topologies: We used two realistic topologies for our simulation, namely the backbones of Abilene and Exodus shown in Fig. 15. Abilene is a high-speed research network operating in the U.S. and information about its backbone is available online [30]. Exodus is a large commercial ISP, whose backbone map was inferred by the Rocketfuel project [40]. Both topologies were preprocessed to create logical topologies that have degree at least 3. For Exodus, nodes with degree 2 were merged to create a logical link between the neighbors of such nodes, while nodes with degree 1 were filtered; the resulting logical topology contains 48 nodes and 105 links. For the Abilene topology, due to its small size, in addition to merging some links in tandem, more links were added; the modified topology comprises of 10 nodes and 15 links, and is the one shown in Fig. 13 and used as an example of a general topology throughout Section VI.

For all simulations, the link losses on different links are assumed independent, and may take large values as they reflect losses on logical links, comprising of cascades of physical links, as well as events related to congestion control within the network.

2) Results on the Orientation Algorithm: In Fig. 16, we consider the Exodus topology and we run the orientation algorithm for all possible placements of one and two sources; we call each placement an "instance." We are interested in the following properties of the orientation produced by Algorithm 3 :

1) the number of receivers: a small number allows for local collection of probes and easier coordination.
2) the number of distinct paths per receiver: this relates to the alphabet size and it is also desired to be small.
3) the number of paths per link and links per path: these affect the performance of the BP algorithm.

Fig. 16 shows the aforementioned four metrics, sorting the instances first in increasing number of receivers and then in increasing paths/receiver. The following observations can be

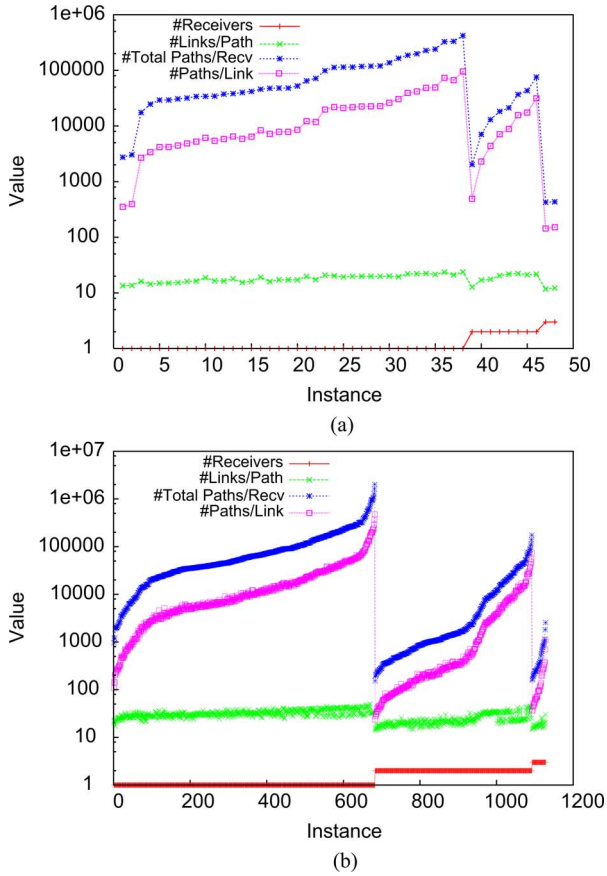


Fig. 16. Running the orientation algorithm on the Exodus topology. (a) All possible placements of one source. (b) All possible placements of two sources.

TABLE III
PROPERTIES OF THE ORIENTATION GRAPHS PRODUCED BY ALGORITHM 3 FOR DIFFERENT TOPOLOGIES AND CHOICES OF SOURCES

Topology	Srcs-Recvs	Coding Points	Links / Path	Paths / Link	Edge Disj. Paths
Abilene	{1}-{9}	4	3.85	1.8	3
	{5}-{6}	4	3.71	1.73	3
	{9}-{2}	4	4.28	2.0	2
	{1,9}-{7}	5	3.25	1.73	4
	{3,6}-{9}	5	4	2.13	4
	{9,6}-{4}	5	3.25	1.73	4
	{1,5,9}-{7}	5	3.2	2.13	5
	{1,4,10}-{9}	6	3	2.33	6
Exodus	{39,45}-{30,40}	25	9.47	56.47	4

made. First, the number of receivers produced by our orientation algorithm is indeed very small, as desired. Second, the number of links per path is almost constant, because by construction, the orientation algorithm tries to balance the path lengths. Third, the paths/receiver and paths/link metrics, which affect the alphabet size and the quality of the estimation, can be quite large; however, they decrease by orders of magnitude for configurations with a few receivers; therefore, such configurations should be chosen in practice. Finally, Table III considers different choices of sources in the (modified) Abilene and Exodus topologies, and shows some properties of the produced orientation.

3) Evaluation of Random Code Design for Real Topologies: In this section, we simulate *random code design schemes* for the example topologies of Abilene and Exodus.

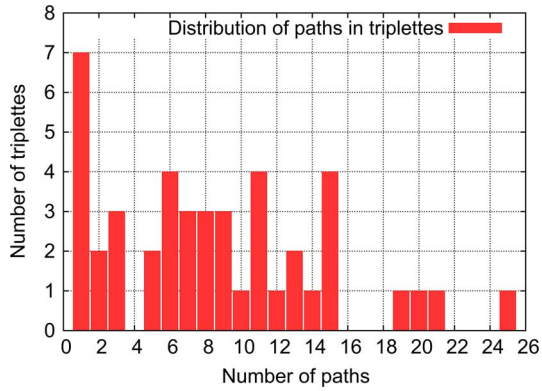


Fig. 17. Distribution of the number of paths for all triplets (S_i, R_j, e_{R_j}) for the Exodus topology.

Consider a particular incoming edge e_{R_j} to a receiver R_j and let m be the number of paths arriving at this edge from the same source S_i. If two subsets of paths lead to the same probe, then they are indistinguishable, which leads to lack of identifiability. In practice, since many of the paths for a triplet (S_i, R_j, e_{R_j}) share links between them, we have much less than 2^m possible distinct probes. The exact number depends on the connectivity of the topology. In the simulations, the content of the probe from each subset of paths is used as a key to a hash table. If two subsets lead to the same probe, then they will end up into the same bucket. The number of unique buckets in the hash table gives us the number of different combinations of failed/nonfailed paths that are distinguishable from each other. We normalize this number by the total number of possible distinct subsets, and we call this number the probability of success (path identifiability) of the code design for this particular triplet (S_i, R_j, e_{R_j}).

For the *Abilene topology* (10 nodes, 15 links), using one source and the orientation algorithm, we obtained a DAG with one receiver (see Fig. 13). The maximum number of paths observed for an incoming edge at the receiver was 3. A random choice of coding coefficients over a finite field of size 2^6 was sufficient to achieve 100% identifiability of all paths on all edges.

For the *Exodus topology* (48 nodes, 105 links), we select five sources, apply the orientation algorithm, and get three receivers. Fig. 17 shows the distribution of the number of paths for all triplets (S_i, R_j, e_{R_j}). There are 16 incoming edges to all three receivers, 44 triplets (S_i, R_j, e_{R_j}), and 377 paths from the sources to the receivers in total; this leads to an average of nine paths and a maximum of 25 paths per triplet (S_i, R_j, e_{R_j}). We visit all nodes in a random order and we assign coefficients from a finite field with increasing size $(2^{10} - 2^{18})$.

In Fig. 18, we show the probability of success in terms of path identifiability for five such triplets (S_i, R_j, e_{R_j}), with 7, 9, 13, 20, and 25 number of paths, respectively. The values are averaged over five different runs for each field size value. When we use random code selection over a field of size 2^{16} or larger, we get good results: for a field of size 2^{18} or larger, we get almost 100% success for all triplets. These are good results for a large realistic topology such as Exodus, since almost 100% success is achieved with much less bits than the 1500 bytes of an IP

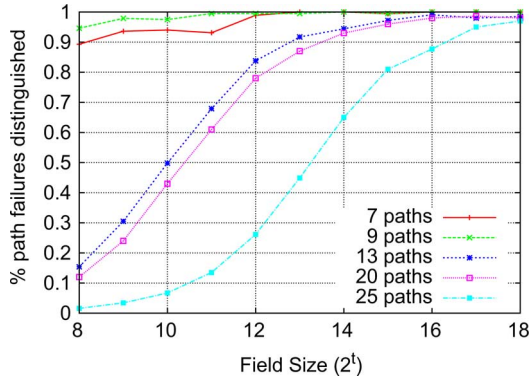


Fig. 18. Random code design for the Exodus topology. The X-axis shows the field size over which we choose the coding coefficients randomly: finite fields with different sizes $(F_{2^8} - F_{2^{18}})$. The Y-axis shows the effect on path identifiability (probability of success, defined as the % of the paths in a triplet (S_i, R_j, e_{R_j}) that we can uniquely distinguish from the observed outcome).

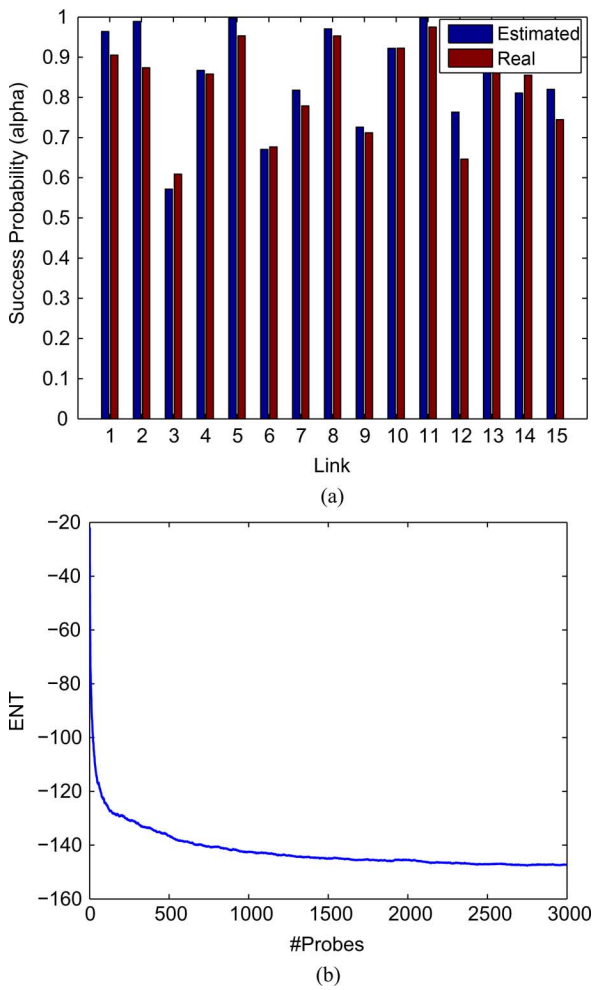


Fig. 19. (Modified) Abilene topology. Loss rates ($\overline{\alpha}$'s) are different across links: they are assigned inversely proportional to the bandwidth of the actual links, as reported in [30]. The resulting average loss rate is 17%. (a) Estimated versus real success rate (for 3000 probes). (b) ENT metric versus number of probes.

packet. Random assignment of coefficients over a set of prime numbers leads to success probability above 98% when we use up to prime 907 and field size 2^{18} for the linear operations.

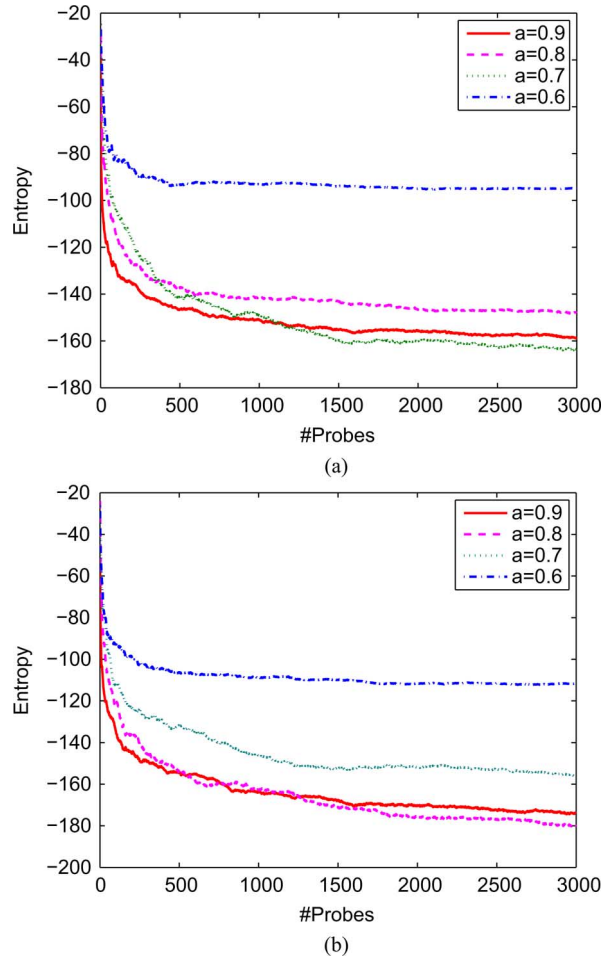


Fig. 20. Abilene topology with the same α on all links. (a) One source: node 1. (b) Two sources: nodes 1 and 9.

4) Results on BP Inference: This section presents results on the quality of the BP estimation for different assignments of loss rates to the links of the two considered topologies.

In Fig. 19, we consider the Abilene topology with loss rates inversely proportional to the bandwidth of the actual links; the intuition for this assignment is that links with high bandwidth are less likely to be congested. We see that the estimation error for each link (MSE) and for all links ENT decreases quickly. In Fig. 20, the same topology is considered, but with the same α on all links: again, ENT decreases with the number of probes; as expected, the larger the $\overline{\alpha}$, the slower the convergence; there is not a big difference between having one or two sources in this case. Fig. 21 shows the estimation error ENT for the Exodus topology with uniform loss rates. Finally, Table IV shows the results for different numbers and placements of sources in the (modified) Abilene topology. Unlike Fig. 20, Table IV shows that the choice of sources matters and that increasing the number of sources helps in decreasing the ENT.

5) NC-Tomography Versus Multicast Tomography: We finally compare the network coding approach to traditional multicast tomography for general topologies [3]. In the traditional approach, multiple MTs are used to cover the general topology, and the estimates from different trees are combined into one, using approaches in [3].

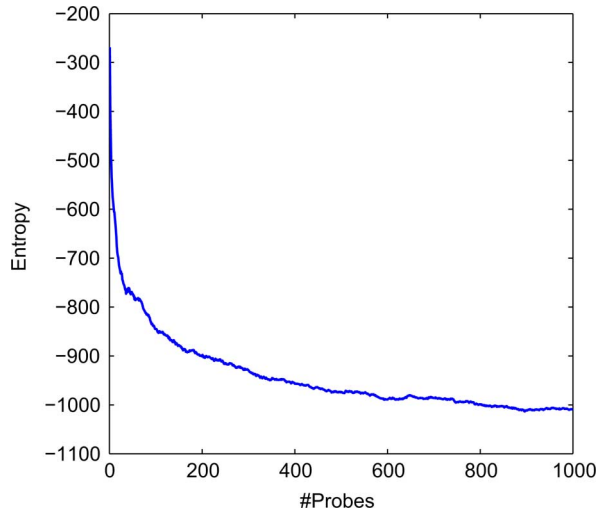


Fig. 21. Exodus topology, considering different loss rates across links: uniformly in $[1\%, 35\%]$.

TABLE IV
QUALITY OF ESTIMATION FOR THE (MODIFIED) ABILENE TOPOLOGY AND FOR DIFFERENT CHOICES OF SOURCE(S)

Srcs-Rcvs	Entropy for loss rate same over all links					
	$\overline{\alpha}$=0.05	$\overline{\alpha}$=0.1	$\overline{\alpha}$=0.15	$\overline{\alpha}$=0.2	$\overline{\alpha}$=0.25	$\overline{\alpha}$=0.3
{1}-{9}	-178.6	-158.8	-147.9	-147.7	-161.6	-163.5
{5}-{6}	-178.1	-158.3	-149.6	-154.5	-160.4	-156.5
{9}-{2}	-176.1	-163.3	-155.8	-161.2	-166.6	-151.7
{1,9}-{7}	-189.3	-173.9	-166.5	-180.3	-171.7	-156.2
{3,6}-{9}	-186.2	-176.2	-171.3	-177.8	-166.7	-151.4
{9,6}-{4}	-186.9	-174.1	-169.5	-178.7	-173.2	-165.4
{1,5,9}-{7}	-199.8	-190.6	-180.9	-184.4	-172.3	-166.9
{1,4,10}-{9}	-186.4	-183.9	-178.3	-182.3	-177.3	-173.2

Fig. 22(a) shows the topology we used in the comparison, which is taken from [3]: Nodes $\{0, 1, 2, 5\}$ are sources, nodes $\{12, \ldots, 19\}$ are receivers, and all remaining nodes (shown as boxes) are intermediate nodes. When the traditional approach is used, probes are sent from each of the four sources to all receivers using an MT, an estimate is computed from every tree, and then, the four estimates are combined into one using the minimum variance weighted average [3]. When the network coding approach is used, the same four sources and the same receivers are used, but probes are combined at intermediate nodes $\{6, 7\}$. For a fair comparison, the same BP algorithm has been used for estimation over MTs and using the network coding approach. Fig. 22(b) shows the performance of both schemes. We see that the network coding approach achieves a better error versus number of probes tradeoff. The main benefit in this case comes from the fact that the network coding approach eliminates the overlap of the MTs below nodes 6 and 7.

There is of course a wealth of other tomographic techniques that are not simulated here. (For example, we could cover a general graph with unicast probes, but this would perform worse than using multicast probes.) The reason is that [3] is directly comparable to our approach and thus highlights the intuitive benefits of network coding, everything else being equal. Network coding ideas could also be developed for and combined with other tomographic approaches.

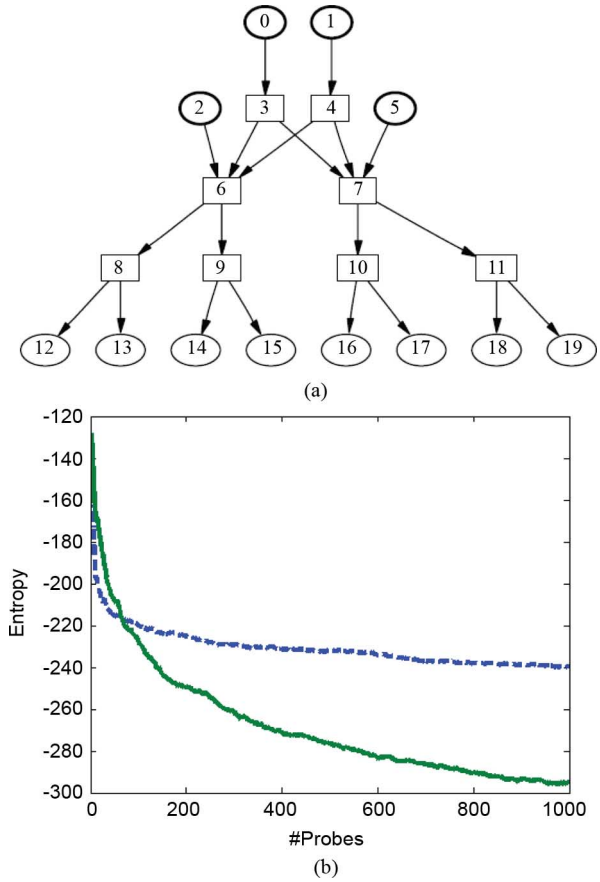


Fig. 22. Comparison of the network coding approach to traditional tomography. In both cases, the same sources and receivers are used. In the traditional case, four MTs are used and the estimates are combined using methods from [3]. In the network coding case, probes are combined wherever they meet in the network (nodes 6 and 7). (a) Simulation topology from [3]. Nodes $\{0, 1, 2, 5\}$ are sources, nodes $\{12, \ldots, 19\}$ are receivers, and all remaining nodes (shown as boxes) are intermediate nodes. (b) Performance of tomography: error (ENT) versus number of probes. Solid and dashed lines correspond to the network coding approach and the traditional approach, respectively. All links have loss rate $\overline{\alpha} = 0.04$.

VII. CONCLUSION

In this paper, we revisited the well studied and hard problem of link loss tomography using new techniques in networks equipped with network coding capabilities. We developed a novel framework for estimating the loss rates of some or all links in this setting. We considered trees and general topologies. We showed that network coding capabilities can improve virtually all aspects of loss tomography, including identifiability, routing complexity, and the tradeoff between estimation accuracy and bandwidth overhead.

APPENDIX A
PROOFS OF THEOREMS

A) Proof of Theorem 4.1:

Proof: To prove that conditions 1 and 2 are necessary, consider that condition 1 is not satisfied. Then, C can only receive one stream of probe packets, since it is connected to only one source. There exists an edge e through which this stream of

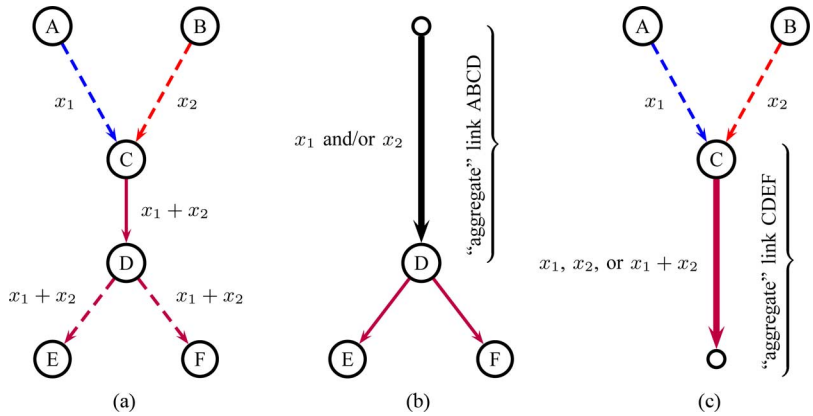


Fig. 23. *Reductions.* (a) Real topology based on conditions 1(b) and 2(b). The goal is to identify the loss rate of link CD. A, B are sources and E, F are receivers. AC, BC, DE, DF can be either links or paths from/to the sources/receivers. (b) Reduce the real topology to an MT with three links: "aggregate" link $ABCD$ (which transmits *some* symbol, x_1, x_2 or $x_1 \oplus x_2$, below D), and links DE, DF (which broadcast that symbol). (c) Reduce the real topology to an RMT with three links: AC, BC, and "aggregate" link $CDEF$ (which transmits the symbol coming in CD to at least one receiver). As shown in detail in Table II, the observations in the reduced topologies are simply unions of disjoint observations in the original topology, and their probabilities are the sum of the probabilities of the corresponding observations in the original topology.

probe packets arrives at node C. The link success rate associated with link CD cannot be distinguished from the link success rate associated with link e. More formally, if α_e is the success probability associated with link e and α_{CD} is the success probability associated with link CD, then the variables α_e and α_{CD} appear always together (*e.g.*, in the expression $1 - \alpha_e\alpha_{CD}$ in the probability function P_α). Therefore, there are many pairs of values (α_e, α_{CD}) that lead to the same P_α. According to definition 2, this means that link CD is not identifiable. Similar arguments hold for the other conditions and this completes the forward argument.

Next, we prove that conditions 1 and 2 are sufficient for identifying link CD.

First, let us consider Case 1, where Conditions 1(b) and 2(b) are satisfied. The remaining cases are similar and are discussed at the end of this proof. These conditions mean that the paths involving link CD should be as depicted in Fig. 23(a): AC, BC, DE, DF can be either links or paths from/to the sources/receivers, respectively. In the latter case (when AC, BC and DE, DF depict paths), the path success probability can be computed from the success rates of the corresponding links. Essentially, Case 1 (also shown in Fig. 2– 5-links, Case 1) generalizes the motivating example of Section IV, where the links AC, BC, DE, DF are replaced by paths AC, BC, DE, DF with the same success probability.

In Definition 2, and consistently with [2], we defined the links as identifiable iff the probability distribution P_α uniquely determines the parameters α,[21] *i.e.*, iff for $\alpha, \alpha' \in (0,1]^{|E|}$, $P_\alpha = P_{\alpha'}$ implies $\alpha = \alpha'$. To establish the identifiability of link CD, we repeatedly apply the identifiability result for a three-link MT (from [2]) and for an RMT (leveraging the reversibility property in Theorem 5.1, Section V-B2). Consider the two reductions of the actual five-link topology (as described in Section V-B3), to an MT shown in Fig. 23(b), and to an RMT shown in Fig. 23(c), respectively.

[21]Recall that α refers to the vector of all success probabilities, and α_e refers to the success probability of one particular edge e.

In the case of the three-link MT consisting of $ABCD$ and DE, DF, Theorems 2 and 3 in [2] guarantee that α_{DE}, α_{DF}, and α_{ABCD} are identifiable. Namely, $P^m_{\alpha'} = P^m_\alpha$ implies $\alpha'^m = \alpha^m$.

On the other hand, since the MLE for the RMT has the same functional form as the MT (as described in Section V-B2), using again the main result of [2], we have that $P^r_{\alpha'} = P^r_\alpha$ implies $\alpha'^r = \alpha^r$.

Proving identifiability in the original topology, via contradiction: Consider the five-link tree in Fig. 23(a), and assume that there exist $\alpha, \alpha' \in (0,1]^{|E|}$ for which $P_\alpha = P_{\alpha'}$ and $\alpha \neq \alpha'$.

Use the MT reduction to map the success rates α to α^m and associated probabilities P_α to P^m_α. Similarly, reduce the success rates α' to α'^m, and associated probabilities $P_{\alpha'}$ to $P^m_{\alpha'}$. Since $P_\alpha = P_{\alpha'}$, we conclude that $P^m_\alpha = P^m_{\alpha'}$. Because the topology in Fig. 23(b) is identifiable [2], we conclude that $\alpha^m = \alpha'^m$. This implies that:

$$\alpha'_{DE} = \alpha'^m_{DE} = \alpha^m_{DE} = \alpha_{DE} \tag{30}$$

$$\alpha'_{DF} = \alpha'^m_{DF} = \alpha^m_{DF} = \alpha_{DF} \tag{31}$$

$$\begin{aligned}(1-\overline{\alpha'}_{AC}\overline{\alpha'}_{BC})\alpha'_{CD} &= \alpha'^m_{ABCD} \\ &= \alpha^m_{ABCD} = (1-\overline{\alpha}_{AC}\overline{\alpha}_{BC})\alpha_{CD}.\end{aligned} \tag{32}$$

Applying similar arguments for the reduction to an RMT, we get that $\alpha^r = \alpha'^r$, and as a result:

$$\alpha'_{AC} = \alpha'^r_{AC} = \alpha^r_{AC} = \alpha_{AC} \tag{33}$$

$$\alpha'_{BC} = \alpha'^r_{BC} = \alpha^r_{BC} = \alpha_{BC} \tag{34}$$

$$\begin{aligned}(1-\overline{\alpha'}_{DE}\overline{\alpha'}_{DF})\alpha'_{CD} &= \alpha'^r_{CDEF} \\ &= \alpha^r_{CDEF} = (1-\overline{\alpha}_{DE}\overline{\alpha}_{DF})\alpha_{CD}.\end{aligned} \tag{35}$$

From (30)–(35), we conclude that $\alpha = \alpha'$, which is a contradiction. Therefore, $P_\alpha = P_{\alpha'}$ implies that $\alpha = \alpha'$, *i.e.*, identifiability.

TABLE V
CASE 2

Received at			Is link ok?				
B	E	F	AC	BC	CD	DE	DF
-	-	-	Multiple possible events				
-	-	x	1	0	1	0	1
-	x	-	1	0	1	1	0
-	x	x	1	0	1	1	1
x	-	-	1	1	0	*	*
x	-	-	1	1	1	0	0
x	-	x	1	1	1	0	1
x	x	-	1	1	1	1	0
x	x	x	1	1	1	1	1

The remaining cases (combinations of clauses (a), (b), (c) in Conditions 1 and 2, other than 1(b) and 2(b)) are shown in Fig. 2. For example, Condition 1(a) or 2(a) corresponds to the three-link MT or RMT, and the MINC MLE can then be used directly on these trees. Condition 1(c) or 2(c) leads to the Cases 2–4 in Fig. 2, and similar reductions as in Case 1 can be used to prove identifiability. This completes the proof. ■

B) Estimating α_{CD}:

Proof: Let us denote the outcomes in which link CD has worked by x_{CD}; the outcomes in which at least one of the upstream paths to C has worked by x_{up}; and the outcomes in which at least one of the downstream paths after D has worked by x_{dn}. For the intersection of any two of these outcomes, *e.g.*, x_{up} and x_{dn}, we use the notation $x_{\text{up,dn}}$. The independence of link loss rates indicates that x_{up}, x_{dn}, and x_{CD} are independent. Therefore:

$$\hat{\alpha}_{CD} = \hat{p}(x_{CD}) = \hat{p}(x_{CD}|x_{\text{up,dn}}) = \frac{\hat{p}(x_{CD}\,\&\,x_{\text{up,dn}})}{\hat{p}(x_{\text{up}})\hat{p}(x_{\text{dn}})}. \quad (36)$$

The numerator equals $1-\hat{p}([0,0,\ldots,0]) = \hat{\gamma}_C^r = \hat{\gamma}_D^m$. Also, we have that

$$\begin{aligned}\hat{p}(x_{\text{dn}}) &= \hat{p}(x_{\text{dn}}|x_{\text{up},CD}) = \hat{p}(x_{\text{dn}}|X_D \neq [0,\ldots,0]) \\ &= 1-\hat{p}(x_{\text{dn}}^c|X_D \neq [0,\ldots,0]) = 1-\prod_{j=1}^{Q}\overline{\beta}_{d(D)_j}^m. \quad (37)\end{aligned}$$

We can derive a similar expression for $\hat{p}(x_{\text{up}})$. Therefore:

$$\hat{\alpha}_{CD} = \frac{1-\hat{p}([0,0,\ldots,0])}{(1-\prod_{i=1}^{P}\overline{\beta}_{f(C)_i}^r)(1-\prod_{j=1}^{Q}\overline{\beta}_{d(D)_j}^m)}. \quad (38)$$

By writing (13) for $\overline{\beta}_D^m$ in Fig. 4, and by writing (18) for $\overline{\beta}_C^r$ in Fig. 4, we conclude that:

$$1-\prod_{j=1}^{Q}\overline{\beta}_{d(D)_j}^m = \frac{\beta_D^m}{\alpha_{\text{agg}}^m} = \frac{\gamma_D^m}{A_D^m},\ 1-\prod_{i=1}^{P}\overline{\beta}_{f(C)_i}^r = \frac{\beta_C^r}{\alpha_{\text{agg}}^r} = \frac{\gamma_C^r}{A_C^r}. \quad (39)$$

Equation (24) then follows from replacing these results into (38). ■

C) Proof of Lemma 5.5:

Proof: In [2], it has been shown that the likelihood function of the reduced MT in Fig. 4(a), $\mathcal{L}^m(\alpha^m)$, can be written as the sum of three distinct parts in which the derivative $\partial \log p^m(x^m)/\partial\alpha_k^m$ is constant. These parts are $\Omega^m(k)$, the $\Omega^m(f^i(k))\backslash\Omega^m(f^{i-1}(k))$, which we represent by Ω_2^m for simplicity, for $i=1,2,\ldots,l^m(k)$, and $(\Omega^m(0))^c$. The derivative in these parts is equal to $\frac{1}{\alpha_k^m}$, $\frac{1}{\overline{\beta}_{f^{i-1}(k)}^m}\frac{\partial\overline{\beta}_{f^{i-1}(k)}^m}{\partial\alpha_k^m}$, and $\frac{1}{\overline{\beta}_0^m}\frac{\partial\overline{\beta}_0^m}{\partial\alpha_k^m}$, respectively. Thus, the likelihood equation can be written as:

$$\begin{aligned}\frac{\partial\mathcal{L}^m}{\partial\alpha_k^m} &= \frac{1}{\alpha_k^m}\sum_{x^m\in\Omega^m(k)} n^m(x^m) \\ &+ \sum_{i=1}^{l^m(k)}\Big\{\frac{1}{\overline{\beta}_{f^{i-1}(k)}^m}\frac{\partial\overline{\beta}_{f^{i-1}(k)}^m}{\partial\alpha_k^m}\sum_{x^m\in\Omega_2^m} n^m(x^m)\Big\} \quad (40) \\ &+ \frac{1}{\overline{\beta}_0^m}\frac{\partial\overline{\beta}_0^m}{\partial\alpha_k^m}\sum_{x^m\in(\Omega^m(0))^c} n^m(x^m).\end{aligned}$$

Similarly, we can split the likelihood function of the original tree $\mathcal{L}(\alpha)$ into three parts in which $\partial\log p(x)/\partial\alpha_k$ is constant. These parts will be similar to those of an MT, only with $\Omega^m(k)$ as defined for the original tree in Section V-B2, and with $l^m(k)$ representing the number of ancestors of node k up to node C (instead of the root 0 in the MT). The derivative $\partial\log p(x)/\partial\alpha_k$ over these parts is also similar to the MT, *i.e.*, $\frac{1}{\alpha_k}$, $\frac{1}{\overline{\beta}_{f^{i-1}(k)}^m}\frac{\partial\overline{\beta}_{f^{i-1}(k)}^m}{\partial\alpha_k}$, and $\frac{1}{\overline{\beta}_C^m}\frac{\partial\overline{\beta}_C^m}{\partial\alpha_k}$, respectively. Therefore, we have that:

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\alpha_k} &= \frac{1}{\alpha_k}\sum_{x\in\Omega^m(k)} n(x) \\ &+ \sum_{i=1}^{l^m(k)}\Big\{\frac{1}{\overline{\beta}_{f^{i-1}(k)}^m}\frac{\partial\overline{\beta}_{f^{i-1}(k)}^m}{\partial\alpha_k}\sum_{x\in\Omega_2^m} n(x)\Big\} \quad (41) \\ &+ \frac{1}{\overline{\beta}_C^m}\frac{\partial\overline{\beta}_C^m}{\partial\alpha_k}\sum_{x\in(\Omega^m(C))^c} n(x)\end{aligned}$$

TABLE VI
CASE 3

Received at		Is link ok?				
B	F	AC	BC	CD	DE	DF
-	-	Multiple possible events				
-	x_1	1	0	1	0	1
-	x_2	1	0	0	1	1
-	x_2	0	*	*	1	1
-	$x_1 \oplus x_2$	1	0	1	1	1
x_1	-	1	1	0	0	1
x_1	-	1	1	*	*	0
x_1	x_1	1	1	1	0	1
x_2	x_2	1	1	0	1	1
x_1	$x_1 \oplus x_2$	1	1	1	1	1

TABLE VII
CASE 4

Received at	Is link ok?				
F	AC	BC	CD	DE	DF
-	Multiple possible events				
x_1	1	0	1	0	1
x_2	0	1	1	0	1
x_3	0	0	1	1	1
x_3	*	*	0	1	1
$x_1 \oplus x_2$	1	1	1	0	1
$x_1 \oplus x_3$	1	0	1	1	1
$x_2 \oplus x_3$	0	1	1	1	1
$x_1 \oplus x_2 \oplus x_3$	1	1	1	1	1

1) $\hat{\alpha}_k^m$ versus $\hat{\alpha}_k$, $k < D$: We first compare the solutions $\hat{\alpha}_k^m$ of (40) and $\hat{\alpha}_k$ of (41) for $k < D$. From (21), we have:

$$\sum_{x \in \Omega^m(k)} n(x) = \sum_{x^m \in \Omega^m(k)} n^m(x^m) \tag{42}$$

$$\sum_{x \in \Omega_2^m} n(x) = \sum_{x^m \in \Omega_2^m} n^m(x^m) \tag{43}$$

$$\sum_{x \in (\Omega^m(C))^c} n(x) = \sum_{x^m \in (\Omega^m(0))^c} n^m(x^m). \tag{44}$$

Therefore, for any link k located below node D, we have that:

$$\frac{\partial \mathcal{L}^m}{\partial \alpha_k^m} = \frac{\partial \mathcal{L}}{\partial \alpha_k} \Longrightarrow \hat{\alpha}_k^m = \hat{\alpha}_k, \quad k < D \tag{45}$$

2) $\hat{\alpha}_{\text{agg}}^m$ versus $\hat{\alpha}_{CD}$: For α_{agg}^m and α_{CD}, (40) and (41) consist of only the first and the last terms. We have that:

$$\frac{\partial \mathcal{L}^m}{\partial \alpha_{\text{agg}}^m} = \frac{1}{\alpha_{\text{agg}}^m} \sum_{\Omega^m(D)} n^m(x^m) + \frac{1}{\overline{\beta}_0^m} \frac{\partial \overline{\beta}_0^m}{\partial \alpha_{\text{agg}}^m} \sum_{(\Omega^m(0))^c} n^m(x^m) \tag{46}$$

$$\frac{\partial \mathcal{L}}{\partial \alpha_{CD}} = \frac{1}{\alpha_{CD}} \sum_{x \in \Omega^m(D)} n(x) + \frac{1}{\overline{\beta}_C^m} \frac{\partial \overline{\beta}_C^m}{\partial \alpha_{CD}} \sum_{x \in (\Omega^m(C))^c} n(x). \tag{47}$$

Thus, $\frac{\partial \mathcal{L}^m}{\partial \alpha_{\text{agg}}^m} \neq \frac{\partial \mathcal{L}}{\partial \alpha_{CD}}$, but the definition of $\overline{\beta}_k^m$ indicates that:

$$\overline{\beta}_0^m = 1 - \alpha_{\text{agg}}^m (1 - \prod_{j=1}^{Q} \overline{\beta}_{d(D)_j}^m) \tag{48}$$

$$\overline{\beta}_C^m = 1 - (1 - \prod_{i=1}^{P} \overline{\beta}_{f(C)_i}^r) \alpha_{CD} (1 - \prod_{j=1}^{Q} \overline{\beta}_{d(D)_j}^m). \tag{49}$$

From (42), (44), (48), and (49), we find out that the solutions $\hat{\alpha}_{\text{agg}}^m$ of (46) and $\hat{\alpha}_{CD}$ of (47) are related via:

$$\hat{\alpha}_{CD} = \frac{\hat{\alpha}_{\text{agg}}^m}{1 - \prod_{i=1}^{P} \overline{\beta}_{f(C)_i}^r}. \tag{50}$$

■

Note: The proof of Lemma 5.6 is similar to the proof of Lemma 5.5 above.

D) Proof of Theorem 5.4:

Proof: In [2], it has been shown that $\hat{\alpha}_k^m$ in (12) are the MLE of the MT. Therefore, $\hat{\alpha}_k = \hat{\alpha}_k^m$, $k < D$, are also the MLE of the corresponding links in the original tree. In addition, by following the same approach as in [2] and due to the reversibility property, one can show that $\hat{\alpha}_k = \hat{\alpha}_k^r$, $k > C$, are also the MLE of the corresponding links in the original tree. For $\hat{\alpha}_{CD}$, since $\hat{\alpha}_{\text{agg}}^m = \hat{A}_D^m$ and using (39), one can obtain (24) from (50). Therefore, (24) is a solution of $\frac{\partial \mathcal{L}}{\partial \alpha_{CD}} = 0$. Furthermore, from (47) and (49), we have that:

$$\frac{\partial^2 \mathcal{L}}{\partial \alpha_{CD}^2} = \frac{-1}{\alpha_{CD}^2} \sum_{x \in \Omega^m(D)} n(x) - \frac{1}{\overline{\beta}_C^{m\,2}} \left(\frac{\partial \overline{\beta}_C^m}{\partial \alpha_{CD}}\right)^2 \sum_{x \in (\Omega^m(C))^c} n(x). \tag{51}$$

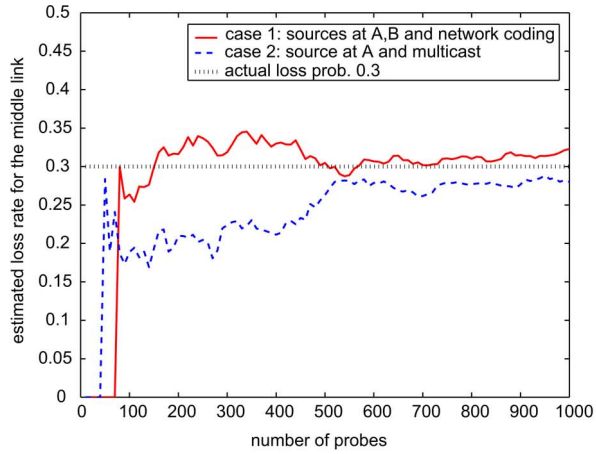
This is always negative. Therefore, $\mathcal{L}$ is concave in α_{CD} and (24) is the unique solution of the likelihood equation. This solution is also in the desired range $(0, 1]$, because from (38), we have that:

$$\hat{\alpha}_{CD} > 0 \Longleftrightarrow \hat{p}([0, 0, \ldots, 0]) < 1$$

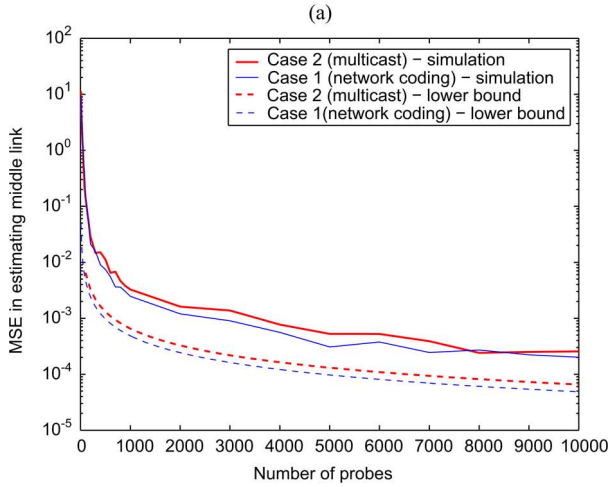
i.e., not all packets are lost, which is the default assumption in tomography: no inference can be made without data. Also:

$$\hat{\alpha}_{CD} < 1 \Longleftrightarrow 1 - \hat{p}([0, \ldots, 0]) < (1 - \prod_{i=1}^{P} \overline{\beta}_{f(C)_i}^r)(1 - \prod_{j=1}^{Q} \overline{\beta}_{d(D)_j}^m).$$

This is asymptotically true for $\alpha_{CD} > 0$, because as $n \to \infty$, the percentage of packets that are *not* lost approaches the probability $(1 - \prod_{i=1}^{P} \overline{\beta}_{f(C)_i}^r) \alpha_{CD} (1 - \prod_{j=1}^{Q} \overline{\beta}_{d(D)_j}^m)$, which is $< (1 - \prod_{i=1}^{P} \overline{\beta}_{f(C)_i}^r)(1 - \prod_{j=1}^{Q} \overline{\beta}_{d(D)_j}^m)$. Therefore, (24) is the MLE of α_{CD} in the original tree. ■



(a)



(b)

Fig. 24. Convergence of the ML estimator for cases 1 and 2, (a) Estimator versus number of probes. (b) Estimation variance versus number of probes.

We now provide additional details and simulation results on the effect of the number and location of sources.

Appendix B
Effect of the Number and Location of Sources

A) Various Configurations for the Five-Link Topology: Let us consider again the four cases shown in Fig. 2 for the basic five-link topology. The first case, also shown in Fig. 1, has been

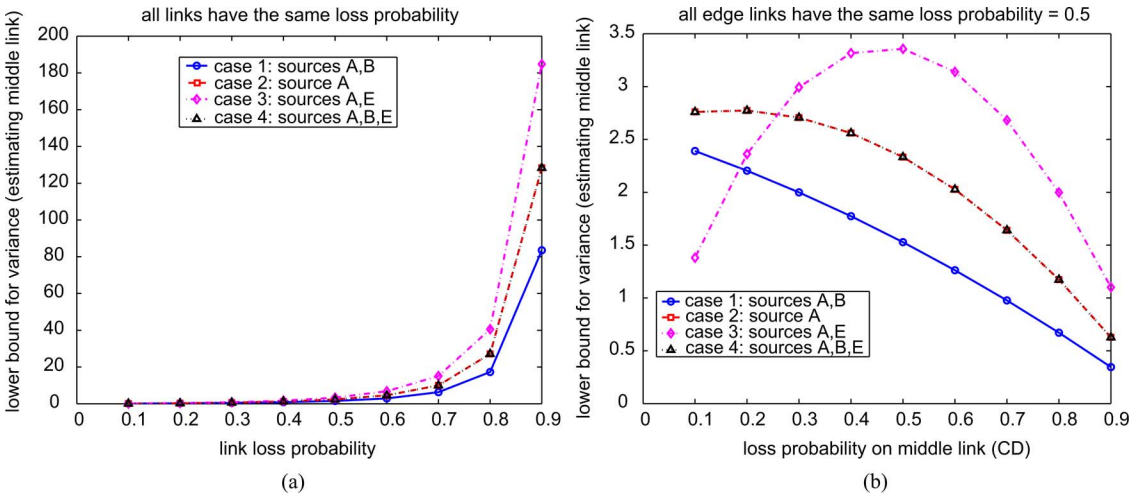


Fig. 25. Comparing the four cases in Fig. 2 in terms of the lower bound of variance. (a) All links have the same $\overline{\alpha}$. (b) All edge links have the same $\overline{\alpha}_{\text{edge}} = 0.5$.

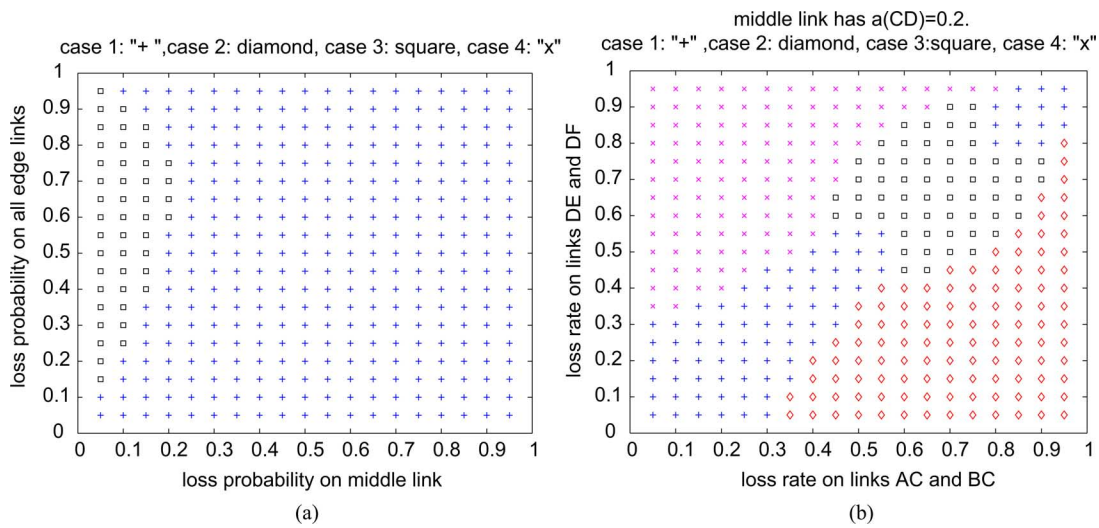


Fig. 26. Possible combinations of loss rates on all the five links. We indicate which Case (among the four) performs better (has the lowest Cramer–Rao bound). (a) All edge links have the same $\overline{\alpha}_{\text{edge}}$. Consider all possible combinations of $(\overline{\alpha}_{\text{edge}}, \overline{\alpha}_{\text{middle}})$. (b) $\overline{\alpha}_{AC} = \overline{\alpha}_{BC} = \overline{\alpha}_s, \overline{\alpha}_{DE} = \overline{\alpha}_{DF} = \overline{\alpha}_r, \overline{\alpha}_{CD} = 0.8$. Consider all combinations of $(\overline{\alpha}_s, \overline{\alpha}_r)$.

discussed in length in Table I and in Section IV. The corresponding tables used for estimation in Cases 2–4 of Fig. 2 are shown for completeness in Tables V–VII.

B) Simulation Results for the Five-Link Topology: Consider again the basic five-link topology of Fig. 2 and focus on estimating the middle link *CD*. Here, we show that, even though with network coding links are identifiable for all four cases, the estimation accuracy differs.

In Fig. 24, we assume that all five links have $\overline{\alpha} = 0.3$ and we look at the convergence of the MLE versus number of probes for *Case 1* (using network coding) and for *Case 2* (multicast probes with source A). Fig. 24(a) shows the estimated value (for one loss realization). Both estimators converge to the true value, with the network coding being only slightly faster in this scenario.

In Fig. 24(b), we plot the MSE of the MLE for *Case 1* (using network coding) and for *Case 2* (multicast) across number of probes. For comparison, we have also plotted the Cramer–Rao bound for link CD, which is consistent with the simulation results. For this scenario, *Case 1* does slightly better than *Case 2*, but not by a significant amount. This motivated us to exhaustively compare all four cases in Fig. 2, for all combinations of loss rates on the five links.

Fig. 25 plots the Cramer–Rao bound for the four cases as a function of the link-loss probability on the middle link. The left plot assumes that $\overline{\alpha}$ is the same for all five links, while the right plot looks at the case where the edge links have a fixed loss rate equal to 0.5. We observe that *Case 1* shows to achieve a lower MSE bound. Interestingly, the curves for *Cases 2* (multicast) and *Case 4* (reverse multicast) coincide. The difference between the performance of different cases is more evident in the right plot [see Fig. 25(b)].

In Fig. 26, we systematically consider possible combinations of loss rates on the five links, and we show which case estimates better the middle link. In the left figure, we assume that all edge links have the same loss rate and we observe that for most combinations of $(\overline{\alpha}_{\text{middle}}, \overline{\alpha}_{\text{edge}})$, *Case 1* (shown in "+") performs better. In the right plot, we assume that the middle link is fixed

at $\overline{\alpha}_{CD} = 0.8$ and that $\overline{\alpha}_{AC} = \overline{\alpha}_{BC} = \overline{\alpha}_s, \overline{\alpha}_{DE} = \overline{\alpha}_{DF} = \overline{\alpha}_r$. Considering all combinations $(\overline{\alpha}_s, \overline{\alpha}_r)$, each one of the four cases dominates for some scenarios. An interesting observation is, again, the symmetry between *Case 2* (multicast) and *Case 4* (reverse multicast).

ACKNOWLEDGMENT

The authors would like to thank Suhas Diggavi and Ramya Srinivasan for interactions on the problem of source selection.

REFERENCES

[1] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, "Network tomography: Recent developments," *Statist. Sci.*, vol. 19, no. 3, pp. 499–517, 2004.

[2] R. Caceres, N. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network-internal loss characteristics," *IEEE Trans. Inf. Theory*, vol. 45, no. 7, pp. 2462–2480, Nov. 1999.

[3] T. Bu, N. Duffield, F. Presti, and D. Towsley, "Network tomography on general topologies," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 30, no. 1, pp. 21–30, 2002.

[4] M. Rabbat, R. Nowak, and M. Coates, "Multiple source multiple destination network tomography," in *Proc. IEEE 23rd Annu./Joint Conf. IEEE Comput. Commun. Soc.*, Hong Kong, 2004, pp. 1628–1639.

[5] N. Duffield, F. Presti, V. Paxson, and D. Towsley, "Inferring link loss using striped unicast probes," in *Proc. IEEE 20th Annu./Joint Conf. IEEE Comput. Commun. Soc.*, New York, NY, USA, 2001, pp. 915–923.

[6] M. Coates and R. Nowak, "Network loss inference using unicast end-to-end measurement," presented at the ITC Conf. IP Traffic, Model. Manage., Monterey, CA, USA, 2000.

[7] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[8] S. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[9] The Network Coding Webpage [Online]. Available: http://www.netcod.org

[10] Y. Vardi, "Network tomography: Estimating source-destination traffic intensities from link data," *J. Amer. Statist. Assoc.*, vol. 91, no. 433, pp. 365–377, 1996.

[11] F. Presti, N. Duffield, J. Horowitz, and D. Towsley, "Multicast-based inference of network internal delay distributions," *IEEE/ACM Trans. Netw.*, vol. 10, no. 6, pp. 761–775, Dec. 2002.

[12] E. Lawrence, G. Michailidis, and V. Nair, "Statistical inverse problems in active network tomography," *IMS Lecture Notes—Monograph Series*, vol. 54, pp. 24–44, 2007.

[13] Y. Tsang, M. Coates, and R. Nowak, "Passive network tomography using EM algorithms," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, New York, NY, USA, 2001, vol. 3, pp. 1469–1472.

[14] V. Padmanabhan, L. Qiu, and H. Wang, "Passive network tomography using Bayesian inference," in *Proc. 2nd ACM SIGCOMM Workshop Internet Meas.*, New York, NY, USA, 2002, pp. 93–94.

[15] C. Fragouli and E. Soljanin, "Monograph on network coding: Fundamentals and applications," in *Foundations and Trends in Networking*. Delft, The Netherlands: Now, 2007, vol. 2.

[16] T. Ho, B. Leong, Y. Chang, Y. Wen, and R. Koetter, "Network monitoring in multicast networks using network coding," in *Proc. Int. Symp. Inf. Theory*, 2005, pp. 1977–1981.

[17] G. Sharma, S. Jaggi, and B. Dey, "Network tomography via network coding," in *Proc. Inf. Theory Appl. Workshop*, San Diego, CA, USA, 2008, pp. 151–157.

[18] H. Yao, S. Jaggi, and M. Chen, Passive network tomography for erroneous networks: A network coding approach 2010 [Online]. Available: arXiv:0908.0711

[19] M. Jafarisiavoshani, C. Fragouli, S. Diggavi, and C. Gkantsidis, "Bottleneck discovery and overlay management in network coded peer-to-peer systems," in *Proc. SIGCOMM Internet Netw. Manag. Workshop*, Tokyo, Japan, 2007, pp. 293–298.

[20] C. Fragouli and A. Markopoulou, "A network coding approach to overlay network monitoring," presented at the 43rd Allerton Conf. Commun., Control, Comput., Monticello, IL, 2005.

[21] C. Fragouli, A. Markopoulou, R. Srinivasan, and S. Diggavi, "Network monitoring: It depends on your points of view," presented at the Inf. Theory Appl. Workshop, San Diego, CA, 2007.

[22] M. Gjoka, C. Fragouli, P. Sattari, and A. Markopoulou, "Loss tomography in general topologies with network coding," in *Proc. IEEE Global Telecommun. Conf.*, Washington, DC, Nov. 2007, pp. 381–386.

[23] P. Sattari, A. Markopoulou, and C. Fragouli, "Maximum likelihood estimation for multiple-source loss tomography with network coding," in *IEEE Int. Symp. Netw. Coding*, Beijing, China, Jul. 2011.

[24] J. Ebrahimi and C. Fragouli, "Vector network coding algorithms," in *Proc. IEEE Int. Symp. Inf. Theory*, Austin, TX, Jun. 2010, pp. 2408–2412.

[25] T. Cover and J. Thomas, *Elements of Information Theory*. New York, NY, USA: Wiley, 2006.

[26] E. Lehmann, *Elements of Large-Sample Theory*. New York, NY, USA: Springer-Verlag, 1999.

[27] P. Sattari, A. Markopoulou, C. Fragouli, and M. Gjoka, A network coding approach to loss tomography 2012 [Online]. Available: arXiv:1005.4769

[28] J. A. Rice, *Mathematical Statistics and Data Analysis*. Belmont, CA: Duxbury Press, 1995.

[29] Y. Mao, F. Kschischang, B. Li, and S. Pasupathy, "A factor graph approach to link loss monitoring in wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 820–829, Sep. 2006.

[30] The Abilene Research Network [Online]. Available: http://abilene.internet2.edu

[31] M. Adler, T. Bu, R. Sitaraman, and D. Towsley, "Tree layout for internal network characterizations in multicast networks," *Proc. Int. Workshop Netw. Group Commun.*, pp. 189–204, 2001.

[32] Z. Li, B. Li, D. Jiang, and L. Lau, "On achieving optimal throughput with network coding," in *Proc. 24th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, Miami, FL, USA, 2005, pp. 2184–2194.

[33] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[34] J. Schwartz, "Fast probabilistic algorithms for verification of polynomial identities," *J. ACM*, vol. 27, no. 4, pp. 717–717, 1980.

[35] N. Harvey, "Deterministic network coding by matrix completion," M.S. Thesis, Dept. Electr. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2005.

[36] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[37] M. Yazdani, S. Hemati, and A. Banihashemi, "Improving belief propagation on graphs with cycles," *IEEE Commun. Lett.*, vol. 8, no. 1, pp. 57–59, Jan. 2004.

[38] M. Fossorier, "Iterative reliability-based decoding of low-density parity check codes," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 5, pp. 908–917, May 2001.

[39] Y. Mao and A. Banihashemi, "Decoding low-density parity-check codes with probabilistic schedule," *IEEE Commun. Lett.*, vol. 5, no. 10, pp. 414–416, Oct. 2001.

[40] Rocketfuel: an ISP Mapping Engine [Online]. Available: http://www.cs.washington.edu/research/networking/rocketfuel

Pegah Sattari (S'08) received the B.S. degree in Electrical Engineering from Sharif University of Technology, Tehran, Iran, in 2006, and the M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of California, Irvine, in 2007 and 2012, respectively. She is currently a senior software engineer at Jeda Networks Inc. Her research interests include network measurement and analysis, network coding, and network tomography/inference problems.

Athina Markopoulou (S'98–M'02–SM'12) is an Associate Professor in the EECS Department at the University of California, Irvine. She received the Diploma degree in Electrical and Computer Engineering from the National Technical University of Athens, Greece, in 1996, and the M.S. and Ph.D. degrees in Electrical Engineering from Stanford University in 1998 and 2003, respectively. She has been a postdoctoral fellow at Sprint Labs and at Stanford University, and a member of the technical staff at Arastra Inc. Her research interests include network coding, network measurement and security, media streaming and online social networks. She received the NSF CAREER award in 2008.

Christina Fragouli (M'00) is an Associate Professor in the School of Computer and Communication Sciences, EPFL, Switzerland. She received the B.S. degree in Electrical Engineering from the National Technical University of Athens, Greece, in 1996, and the M.Sc. and Ph.D. degrees in Electrical Engineering from the University of California, Los Angeles, in 1998 and 2000, respectively. She has worked at the Information Sciences Center, AT&T Labs, and the National University of Athens. She has also visited Bell Labs and DIMACS, Rutgers University. Her research interests include network coding, network information flow theory and algorithms, and connections between communications and computer science. She received the ERC Starting Grant from the European Research Council in 2009.

Minas Gjoka received his B.S. (2005) degree in Computer Science at the Athens University of Economics and Business, Greece, and his M.S. (2008) and Ph.D. (2010) degrees in Networked Systems at the University of California, Irvine. He is currently a postdoc at the University of California, Irvine. His research interests are in the general areas of networking and distributed systems, with emphasis on online social networks, peer-to-peer systems, and network measurement.