Accurate and Efficient Network Tomography Through Network Coding

Jiaqi Gui, Vahid Shah-Mansouri, *Student Member, IEEE*, and Vincent W. S. Wong, *Senior Member, IEEE*

***Abstract*—Accurate and efficient measurement of network-internal characteristics is critical for the management and maintenance of large-scale networks. In this paper, we propose a *linear algebraic network tomography* (LANT) framework for the active inference of link loss rates on mesh topologies through network coding. Probe packets are transmitted from the sources to the destinations along a set of paths. Intermediate nodes linearly combine the received probes and transmit the coded probes using predetermined coding coefficients. Although a smaller probe size can reduce the bandwidth usage of the network, the inference framework is not valid if the probe size falls below a certain threshold. To this end, we determine the minimum probe packet size, which is necessary and sufficient to establish the mapping between the contents of the received probes and the losses on the different sets of paths. Then, we develop algorithms to find the coding coefficients such that the minimum probe size is achieved. We propose a linear algebraic approach to develop consistent estimators of link loss rates, which converge to the actual loss rates as the number of probes increases. Simulation results show that the LANT framework achieves better estimation accuracy than the belief propagation algorithm for a large number of probe packets.**

***Index Terms*—Link loss rate estimation, network coding, network tomography.**

I. Introduction

ACCURATE and efficient measurement of network-internal characteristics is critical for the management and maintenance of large-scale networks. However, the traditional approach for characterizing network performance requires access to a wide range of routers to obtain link-level statistics. The routers are operated by different companies or service providers, which make it difficult to collect detailed information at individual devices. Network tomography, which was first coined in [1], collects and analyzes end-to-end measurements to infer link loss rate, delay [2], [3], and network topology [4], [5]. It can be performed either in an *active* or *passive* manner [6]. Active network tomography refers to the case where probe packets are sent from the sources to the receivers located on the periphery of the network [7]–[9]. It provides more informative and reliable path-level measurements, at the cost of utilizing additional network resources such as bandwidth and energy. On the contrary, passive network tomography reveals information from the existing data traffic so that it is more attractive for networks (e.g., wireless sensor networks) with limited power supply and bandwidth constraints [10]–[13].

In this paper, we consider the problem of link loss tomography on mesh topologies. Although there are extensive studies of link loss inference on multicast tree topologies [7], [14]–[16], loss tomography on mesh topologies is still a challenging problem. Bu *et al.* proposed an approach using multiple trees to cover a mesh topology and combine the inferred loss rates [17]. However, this approach may have *low bandwidth efficiency*, because the links that are part of multiple trees would be traversed by multiple probe packets in each time slot and thus create additional traffic. In addition, it may incur *high monitoring cost*, because it requires a large number of receivers to be deployed in each multicast tree.

Recent studies have shown that applying network coding [18], [19] in loss tomography can increase bandwidth efficiency [20]. In a network that can perform network coding in addition to multicast, the intermediate nodes linearly combine incoming probe packets and forward the coded probe packets to the outgoing links. Results in [21] show that, for active monitoring using network coding, an appropriate selection of the number and location of sources and receivers can affect the accuracy of estimation in general tree topologies. The work in [22] established a framework for loss tomography on mesh topologies. An orientation algorithm is proposed to find a directed acyclic graph from an undirected graph with selected sources. One example is illustrated such that each link in a mesh topology can be traversed in each time slot by exactly one probe. The work in [23] studied passive network tomography in the presence of network failures, under the setting of random linear network coding. Several sets of algorithms for topology estimation and failure detection are proposed under various settings of adversarial random failures.

To reduce monitoring cost, a set of end-to-end paths on mesh topologies only requires a limited number of sources and receivers. Nonetheless, existing approaches have not exploited the inherent information in the end-to-end observations. Thus, the linear system of link and path loss rates usually has a coefficient matrix with a deficient column rank,[1] which makes

Manuscript received May 7, 2010; revised December 5, 2010 and March 16, 2011; accepted March 28, 2011. Date of publication May 2, 2011; date of current version July 18, 2011. This work was supported by the Natural Sciences and Engineering Research Council of Canada. This paper was presented in part at the IEEE International Conference on Communications (ICC), Cape Town, South Africa, May 2010. The review of this paper was coordinated by Prof. B. Hamdaoui.

The authors are with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, V6T 1Z4, Canada (e-mail: jiaqig@ece.ubc.ca; vahids@ece.ubc.ca; vincentw@ece.ubc.ca).

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/TVT.2011.2149549

[1]The column rank of an $m \times n$ matrix is the maximum number of linearly independent columns of the matrix. If the matrix has rank n, then it has a full column rank; otherwise, the matrix has a deficient column rank.

it difficult to accurately infer the link loss rates [24]. A belief propagation (BP) algorithm for link loss inference in wireless sensor networks is proposed in [25] and is combined with the use of network coding in [22]. The approach in [26] first estimates the number of faulty links on a path and then uses the global information to estimate link loss rates.

In general, most of the previously proposed loss tomography approaches in the literature have one or more of the following performance bottlenecks:

1) low bandwidth efficiency;
2) high monitoring cost;
3) estimation of not being always accurate;
4) requiring additional assumptions.

In this paper, we propose a *linear algebraic network tomography* (LANT) framework for the active inference of link loss rates on mesh topologies. To increase bandwidth efficiency and reduce monitoring cost, we send probe packets along a set of end-to-end paths rather than multicast trees and apply network coding. To increase the estimation accuracy, we exploit the inherent correlation between the losses on the links and the losses on different sets of paths, which is captured through network coding. We refer to probe packets and network coding schemes jointly as *probe-coding schemes*. In our LANT framework, a valid probe-coding scheme enables us to establish the mapping between the contents of received probe packets and the losses on the different sets of paths. Using valid probe-coding schemes, we obtain valid end-to-end observations, based on which we can distinguish which paths have successfully transmitted a probe and which paths have not. We also define *link identifiability*, a link property that only depends on the network topology. For identifiable links, we develop consistent estimators that converge to the actual loss rates as the number of probes increases. Because the number of all path sets can exponentially grow as the total number of paths increases, we selectively monitor a subset of all path sets (the method of row selection), which are sufficient to infer the loss rates of all identifiable links. The contributions of this paper are summarized as follows.

- We determine the minimum probe packet size, which is necessary and sufficient for valid probe-coding schemes when network coding is applied. Then, we develop algorithms to find a valid probe-coding scheme such that the minimum probe packet size is achieved.
- We propose a linear algebraic (LA) approach to develop consistent estimators of link loss rates, which converge to the actual loss rates as the number of probes increases. We combine the methods of normal equations and row selection with the LA approach and analyze the computational complexity.
- We prove that the identifiability of a link, which only depends on the network topology, is a necessary and sufficient condition for the consistent estimation of its loss rate using the LANT framework.
- Simulation results show that the LA approach, using the method of row selection, can effectively decrease computational complexity without reducing estimation accuracy.

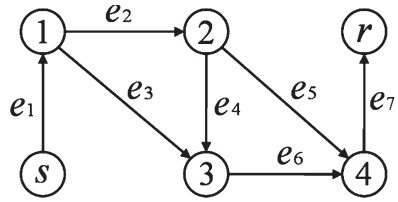


Fig. 1. Directed acyclic graph with $\mathcal{V} = \{s, r, 1, 2, 3, 4\}$ and $\mathcal{E} = \{e_1, e_2, \ldots, e_7\}$. The set of monitored end-to-end paths $\mathcal{P} = \{P_1, P_2, P_3\}$, where $P_1 = \{e_1, e_2, e_5, e_7\}$, $P_2 = \{e_1, e_2, e_4, e_6, e_7\}$, and $P_3 = \{e_1, e_3, e_6, e_7\}$. For link $e_2 = (1, 2)$, we have $\mathcal{P}(e_2) = \{P_1, P_2\}$.

The LA approach achieves better estimation accuracy than the BP algorithm when the estimators converge.

The framework that we present in this paper is unique compared to the prior work done in loss tomography using network coding. In terms of bandwidth efficiency, the work in [22] determines the necessary condition on probe size for valid probe-coding schemes, whereas the problem of finding coding coefficients remains unexplored. We find the minimum probe size and also develop an algorithm to find a valid probe-coding scheme such that the minimum probe size is achieved. In terms of inference approaches, the BP algorithm [25] only uses the information of the losses on different paths such that the estimation may not be accurate for networks with relatively high link loss rates. The work in [26] requires additional assumptions, e.g., *a priori* probability distribution function and the majority of links being lossless. In contrast, our LA approach needs no extra assumptions, whereas it can still obtain additional useful information, e.g., the losses on the different sets of paths. This information can only be obtained through probe-coding schemes and cannot be achieved by routing probes in general.

This paper is organized as follows. In Section II, we present the LANT framework. In Section III, we first determine the minimum probe size when network coding is applied. Then, we develop algorithms to find a valid probe-coding scheme with a minimal probe size. Section IV presents an LA approach for the consistent estimation of link loss rates. Simulation results are discussed in Section V. Conclusions are given in Section VI.

II. System Model and Framework

We model the network as a directed acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which consists of a set of nodes $\mathcal{V}$ and a set of links $\mathcal{E}$. The node set $\mathcal{V}$ includes routers and periphery devices where probe packets are sent and received. A link $e = (v, v') \in \mathcal{E}$ denotes a directed communication link from node v to node v'. Let $\mathcal{S}$ and $\mathcal{R}$ denote the set of source nodes and the set of receiver nodes, respectively. The set of monitored end-to-end paths is denoted by $\mathcal{P}$. A path $P \in \mathcal{P}$ is a set of directed links from a source to a receiver. Let $\mathcal{P}(e)$ denote the set of paths that include link e. We define a path–link matrix $\mathbf{M} = (m_{i,j})_{|\mathcal{P}| \times |\mathcal{E}|}$, whose $|\mathcal{P}|$ rows correspond to the $|\mathcal{P}|$ paths, and the $|\mathcal{E}|$ columns correspond to the $|\mathcal{E}|$ links as follows. The element $m_{i,j}$ is equal to 1 if the ith path in set $\mathcal{P}$ includes the jth link in set $\mathcal{E}$; otherwise, it is equal to 0. As an example, the directed acyclic graph in Fig. 1 has three paths from

source s to receiver r. Its path–link matrix is a 3×7 binary matrix, i.e.,

$$\mathbf{M} = \begin{array}{c} \\ P_1 \\ P_2 \\ P_3 \end{array} \begin{array}{c} \begin{array}{ccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array}. \quad (1)$$

Given a directed acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a set of monitored end-to-end paths $\mathcal{P}$, a link $e \in \mathcal{E}$ is called *identifiable* if, for each link pair (e, e'), where $e' \in \mathcal{E} \setminus \{e\}$, there exists at least one path in $\mathcal{P}$ that includes only one of the two links, i.e., $\mathcal{P}(e) \neq \mathcal{P}(e')$. As shown in Fig. 1, links $e_2, e_3, \ldots, e_6$ are identifiable links, whereas links e_1 and e_7 are nonidentifiable links, because $\mathcal{P}(e_1) = \mathcal{P}(e_7)$. We notice that the identifiability of a link depends only on the network topology.

The following proposition shows that the identifiability of a link is a necessary condition for the estimation of its loss rate.

Proposition 1: The loss rate of a link can be estimated *only if* the link is an identifiable link.

The proof of Proposition 1 is provided in Appendix A. We divide the set of nonidentifiable links into several groups, where each group contains a set of links that are included in the same set of paths. We refer to each group as a *virtual link*. As shown in Fig. 1, because $\mathcal{P}(e_1) = \mathcal{P}(e_7)$, we refer to e_1 and e_7 as one virtual link e_{v_1}. Let $\mathcal{E}_I$ and $\mathcal{E}_V$ denote the set of identifiable links and the set of virtual links, respectively. We have $\mathcal{E}_I = \{e_2, e_3, \ldots, e_6\}$ and $\mathcal{E}_V = \{e_{v_1}\}$. Note that $\mathcal{E}_I \cap \mathcal{E}_V = \varnothing$.

Thus, for each link $e \in \mathcal{E}_I \cup \mathcal{E}_V$, we have $\mathcal{P}(e) \neq \mathcal{P}(e')$ for all $e' \in \mathcal{E}_I \cup \mathcal{E}_V \setminus \{e\}$. We fix the order of elements in $\mathcal{E}_I \cup \mathcal{E}_V$. Accordingly, we define a type-1 modified path–link matrix $\overline{\mathbf{M}} = (\overline{m}_{i,j})_{|\mathcal{P}| \times |\mathcal{E}_I \cup \mathcal{E}_V|}$ as follows. The element $\overline{m}_{i,j}$ is equal to 1 if the ith path in set $\mathcal{P}$ includes the jth link in set $\mathcal{E}_I \cup \mathcal{E}_V$; otherwise, it is equal to 0. The type-1 modified path–link matrix for the graph in Fig. 1 is shown as

$$\overline{\mathbf{M}} = \begin{array}{c} \\ P_1 \\ P_2 \\ P_3 \end{array} \begin{array}{c} \begin{array}{cccccc} e_{v_1} & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array}. \quad (2)$$

We model the loss of packets on different links by a set of mutually independent Bernoulli processes. Losses are therefore spatially and temporally independent. This model is commonly used in the literature [7]–[9], [14]–[16] for network tomography. We define $\alpha_j \in (0, 1]$ as the *link success rate* of the jth link in set $\mathcal{E}_I \cup \mathcal{E}_V$, which is the probability that a packet can successfully be transmitted on the jth link. Thus, $1 - \alpha_j$ denotes the loss rate of the jth link in set $\mathcal{E}_I \cup \mathcal{E}_V$. Moreover, we define $\beta_i \in (0, 1]$ as the *path success rate* of the ith path in set $\mathcal{P}$, which is the probability that a probe packet can successfully be transmitted on the ith path in set $\mathcal{P}$.

Unlike data packets, probe packets would not be retransmitted if they are dropped. We have

$$\prod_{j=1}^{|\mathcal{E}_I \cup \mathcal{E}_V|} (\alpha_j)^{\overline{m}_{i,j}} = \beta_i, \qquad i = 1, \ldots, |\mathcal{P}|. \quad (3)$$

Taking logarithm on both sides of (3), we can reformulate it as linear equations

$$\sum_{j=1}^{|\mathcal{E}_I \cup \mathcal{E}_V|} \overline{m}_{i,j} \log \alpha_j = \log \beta_i, \qquad i = 1, \ldots, |\mathcal{P}| \quad (4)$$

where $\log \alpha_j$ and $\log \beta_i$ are the variables of linear equations. By setting $a_j = \log \alpha_j$ and $b_i = \log \beta_i$, we have

$$\sum_{j=1}^{|\mathcal{E}_I \cup \mathcal{E}_V|} \overline{m}_{i,j} a_j = b_i, \qquad i = 1, \ldots, |\mathcal{P}|. \quad (5)$$

We define two column vectors $\mathbf{a} = (a_j)_{|\mathcal{E}_I \cup \mathcal{E}_V| \times 1}$ and $\mathbf{b} = (b_i)_{|\mathcal{P}| \times 1}$. The system can be represented in the matrix form as

$$\overline{\mathbf{M}}\mathbf{a} = \mathbf{b}. \quad (6)$$

Equation (6) shows the relation between the path and link success rates. The objective of loss tomography is to infer the link loss rates using end-to-end observations (i.e., the number and the contents of the received probe packets). Let $\hat{\mathbf{a}}$ and $\hat{\mathbf{b}}$ denote the estimator of $\mathbf{a}$ and $\mathbf{b}$, respectively. By measuring the path success rates, we can estimate $\hat{\mathbf{b}}$, whereas $\hat{\mathbf{a}}$ remains unknown. Thus, (6) becomes a system of $|\mathcal{P}|$ equations with $|\mathcal{E}_I \cup \mathcal{E}_V|$ unknowns as

$$\overline{\mathbf{M}}\hat{\mathbf{a}} = \hat{\mathbf{b}}. \quad (7)$$

In most cases, the number of identifiable and virtual links is greater than the number of paths. That is, $|\mathcal{E}_I \cup \mathcal{E}_V| > |\mathcal{P}|$. Thus, (7) is underdetermined. We propose the LANT framework to obtain additional useful information and determine $\hat{\mathbf{a}}$.

The LANT framework is composed of two phases. In the first phase, we apply network coding and perform end-to-end measurements on the set of paths $\mathcal{P}$, and n batches of probe packets are sent from the sources in a synchronized manner. In each time slot, the intermediate nodes linearly combine the incoming probes according to specific coding coefficients. The key objective in this phase is to find the minimum probe packet size that can establish the mappings between the contents of the received probe packets and the losses on the different sets of paths. In the second phase, we inspect the contents of the received probe packets. We show that it can provide us with more information than path success rates. We establish a linear system whose coefficient matrix has a full column rank and use computational efficient algorithms to develop consistent estimators of link loss rates. In the next two sections, we describe these two phases in detail.

III. Probe-Coding Schemes

We refer to probe packets and network-coding schemes jointly as *probe-coding schemes*. A probe-coding scheme is *valid* if we can determine which paths have successfully transmitted a probe and which paths have not from the end-to-end observations. We adopt linear network-coding schemes

[27] that are sufficient for our task. In this section, we first determine the minimum probe packet size (i.e., number of bits in each probe packet), which is necessary for valid probe-coding schemes. Then, we propose algorithms to find a valid probe-coding scheme with the minimum probe size.

A. Minimum Probe Packet Size

A probe packet is a binary vector $(\cdot)_2$ of length ℓ, which can be interpreted as an element in a finite field $\mathbb{F}_q$ with an alphabet of size q ($q = 2^\ell$). A coding coefficient can also be interpreted as an element in the finite field $\mathbb{F}_q$. Within valid probe-coding schemes, the probe size ℓ is desired to be as small as possible, because it is directly related to bandwidth efficiency. Although a smaller probe size can reduce the bandwidth usage of the network, the inference framework is not valid if the probe size falls below a certain threshold. For example, in Fig. 1, receiver r receives coded packets that are combined from packets on three different paths. Using 1-b probe packets, we cannot distinguish which of these three paths have successfully transmitted a probe packet. In this case, we need probe packets with at least 3 b for valid probe-coding schemes, whereas smaller probe sizes cannot constitute valid probe-coding schemes.

Before we find the minimum probe size, which is necessary and sufficient for valid probe-coding schemes, we present the notations used in our approach. In a directed acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a link is an *end link* if it is adjacent to a receiver $r \in \mathcal{R}$. The set of all end links is denoted by $\mathcal{E}_R$. For an end link $e \in \mathcal{E}_R$, let $\mathcal{G}_e$ denote a subgraph of $\mathcal{G}$ that consists of the links and nodes involved in set $\mathcal{P}(e)$. We notice that, if receiver r has multiple end links, it would know from which link a packet is received. Let q_e and ℓ_e denote the alphabet size and the length of the probe packets transmitted on subgraph $\mathcal{G}_e$, respectively. The following theorem presents the necessary condition on probe packet size for valid probe-coding schemes.

Theorem: For the probes that are transmitted on subgraph $\mathcal{G}_e$, where $e \in \mathcal{E}_R$, the probe size should satisfy $\ell_e \geq |\mathcal{P}(e)|$ (i.e., $q_e \geq 2^{|\mathcal{P}(e)|}$) to obtain valid end-to-end observations.

The proof of Theorem 1 is given in Appendix B. Although Theorem 1 separately provides the necessary condition on probe size for the probes that are transmitted on different subgraphs, it is not a sufficient condition. For example, in Fig. 2, we have $\ell_{e_6} \geq 2$ and $\ell_{e_7} \geq 4$ for subgraphs $\mathcal{G}_{e_6}$ and $\mathcal{G}_{e_7}$, respectively. However, there are overlapping links in $\mathcal{G}_{e_6}$ and $\mathcal{G}_{e_7}$, e.g., links e_1, e_2 and e_3. In this case, a valid probe size should be 4. Let $\mathcal{G}$ denote a set of subgraphs with overlapping links. The probes that are transmitted on these subgraphs should have the same size. Let $\ell_{\mathcal{G}}$ denote the size of such probes. Thus, the set of end links in the subgraph set $\mathcal{G}$ is denoted by $\mathcal{E}_R(\mathcal{G})$. The following proposition presents the minimum probe size for valid probe-coding schemes.

Proposition 2: For the probes that are transmitted on subgraph set $\mathcal{G}$, the probe size should satisfy $\ell_{\mathcal{G}} \geq \max_{e \in \mathcal{E}_R(\mathcal{G})} |\mathcal{P}(e)|$ to obtain valid end-to-end observations. The minimum probe packet size is equal to $\max_{e \in \mathcal{E}_R(\mathcal{G})} |\mathcal{P}(e)|$.

The proof of Proposition 2 is presented in Appendix C.

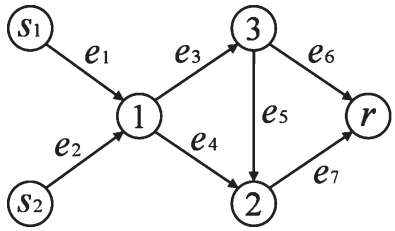


Fig. 2. Directed acyclic graph with two end links e_6 and e_7.

B. Algorithms for Finding a Valid Probe-Coding Scheme

We propose an approach to find a valid probe-coding scheme such that the minimum probe packet size obtained from Proposition 2 is achieved. The approach is divided into three processes, as described in this section.

1) Constructing Auxiliary Trees: For each end link $e = (h, r) \in \mathcal{E}_R$, we introduce an auxiliary tree topology $\mathcal{T}_e$. Each auxiliary tree is associated with one particular end link e to the root node, which is the destination. The leaves of the auxiliary tree correspond to the sources that use link e to relay packets to the root node. The number of leaves is equal to the number of paths that traverse link e. Based on the auxiliary tree topology, we can determine the coding coefficients of the intermediate nodes. To construct an auxiliary tree topology, we first start from the end link e and determine the upstream nodes and links that use link e to relay packets to the root node. This process is repeated in an iterative manner until we find the source nodes whose paths include end link e. New auxiliary nodes are introduced based on the number of outgoing links that are in the same path set as link e. By introducing these auxiliary nodes, we ensure that the topology is a tree instead of a mesh topology.

Algorithm 1: Algorithm for constructing auxiliary trees. Assume that graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is given.

for each end link $e = (h, r) \in \mathcal{E}_R$ **do**
  Add nodes $u_0(r)$ and $u_1(h)$ and link $\varepsilon_1(e)$ into auxiliary tree $\mathcal{T}_e$.
  Initialize the set of leaf nodes $\mathcal{L}_e \leftarrow \{u_1(h)\}$
  $i \leftarrow 2$
  while $\exists$ node $u_k(v) \in \mathcal{L}_e$, $k \in \{1, 2, \ldots, i\}$ and $v \notin \mathcal{S}$ **do**
    for each $v' : \exists(v', v) \in \mathcal{E}$ **do**
      Add node $u_i(v')$ and link $\varepsilon_i(v', v)$ into $\mathcal{T}_e$
      Update $\mathcal{L}_e \leftarrow \mathcal{L}_e \bigcup \{u_i(v')\}$
      $i \leftarrow i + 1$
    end for
    Update $\mathcal{L}_e \leftarrow \mathcal{L}_e \setminus \{u_k(v)\}$.
  end while
end for

Algorithm 1 shows how we can construct the auxiliary trees that correspond to each end link in set $\mathcal{E}_R$. We use $u_0(r)$ to denote the tree node that corresponds the root node r in graph $\mathcal{G}$. We use $u_i(v)$ to denote the ith tree node that corresponds to the nonreceiver node v in graph $\mathcal{G}$. Similarly, we use $\varepsilon_i(e)$

to denote the ith tree link that corresponds to link e in graph $\mathcal{G}$. For each end link $e=(h,r)\in\mathcal{E}_R$, nodes $u_0(r)$ and $u_1(h)$ and link $\varepsilon_1(e)$ are first added into $\mathcal{T}_e$ (step 2). We define a leaf node as a node, only with outgoing links in $\mathcal{T}_e$. Node $u_0(r)$ is the destination node. The set of leaf nodes $\mathcal{L}_e$ initially includes node $u_1(h)$ (step 3). If there exists tree node $u_k(v)\in\mathcal{L}_e$, where $k\in\{1,2,\ldots,i\}$, and v is not a source node in $\mathcal{G}$, then along the incoming links of node v while ignoring the outgoing links, we find a set of nodes. Corresponding to these nodes and the incoming links, new tree nodes and tree links are introduced and added into $\mathcal{T}_e$ (step 7). The set of leaf nodes $\mathcal{L}_e$ is updated in steps 8 and 11. The counter i for the number of tree links in $\mathcal{T}_e$ is updated in step 9.

2) Selecting Coding Coefficients: The coding coefficients are readily obtained based on the auxiliary trees. The nonreceiver nodes with multiple incoming links in $\mathcal{G}$ are the nodes that perform network coding. The corresponding tree nodes would also have multiple incoming tree links. The remaining nodes in $\mathcal{G}$ perform either forwarding or multicasting.

Algorithm 2 shows how we can select coding coefficients. For each node $u_k(v)$ in $\mathcal{T}_e$, suppose that it has a set of $t(u_k(v))$ incoming links $\{\varepsilon_{\text{in}}^1(e_{\text{in}}^1),\ \varepsilon_{\text{in}}^2(e_{\text{in}}^2),\ldots,\varepsilon_{\text{in}}^{t(u_k(v))}(e_{\text{in}}^{t(u_k(v))})\}$ and one outgoing link $\varepsilon_{\text{out}}(e_{\text{out}})$. Then, node v has coding coefficients $[\delta(e_{\text{in}}^1,e_{\text{out}}),\ \delta(e_{\text{in}}^2,e_{\text{out}}),\ldots,\delta(e_{\text{in}}^{t(u_k(v))},e_{\text{out}})]$. Suppose that tree links $\varepsilon_{\text{in}}^1(e_{\text{in}}^1),\ \varepsilon_{\text{in}}^2(e_{\text{in}}^2),\ldots,\varepsilon_{\text{in}}^{t(u_k(v))}(e_{\text{in}}^{t(u_k(v))})$ have $n_1,n_2,\ldots,n_{t(u_k(v))}$ corresponding leaf nodes, respectively. Then, we choose the values of coding coefficients as $[2^0,2^{n_1},2^{n_1+n_2},\ldots,2^{n_1+n_2+\cdots+n_{t(u_k(v))-1}}]$.

Algorithm 2: Algorithm for selecting coding coefficients. Assume that the auxiliary trees are given.

1: **for** each auxiliary tree $\mathcal{T}_e$ **do**
2: **for** each node $u_k(v)$ in $\mathcal{T}_e$ with outgoing link $\varepsilon_{\text{out}}(e_{\text{out}})$ **do**
3: $n_0\leftarrow 0$
4: **for** each incoming link $\varepsilon_{\text{in}}^i(e_{\text{in}}^i)$ of node $u_k(v)$, $i=1,2,\ldots,t(u_k(v))$ **do**
5: Find its corresponding leaf-node set $\mathcal{L}(\varepsilon_{\text{in}}^i(e_{\text{in}}^i))\subseteq\mathcal{L}_e$
6: $n_i\leftarrow|\mathcal{L}(\varepsilon_{\text{in}}^i(e_{\text{in}}^i))|$
7: $\delta(e_{\text{in}}^i,e_{\text{out}}^i)\leftarrow 2^{n_0+n_1+\cdots+n_{i-1}}$
8: **end for**
9: **end for**
10: **end for**

Designing Probe Packets: The path–link matrix $\mathbf{M}$ can be obtained based on the paths from the leaf nodes to the destination node in each auxiliary tree. Now, we describe how we can find the sets of subgraphs with overlapping links. Each subgraph $\mathcal{G}_e$ originally constitutes a subgraph set $\{\mathcal{G}_e\}$. We check each column of the path–link matrix $\mathbf{M}$. If a column has multiple 1s and it also represents that different subgraph sets include the same link, we combine these subgraph sets as one subgraph set. Then, for each subgraph set $\mathcal{G}$ with its end-link set $\mathcal{E}_R(\mathcal{G})$, according to Proposition 2, we calculate the minimum probe size $\ell_{\mathcal{G}}=\max_{e\in\mathcal{E}_R(\mathcal{G})}|\mathcal{L}_e|$. Thus, probes as $(0\cdots01)_2$ of length $\ell_{\mathcal{G}}$ are sent from the sources to the outgoing links in $\mathcal{G}$.

Finally, for each path $P_i\in\mathcal{P}$, multiplying $(0\cdots01)_2$ of its corresponding length by the coding coefficients along path P_i, we can obtain the contents of a received probe that denotes the case where only path P_i has successfully transmitted a probe. This way, we can establish the mapping between the contents of received probes and the losses on the different combinations of paths for each subgraph and thus obtain a valid probe-coding scheme.

Example 1: We consider a directed acyclic graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$, as depicted in Fig. 2. We use Algorithms 1 and 2 to obtain a valid probe-coding scheme with the minimum probe size. First, we construct two auxiliary trees $\mathcal{T}_{e_6}$ and $\mathcal{T}_{e_7}$, as depicted in Fig. 3. Second, we choose nodes 1 and 2 as the nodes that perform network coding, because they are the nonreceiver nodes with multiple incoming links. Based on the auxiliary tree $\mathcal{T}_{e_6}$, we have $|\mathcal{L}(\varepsilon_3(e_1))|=1$ (see Algorithm 2, steps 5 and 6). Thus, some of the coding coefficients of node 1 are obtained as $[\delta(e_1,e_3),\delta(e_2,e_3)]=[1,2]$ (see Algorithm 2, step 7). Similarly, based on $\mathcal{T}_{e_7}$, we obtain the coding coefficients of node 1 as $[\delta(e_1,e_4),\delta(e_2,e_4)]=[1,2]$. We also obtain the coding coefficients of node 2 as $[\delta(e_4,e_7),\delta(e_5,e_7)]=[1,4]$, because $|\mathcal{L}(\varepsilon_2(e_4))|=2$.

Third, the path–link matrix $\mathbf{M}$ can be obtained as follows:

$$\mathbf{M}=\begin{array}{c}\begin{matrix}e_1&e_2&e_3&e_4&e_5&e_6&e_7\end{matrix}\\\begin{bmatrix}1&0&1&0&0&1&0\\0&1&1&0&0&1&0\\1&0&0&1&0&0&1\\0&1&0&1&0&0&1\\1&0&1&0&1&0&1\\0&1&1&0&1&0&1\end{bmatrix}\end{array}. \tag{8}$$

The top two rows represent the two paths in subgraph $\mathcal{G}_{e_6}$, and the bottom four rows represent the four paths in subgraph $\mathcal{G}_{e_7}$. Because link e_1 is involved in both $\{\mathcal{G}_{e_6}\}$ and $\{\mathcal{G}_{e_7}\}$ (checking the first column of $\mathbf{M}$), we combine the two subgraph sets and obtain one subgraph set $\mathcal{G}=\{\mathcal{G}_{e_6},\mathcal{G}_{e_7}\}$. Counting the number of leaf nodes in each auxiliary tree, we obtain $|\mathcal{P}(e_6)|=2$ and $|\mathcal{P}(e_7)|=4$. Thus, $\ell_{\mathcal{G}}=\max\{2,4\}=4$, and probes $(0001)_2$ are sent from sources s_1 and s_2 to outgoing links e_1 and e_2, respectively. □

IV. Linear Algebraic Approach

As aforementioned, the system in (7) has $|\mathcal{P}|$ equations with $|\mathcal{E}_I\cup\mathcal{E}_V|$ unknowns. However, $|\mathcal{P}|$ may be less than $|\mathcal{E}_I\cup\mathcal{E}_V|$, e.g., the topologies in Figs. 1 and 2, and thus, $\hat{\mathbf{a}}$ in (7) cannot uniquely be determined. Although $|\mathcal{P}|\geq|\mathcal{E}_I\cup\mathcal{E}_V|$, it does not ensure that $\hat{\mathbf{a}}$ can be determined. In this section, we propose a LA approach using the observations from coding operations. We show that $\hat{\mathbf{a}}$ can be determined using least squares [28]. Then, we combine the methods of normal equations and row selection with the LA approach and analyze the computational complexity.

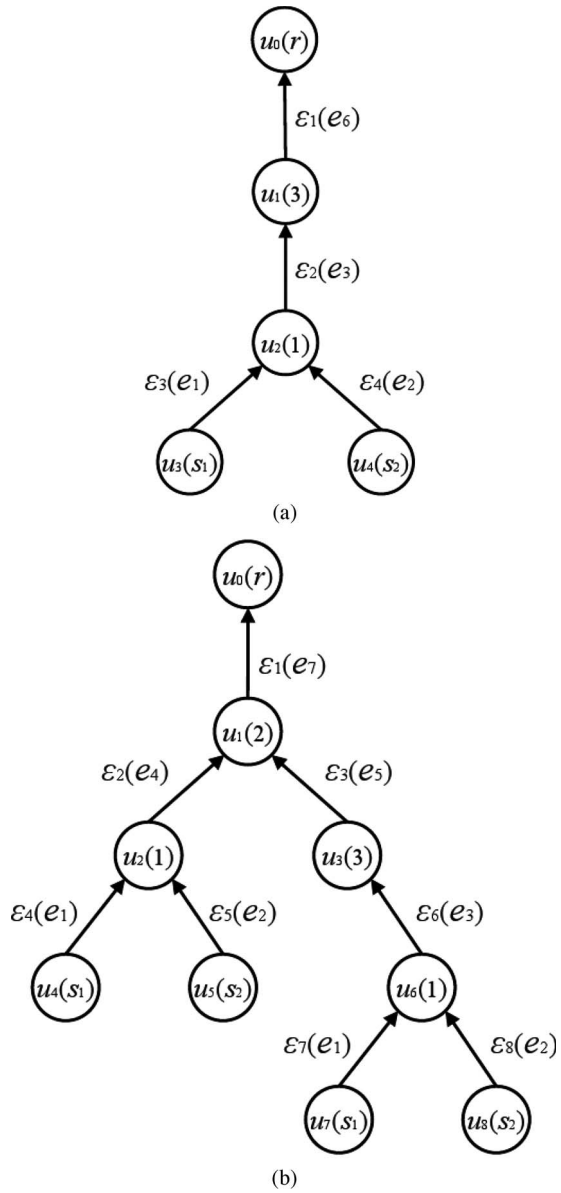


Fig. 3. Two auxiliary trees $\mathcal{T}_{e_6}$ and $\mathcal{T}_{e_7}$, which correspond to end links e_6 and e_7, respectively. (a) Auxiliary tree $\mathcal{T}_{e_6}$. (b) Auxiliary tree $\mathcal{T}_{e_7}$.

A. LA Approach

By inspecting the contents of the received coded probe packets at the destinations, we can estimate the success rate not only of a single path but of a set of paths as well. As the consequence of valid probe-coding schemes, it enables us to distinguish between the paths that have contributed to a coded probe packet and the paths that have not. This condition is unique to networks that use probe-coding schemes and cannot be achieved by routing probes in general.

We denote the power set[2] of $\mathcal{P}$ by $\mathcal{P}$. Thus, $|\mathcal{P}| = 2^{|\mathcal{P}|}$. Each element of $\mathcal{P}$ is a subset of $\mathcal{P}$, which can be used to represent a unique combination of paths. Let θ_i denote the *path set success rate* of the ith path set in $\mathcal{P} \setminus \{\varnothing\}$, which is the probability that a batch of probes can successfully be transmitted on all the paths in the ith path set. We define a path set success rate, except for $\varnothing \in \mathcal{P}$, because we require the probability that at least one path can successfully transmit a probe to obtain an equation of link success rates and a path set success rate.

Accordingly, we define a modified path–link matrix $\widetilde{\mathbf{M}} = (\widetilde{m}_{i,j})_{(|\mathcal{P}|-1)\times|\mathcal{E}_I \cup \mathcal{E}_V|}$ as follows. The element $\widetilde{m}_{i,j}$ is equal to 1 if there exists a path in the ith path set in set $\mathcal{P} \setminus \{\varnothing\}$, which includes the jth link in set $\mathcal{E}_I \cup \mathcal{E}_V$; otherwise, it is equal to 0. We refer to $\widetilde{\mathbf{M}}$ as a type-2 modified path–link matrix.

The type-2 modified path–link matrix for the graph in Fig. 1 is shown as

$$\widetilde{\mathbf{M}} = \left[\begin{array}{cccccc} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ \hline 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right]. \tag{9}$$

Each of the top three rows represents the path set that includes only one path.

We define a column vector, $\mathbf{c} = (c_i)_{(|\mathcal{P}|-1)\times 1}$, where $c_i = \log \theta_i$. The column vector $\mathbf{a}$ is defined as in Section II. Thus, we have a linear system

$$\sum_{j=1}^{|\mathcal{E}_I \cup \mathcal{E}_V|} \widetilde{m}_{i,j} a_j = c_i, \qquad i = 1, \ldots, |\mathcal{P}| - 1 \tag{10}$$

or, in the matrix form, we have

$$\widetilde{\mathbf{M}}\mathbf{a} = \mathbf{c}. \tag{11}$$

For each path set in $\mathcal{P} \setminus \{\varnothing\}$, the n probe batches that are sent from the sources can be considered a binomial experiment that consists of n trials. The associated binomial random variable X_i is defined as the number of received coded probes (or probe batches for more than one incoming link) whose contents represent that all the paths in the ith path set have successfully transmitted a probe packet.

The sample proportion $\hat{\theta}_i = X_i/n$ is a maximum-likelihood (ML) estimator of θ_i [29] (or an ML estimate that results from end-to-end measurement x_i substituted in place of X_i). Accordingly, we can obtain $\hat{\mathbf{c}}$, the estimator of $\mathbf{c}$. The column vector $\hat{\mathbf{a}}$ remains unknown. Thus, we extend (7) to a system of $|\mathcal{P}| - 1$ equations with $|\mathcal{E}_I \cup \mathcal{E}_V|$ unknowns as follows:

$$\widetilde{\mathbf{M}}\hat{\mathbf{a}} = \hat{\mathbf{c}}. \tag{12}$$

B. Least Squares Solutions

The linear system in (12) has more equations than unknowns, i.e.,

$$|\mathcal{P}| - 1 \geq |\mathcal{E}_I \cup \mathcal{E}_V|. \tag{13}$$

This case is because, for every pair of links e, $e' \in \mathcal{E}_I \cup \mathcal{E}_V$, $\mathcal{P}(e)$ is different from $\mathcal{P}(e')$, whereas $|\mathcal{P}|$ paths can have at most $2^{|\mathcal{P}|} - 1$ different combinations of paths. The inequality (13) is a necessary condition for $\hat{\mathbf{a}}$ to be determined.

[2]The power set of a set is the set of all subsets of that set. For example, the power set of $\{a, b\}$ is $\{\varnothing, \{a\}, \{b\}, \{a, b\}\}$.

To show that $\hat{\mathbf{a}}$ in (12) can be determined by the least squares, we introduce a $(|\mathcal{P}|-1)\times(|\mathcal{P}|-1)$ auxiliary matrix $\mathcal{M}(|\mathcal{P}|)$ of a type-2 modified path–link matrix $\widetilde{\mathbf{M}}$ with an end-to-end path set $\mathcal{P}$. Compared with $\widetilde{\mathbf{M}}$, $\mathcal{M}(|\mathcal{P}|)$ has additional column vectors and has $|\mathcal{P}|-1$ column vectors in total. The $|\mathcal{P}|-1$ column vectors in the top $|\mathcal{P}|\times(|\mathcal{P}|-1)$ submatrix can represent all-nonzero vectors in the vector space $\mathbb{F}_2^{|\mathcal{P}|}$. The bottom part of the additional columns is obtained according to the relation between the paths and path sets. One example of $\mathcal{M}(3)$ for $\widetilde{\mathbf{M}}$ in (9) is given as follows:

$$\mathcal{M}(3)=\left[\begin{array}{ccccccc}1&1&0&0&1&0&1\\1&1&0&1&0&1&0\\1&0&1&0&0&1&1\\--&--&--&--&--&--&--\\1&1&0&1&1&1&1\\1&1&1&0&1&1&1\\1&1&1&1&0&1&1\\1&1&1&1&1&1&1\end{array}\right]. \tag{14}$$

Lemma 1: Let $\mathcal{M}(|\mathcal{P}|)$ be an auxiliary matrix of a type-2 modified path–link matrix $\widetilde{\mathbf{M}}$ with an end-to-end path set $\mathcal{P}$. Then, $\text{rank}(\mathcal{M}(|\mathcal{P}|))=2^{|\mathcal{P}|}-1$, i.e., all $2^{|\mathcal{P}|}-1$ column vectors in $\mathcal{M}(|\mathcal{P}|)$ are linearly independent.

The proof of Lemma 1 is shown in Appendix D. With Lemma 1, the following theorem gives the rank of a type-2 modified path–link matrix.

Theorem 2: Let a directed acyclic graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$ be given with a system of linear equations in matrix form $\widetilde{\mathbf{M}}\hat{\mathbf{a}}=\hat{\mathbf{c}}$. Then, $\text{rank}(\widetilde{\mathbf{M}})=|\mathcal{E}_I\cup\mathcal{E}_V|$.

Proof: Let $\mathcal{M}(|\mathcal{P}|)$ be an auxiliary matrix of $\widetilde{\mathbf{M}}$. The $|\mathcal{E}_I\cup\mathcal{E}_V|$ column vectors in $\widetilde{\mathbf{M}}$ are among the $2^{|\mathcal{P}|}-1$ column vectors in $\mathcal{M}(|\mathcal{P}|)$. Based on Lemma 1, all $2^{|\mathcal{P}|}-1$ column vectors in $\mathcal{M}(|\mathcal{P}|)$ are linearly independent. Thus, these $|\mathcal{E}_I\cup\mathcal{E}_V|$ column vectors in $\widetilde{\mathbf{M}}$ are also linearly independent. As a result, $\text{rank}(\widetilde{\mathbf{M}})=|\mathcal{E}_I\cup\mathcal{E}_V|$. ■

Corollary 1: Let $\widetilde{\mathbf{M}}\hat{\mathbf{a}}=\hat{\mathbf{c}}$ be given. Then, $\hat{\mathbf{a}}$ can be determined by least squares.

Proof: When the number of equations is equal to the number of unknowns, i.e., $|\mathcal{P}|-1=|\mathcal{E}_I\cup\mathcal{E}_V|$, $\widetilde{\mathbf{M}}$ is a square matrix. Theorem 2 ensures that $\widetilde{\mathbf{M}}$ is invertible. Thus, $\hat{\mathbf{a}}$ can be determined as

$$\hat{\mathbf{a}}=\widetilde{\mathbf{M}}^{-1}\hat{\mathbf{c}}. \tag{15}$$

When the number of equations is greater than the number of unknowns, i.e., $|\mathcal{P}|-1>|\mathcal{E}_I\cup\mathcal{E}_V|$, the system is overdetermined. We can apply least squares [28] to obtain an approximate solution that minimizes the residual error $\|\hat{\mathbf{c}}-\widetilde{\mathbf{M}}\hat{\mathbf{a}}\|$. Theorem 2 ensures that $\widetilde{\mathbf{M}}^{\mathrm{T}}\widetilde{\mathbf{M}}$ is invertible. Thus, $\hat{\mathbf{a}}$ can be determined as

$$\hat{\mathbf{a}}=(\widetilde{\mathbf{M}}^{\mathrm{T}}\widetilde{\mathbf{M}})^{-1}\widetilde{\mathbf{M}}^{\mathrm{T}}\hat{\mathbf{c}}. \tag{16}$$

We note that (15) is a special case of (16). ■

The aforementioned corollary gives an analytical solution of $\hat{\mathbf{a}}$ using least squares. In statistics, a sequence of estimators $\hat{\theta}_i$ for parameter θ_i is said to be *consistent* if this sequence converges in probability to θ_i. The following theorem demonstrates the consistency of the corresponding estimators.

Theorem 3: $1-\hat{\alpha}_j$ is a consistent estimator of $1-\alpha_j$.

Proof: For each link $e_j\in\mathcal{E}_I\cup\mathcal{E}_V$, $1-\hat{\alpha}_j$ is a function of $\hat{\alpha}_j$, whereas $\hat{\alpha}_j$ is a function of $\hat{\theta}_1,\hat{\theta}_2,\ldots,\hat{\theta}_{|\mathcal{P}|-1}$. Let $\xrightarrow{p}$ denote the convergence in probability. Because $\hat{\theta}_i\xrightarrow{p}\theta_i$, the continuous-mapping and Slutsky's theorems [29] yield that $1-\hat{\alpha}_j\xrightarrow{p}1-\alpha_j$. ■

Proposition 3: The loss rate of a link can consistently be estimated *if and only if* the link is an identifiable link.

Proof: Theorem 3 shows that the loss rates of all identifiable links can consistently be estimated by the estimators. In addition, Proposition 1 shows that, if the loss rate of a link can be estimated, then the link is identifiable. These conditions together prove this proposition. ■

Although the loss rate of the links that are not identifiable cannot consistently be estimated, we can, at least, obtain an upper bound on their loss rate, which is the loss rate of the corresponding virtual link.

C. Method of Normal Equations

One common technique for solving a full-rank least squares problem is the method of normal equations [30]. We define $\mu=|\mathcal{P}|-1$ and $\nu=|\mathcal{E}_I\cup\mathcal{E}_V|$ so that $\widetilde{\mathbf{M}}$ is a $\mu\times\nu$ matrix. The first step in the method of normal equations is to calculate the symmetric matrix (i.e., $\widetilde{\mathbf{M}}^{\mathrm{T}}\widetilde{\mathbf{M}}$). This approach requires $\mu\nu^2$ flops.[3] The second step is to calculate the Cholesky decomposition $\widetilde{\mathbf{M}}^{\mathrm{T}}\widetilde{\mathbf{M}}=\mathbf{L}\mathbf{L}^{\mathrm{T}}$, which requires $\nu^3/3$ flops. The third step is to calculate $\widetilde{\mathbf{M}}^{\mathrm{T}}\hat{\mathbf{c}}$, which requires $2\mu\nu$ flops. The fourth and fifth steps are to solve $\mathbf{L}\mathbf{y}=\widetilde{\mathbf{M}}^{\mathrm{T}}\hat{\mathbf{c}}$ for y using forward substitution and to solve $\mathbf{L}^{\mathrm{T}}\hat{\mathbf{a}}=\widetilde{\mathbf{M}}^{\mathrm{T}}\hat{\mathbf{c}}$ for $\hat{\mathbf{a}}$ using back substitution, each of which requires ν^2 flops. Considering $\mu\geq\nu$ by (13), the complexity of this method is $\mathcal{O}(\mu\nu^2)$.

Although the first step in the method of normal equations includes the dominant term of the complexity, it only needs to be executed once for the initial setup, as long as the network topology remains unchanged. We need to obtain $\hat{\mathbf{c}}$ before we can calculate $\widetilde{\mathbf{M}}^{\mathrm{T}}\hat{\mathbf{c}}$ and perform forward/back substitutions (steps 3–5). The complexity in calculating $\hat{\mathbf{c}}$ is $\mathcal{O}(\mu n)$, where n is the number of probe batches. This step and steps 3–5 can be repeated κ times in a monitoring period. Thus, the LA approach that uses the method of normal equations has a complexity of $\mathcal{O}(\mu\nu^2+\mu n\kappa+\mu\nu\kappa)$ in practice.

D. Method of Row Selection

Because the number of path sets $\mu=2^{|\mathcal{P}|}-1$, μ exponentially grows as $|\mathcal{P}|$ increases. As a result, the method of least squares would lack scalability and thus have high complexity. Nonetheless, according to Theorem 2, $\text{rank}(\widetilde{\mathbf{M}})=\nu$. This condition means that there exist ν linearly independent path sets out of μ path sets, which are sufficient to determine $\hat{\mathbf{a}}$.

[3] A flop is a floating-point operation. The flop count is useful as a rough estimate of complexity and predictor of computational time on modern computers.

To select ν linearly independent path sets, we modify the row selection algorithm proposed in [24] and obtain a reduced linear system as

$$\widetilde{\mathbf{M}}_1 \hat{\mathbf{a}} = \hat{\mathbf{c}}_1 \tag{17}$$

where $\widetilde{\mathbf{M}}_1 \in \{0,1\}^{\nu \times \nu}$, and $\hat{\mathbf{c}}_1 \in \mathbb{R}^{\nu}$ consists of ν rows of $\widetilde{\mathbf{M}}$ and $\hat{\mathbf{c}}$, respectively. Algorithm 3 shows the modified row (path set) selection algorithm. This algorithm incrementally builds a QR factorization $\widetilde{\mathbf{M}}_1^{\mathrm{T}} = QR$, where $Q \in \mathbb{R}^{\nu \times \nu}$ is an orthogonal matrix, and $R \in \mathbb{R}^{\nu \times \nu}$ is an upper triangular matrix. It only needs to be executed once for the initial setup with a complexity of $\mathcal{O}(\mu\nu^2)$.

Algorithm 3: Modified row (path set) selection algorithm.

1: Initialize $\widetilde{\mathbf{M}}_1 \leftarrow$ the first row in $\widetilde{\mathbf{M}}$.
2: Initialize R by calculating the thin QR factorization of $\widetilde{\mathbf{M}}_1^{\mathrm{T}}$.
3: **while** $\widetilde{\mathbf{M}}_1$ is not a square matrix **do**
4: $\quad \omega \leftarrow$ next row in $\widetilde{\mathbf{M}}$
5: $\quad \hat{R}_{12} \leftarrow R^{-\mathrm{T}} \widetilde{\mathbf{M}}_1 \omega^{\mathrm{T}}$
6: $\quad \hat{R}_{22} \leftarrow \|\omega\|^2 - \|\hat{R}_{12}\|^2$
7: $\quad$ **if** $\hat{R}_{22} \neq 0$ **then**
8: $\quad\quad$ Update $R \leftarrow \begin{bmatrix} R & \hat{R}_{12} \\ \mathbf{0} & \hat{R}_{22} \end{bmatrix}$.
9: $\quad\quad$ Update $\widetilde{\mathbf{M}}_1 \leftarrow \begin{bmatrix} \widetilde{\mathbf{M}}_1 \\ \omega \end{bmatrix}$.
10: $\quad$ **end if**
11: **end while**

The complexity of calculating $\hat{\mathbf{c}}_1$ is reduced to $\mathcal{O}(\nu n)$. Then, we calculate $\hat{\mathbf{a}} = \widetilde{\mathbf{M}}_1^{\mathrm{T}} \mathbf{z}$, where $\mathbf{z} = R^{-1}(R^{-1})^{\mathrm{T}} \hat{\mathbf{c}}_1$, whose complexity is $\mathcal{O}(\nu^2)$. We repeat the aforementioned steps for κ times in a monitoring period. Thus, the LA approach that uses the method of row selection has a lower complexity of $\mathcal{O}(\mu\nu^2 + \nu n\kappa + \nu^2\kappa)$ in practice.

E. Mobility and Topology Changes

The movement of nodes can change the network topology. Some new links may appear, whereas some other links may no longer be operational. Consequently, the set of paths would change as the topology changes. For inference purposes, as soon as the topology changes, the matrix $\widetilde{\mathbf{M}}_1$ needs to be updated. The update process consists of deleting some of the existing links and paths and adding new links and paths. In this section, we discuss how we can update the matrix $\widetilde{\mathbf{M}}_1$ for the row selection algorithm. We use the modified version in [24, Alg. 2] to update matrix $\widetilde{\mathbf{M}}_1$.

When the topology changes, both the addition and deletion of paths and links may happen. In our update algorithm, we first update the matrices by removing the deleted links and paths. Then, we consider adding new links and paths to the matrices. Algorithm 4 shows the update algorithm for matrices $\widetilde{\mathbf{M}}_1$ and $\widetilde{\mathbf{M}}$ when the network topology changes. Let $\mathcal{F}^d$ and $\mathcal{F}_1^d$ denote the set of deleted paths from matrices $\widetilde{\mathbf{M}}$ and $\widetilde{\mathbf{M}}_1$, respectively. In the update process, we first remove the columns that correspond to the deleted links in matrices $\widetilde{\mathbf{M}}$ and $\widetilde{\mathbf{M}}_1$ (step 1). To update matrix R, we use the algorithm in [31, p. 338, Algorithm 3.4]. This algorithm is efficient in terms of complexity compared with the QR factorization. Because the matrix $\widetilde{\mathbf{M}}_1$ is full rank before removing the links and paths, the rows of the matrix represent different dimensions in the space spanned by all the row vectors. Before removing the deleted paths from matrix $\widetilde{\mathbf{M}}_1$, for each path $v \in \mathcal{F}_1^d$, we find a vector y_v that describes only the dimension that was removed by deleting the path v. To find y_v, we need to solve the linear system $\widetilde{\mathbf{M}}_1 y_v = e_{i_v}$, where e_{i_v} is the vector with zero in all entries, except for entry i_v which has value one, and i_v is the row number for the path v in $\widetilde{\mathbf{M}}_1$. The solution of linear system is $y_v = \widetilde{\mathbf{M}}_1^{-T} R^{-1} R^{-T} e_{i_v}$ (see steps 3–5). Then, we delete all the paths in $\mathcal{F}^d$ and $\mathcal{F}_1^d$ from matrices $\widetilde{\mathbf{M}}$ and $\widetilde{\mathbf{M}}_1$, respectively, and update matrix R again (see steps 6 and 7). If matrix $\widetilde{\mathbf{M}}_1$ has rank deficiency, we need to add new paths from $\widetilde{\mathbf{M}}$ to $\widetilde{\mathbf{M}}_1$. For each vector y_v, we check whether there is a path in $\widetilde{\mathbf{M}}$, which has a nonzero component on the direction of vector y_v (i.e., the row vector of this path is not orthogonal to vector y_v). If such a path exists, we add it to matrix $\widetilde{\mathbf{M}}_1$ following steps 4–9 of Algorithm 3 (see steps 8–14).

Algorithm 4: Update algorithm.

1: Remove the columns that correspond to the deleted links from $\widetilde{\mathbf{M}}$ and $\widetilde{\mathbf{M}}_1$.
2: Update R using [31, p. 338, Alg. 3.4].
3: **for** $v \in \mathcal{F}_1^d$ **do**
4: $\quad y_v \leftarrow \widetilde{\mathbf{M}}_1^T R^{-1} R^{-T} e_{i_v}$
5: **end for**
6: Remove paths $\mathcal{F}^d$ and $\mathcal{F}_1^d$ from $\widetilde{\mathbf{1} : \mathbf{M}}$ and $\widetilde{\mathbf{M}}_1$, respectively.
7: Update R using [31, p. 338, Alg. 3.4]
8: **while** $\widetilde{\mathbf{M}}_1$ is not a square matrix **do**
9: $\quad v \leftarrow$ next path in $\mathcal{F}_1^d$
10: $\quad r_v \leftarrow \widetilde{\mathbf{M}} y_v$
11: $\quad$ **if** $\exists\ j$ such that the jth entry of r_v is nonzero **then**
12: $\quad\quad$ Add row j from $\widetilde{\mathbf{M}}$ to $\widetilde{\mathbf{M}}_1$ (see Algorithm 3, steps 4–9)
13: $\quad$ **end if**
14: **end while**
15: Add new links to matrices $\widetilde{\mathbf{M}}$ and $\widetilde{\mathbf{M}}_1$.
16: Add paths $\mathcal{F}^a$ to $\widetilde{\mathbf{M}}$.
17: Update R using [31, p. 338, Alg. 3.4].
18: **while** $\widetilde{\mathbf{M}}_1$ is not a square matrix **do**
19: $\quad v \leftarrow$ next path in $\mathcal{F}^a$
20: $\quad$ Add path v from $\widetilde{\mathbf{M}}$ to $\widetilde{\mathbf{M}}_1$ if it can increase the rank of this matrix (see Algorithm 3, steps 4–9).
21: **end while**

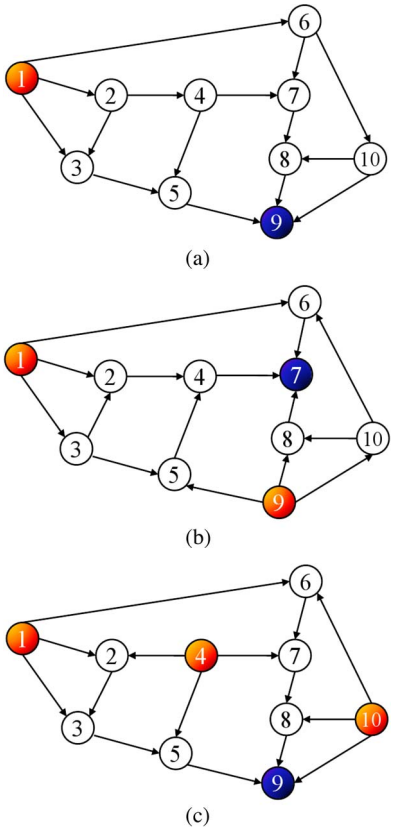


Fig. 4. Directed acyclic graphs with different numbers of sources. (a) One source (node 1) and one receiver (node 9). (b) Two sources (nodes 1 and 9) and one receiver (node 7). (c) Three sources (nodes 1, 4, and 10) and one receiver (node 9).

Next, we update the matrices $\widetilde{\mathbf{M}}$ and $\widetilde{\mathbf{M}}_1$ by adding the new links and paths. Let $\mathcal{F}^a$ denote the set of new paths added to matrix $\widetilde{\mathbf{M}}$. We first add the new links to both matrices $\widetilde{\mathbf{M}}$ and $\widetilde{\mathbf{M}}_1$ (see step 15). The newly added columns are zero vectors, because these links are not part of the existing paths. Then, we add all the new paths to matrix $\widetilde{\mathbf{M}}$ and update matrix R (see steps 16 and 17). Then, we check the new paths one by one and add the paths that can increase the rank of matrix $\widetilde{\mathbf{M}}_1$ (see steps 18–21). The algorithm is terminated as soon as matrix $\widetilde{\mathbf{M}}_1$ is of full rank.

V. Performance Evaluation

In this section, we access the performance of the LANT framework by simulations. For the network topology, we first consider the Internet2 Network Map [32], which is a backbone network that was created by the Internet2 community. The topology is modified as the method used in [22], which consists of 10 nodes and 15 links. We apply the orientation algorithm [22] that converts the modified topology with selected sources to three directed acyclic graphs with different numbers of sources, as shown in Fig. 4, where all the links are identifiable. The destinations are also determined by the orientation algorithm. For other topologies with a larger number of nodes, we use Boston University Representative Internet Topology Generator (BRITE) [33] to generate router-level undirected

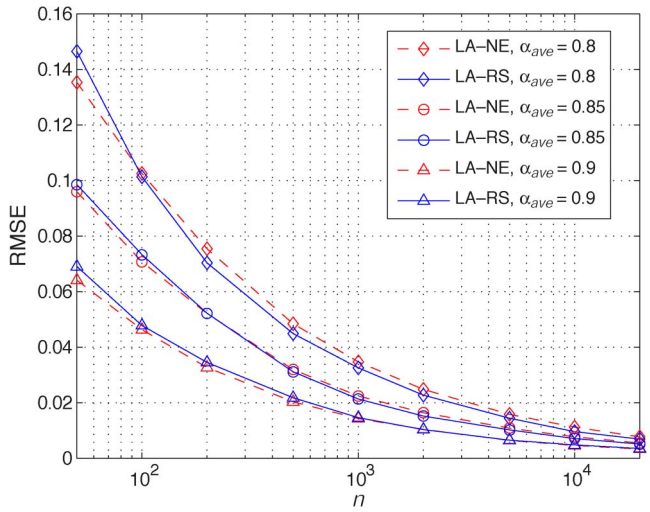


Fig. 5. RMSEs of LA-NE and LA-RS versus the number of probe batches n for different average success rates α_{ave}.

network topologies with the Waxman model. The number of nodes is chosen as 20, 100, and 500.

A random link loss rate $1-\alpha_j$ is assigned to the jth link $e_j \in \mathcal{E}_I \cup \mathcal{E}_V$, where the link success rate α_j is uniformly distributed within $[\alpha_{ave}-0.05, \alpha_{ave}+0.05]$. The value of α_{ave} is chosen as 0.7, 0.75, 0.8, 0.85, 0.9, and 0.95 to adjust the average success rate across all links. After assigning each link a loss rate, we send n batches of probe packets. Each probe that traverses a link is dropped at a fixed probability as the link loss rate.

In each simulation, we obtain an estimate $1-\hat{\alpha}_j$ of the actual link loss rate $1-\alpha_j$ for the jth link in set $\mathcal{E}_I \cup \mathcal{E}_V$. The root-mean-square error (RMSE) is used to determine the estimation accuracy across all identifiable links and virtual links. The RMSE is computed as

$$\text{RMSE} = \left(\sum_{j=1}^{|\mathcal{E}_I \cup \mathcal{E}_V|} \frac{|\alpha_j - \hat{\alpha}_j|^2}{|\mathcal{E}_I \cup \mathcal{E}_V|} \right)^{1/2}. \tag{18}$$

We compare our LANT framework and the BP algorithm [22] for loss rate inference. The results are averaged over 100 simulations to eliminate possible random effects, where each simulation has new loss rate assignments and new loss processes.

A. Simulation Results

First, we investigate the influence of different methods that are adopted by the LA approach on the estimation accuracy based on the graph with one source in Fig. 4(a). The type-2 modified path–link matrix $\widetilde{\mathbf{M}}$ has 127 rows, and we apply the method of normal equations. This case is denoted by LA-NE. Alternatively, we use Algorithm 3 to build a square matrix $\widetilde{\mathbf{M}}_1$, where each row (path set) includes one or two paths. This case is denoted by LA-RS. Fig. 5 shows the RMSE of the LA approach using the two methods as a function of the number of probe batches. We observe that the RMSE of the LA-NE algorithm is lower than the LA-RS algorithm when $n = 50$. Such

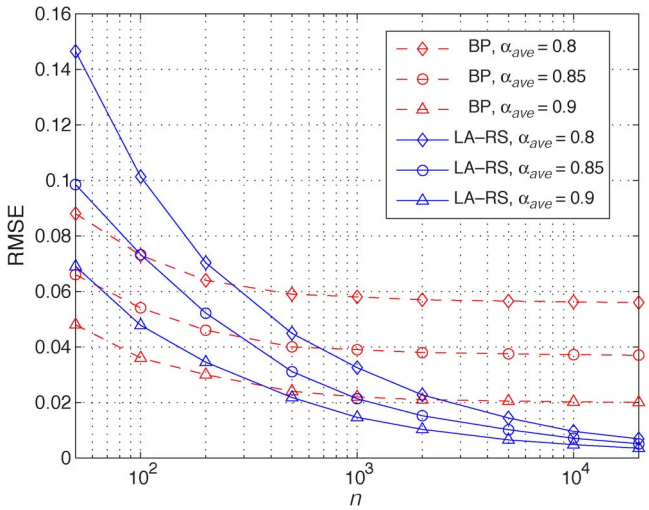


Fig. 6. RMSEs of the BP [25] and LA-RS algorithms versus the number of probe batches n for different average success rates α_{ave}.

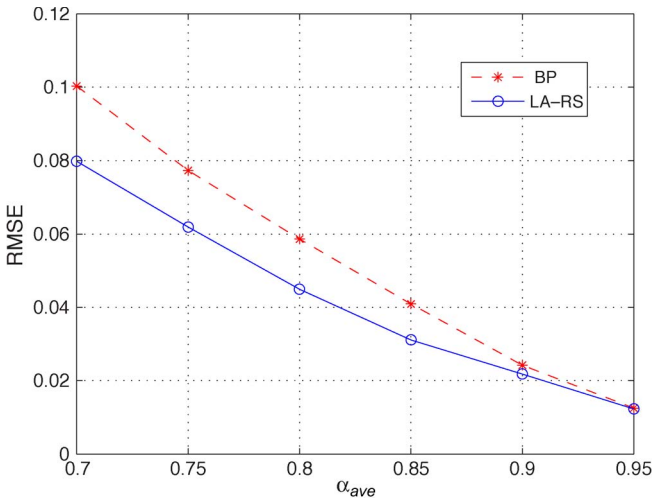


Fig. 7. RMSEs of the BP [25] and LA-RS algorithms versus the average success rate α_{ave} $(n = 500)$.

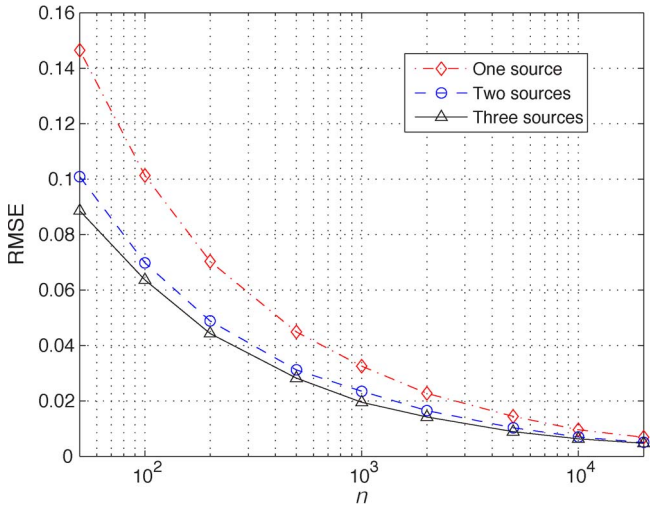


Fig. 8. RMSE of the LA-RS algorithm versus the number of probe batches n for different numbers of sources $(\alpha_{ave} = 0.8)$.

behavior is reasonable, because the LA-NE algorithm uses more equations to obtain link loss rates than the LA-RS algorithm. However, the difference of the RMSE is less than 2%, and it vanishes as the number of probes increases. For a large number of probes, although the RMSE between LA-RS and LA-NE algorithms is similar, the LA-RS algorithm outperforms the LA-NE algorithm in terms of the computational complexity. Note that the number of probe packets indicates the overhead (i.e., additional bandwidth usage) incurred on the network networking scheme. A larger number of probe packets correspond to a higher bandwidth usage. Thus, Fig. 5 also shows the tradeoff between the overhead of the algorithm and the accuracy of the estimations. The higher the overhead is, the higher the accuracy of the estimations becomes.

We then compare the estimation accuracy of the BP and the LA-RS algorithms based on the graph with one source in Fig. 4(a). Fig. 6 shows the RMSE as a function of the number of probe batches for different average link success rates. We observe that the BP algorithm has better accuracy when $n < 400$ and that the LA-RS algorithm achieves better accuracy after sending reasonably sufficient probe batches $(n > 400)$. This case is because the LA-RS algorithm exploits the losses on the different combinations of paths, whereas the BP algorithm only utilizes the losses on paths. Fig. 7 shows the RMSE as a function of the average link success rate with 500 probe batches. The RMSE decreases as the average link success rate increases, which is consistent with the relative position of the curves in Fig. 6. Based on these two graphs, we can predict that, when average loss rates are lower than 0.8, the BP algorithm would perform worse (RMSE $> 6\%$, $n = 20\,000$), whereas the LA-RS algorithm would still achieve satisfactory accuracy (RMSE $< 1\%$, $n = 20\,000$).

For the networks with different numbers of sources in Fig. 4, Fig. 8 shows the RMSE as a function of the number of probe batches, and Fig. 9 shows the RMSE as a function of the average success rate α_{ave}. We compare the relative position of the three curves in Figs. 8 and 9 and obtain the following observation: The graph with more sources achieves better estimation accuracy. However, the improvement of estimation accuracy is negligible with relatively large success rates or sufficient probe batches. In this case, we can use a small number of sources and flexibly choose their locations.

Next, we evaluate the performance of LA-RS for topologies with different numbers of nodes. We use BRITE [33] to generate the topology and apply the orientation algorithm to create the directed graph [22]. We compare the LA-RS algorithm with the BP algorithm for networks with 20, 40, 60, 80, and 100 nodes. A single source–destination pair is considered. Fig. 10 shows the RMSE results for the LA-RS and BP algorithms. The simulation results show that the RMSE linearly increases as the number of nodes increases from 20 to 100. When the average success rate α_{avg} is equal to 0.8 or 0.9, the LA-RS algorithm outperforms the BP algorithm in terms of a lower estimation error.

Finally, we investigate the performance of the LA-RS algorithm in three larger networks that were generated by BRITE. We select four source nodes for the 20-node network and 20 source nodes for the 100- and 500-node networks. The orientation algorithm is applied to obtain the directed acyclic

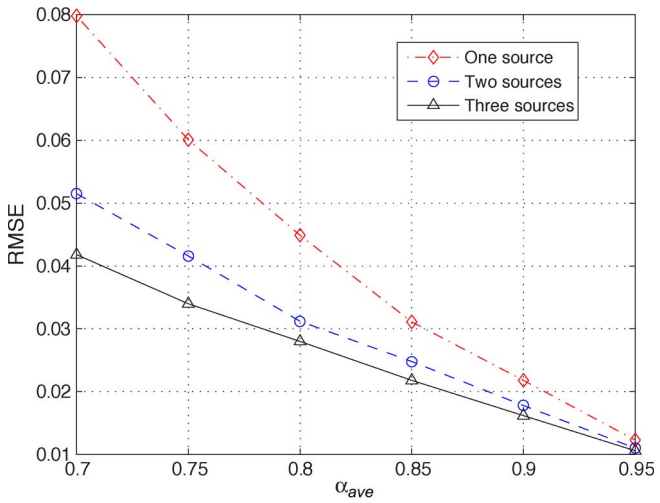


Fig. 9. RMSE of the LA-RS algorithm versus the average success rate α_{ave} for different numbers of sources ($n = 500$).

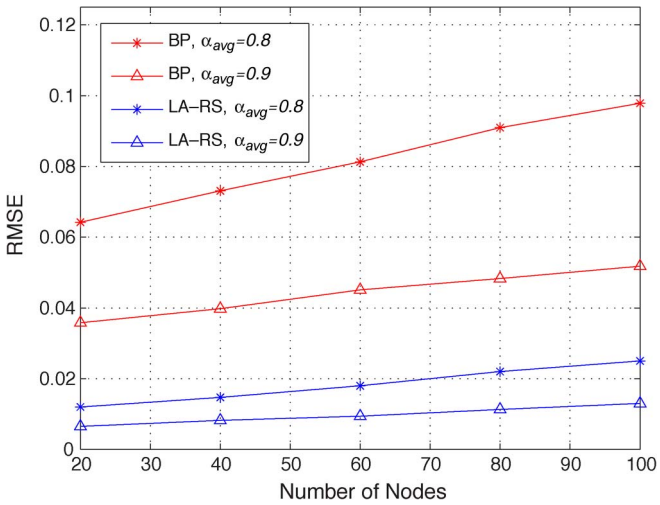


Fig. 10. RMSEs of the LA-RS and BP algorithms for varying numbers of nodes ($n = 5000$).

graphs. For the directed graphs of the 20- (with 40 links), 100- (with 200 links), and 500-node networks (with 1000 links), there are, on average, 68%, 64%, and 26% identifiable links, respectively. Although the effect of the number and location of sources on the accuracy can be negligible with relatively large success rates or sufficient probe batches, different numbers and locations of sources may result in different numbers of identifiable and virtual links. Fig. 11 shows the RMSE as a function of the number of probe batches for networks of different sizes. The LA-RS algorithm still achieves satisfactory accuracy (RMSE $< 1\%$, and $n = 20\,000$), whereas more probe batches are required to achieve the same accuracy in larger networks.

VI. Conclusion

In this paper, we have proposed a LANT framework for the active inference of link loss rates. We first determined the minimum probe size for valid end-to-end observations when network coding is applied. Then, we developed algorithms

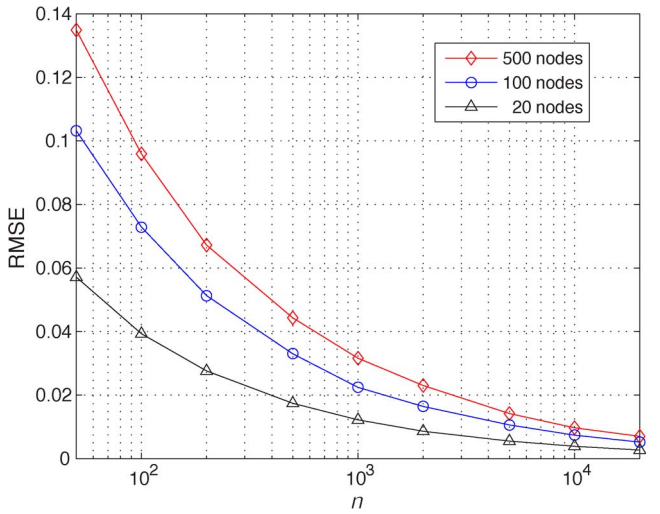


Fig. 11. RMSE of the LA-RS algorithm versus the number of probe batches n for networks of different sizes ($\alpha_{ave} = 0.9$).

to find a valid probe-coding scheme such that the minimum probe size is always achieved. Furthermore, we proposed the LA approach and developed consistent estimators of link loss rates. We also demonstrated that the complexity of LA using the method of row selection is lower than using the method of normal equations. Using our LANT framework, the identifiability of a link is the necessary and sufficient condition for its consistent loss estimation. Simulation results showed that LANT achieves better estimation accuracy than the BP algorithm when the estimators converge. Future work includes minimizing the number of nodes by performing network coding and implementing LANT in network-monitoring tools.

Appendix A
Proof of Proposition 1

We prove by contradiction. Suppose that there exists a link pair (e, e'), where e, $e' \in \mathcal{E}$, such that all paths in $\mathcal{P}$ include either both or none of the links, i.e., $\mathcal{P}(e) = \mathcal{P}(e')$. If a probe packet is dropped on either link e or e', the same end-to-end observation (probe packets with the same contents) will be obtained in either case. Therefore, we cannot diagnose on which link the loss of probe packet occurs, and it is not possible to estimate the loss rate of these links. ■

Appendix B
Proof of Theorem 1

For each end link $e \in \mathcal{E}_R$, let $\mathcal{P}(e) = \{P_1, P_2, \ldots, P_{|\mathcal{P}(e)|}\}$. With regard to valid probe-coding schemes, based on the content of the received probe, a receiver should distinguish which paths have successfully transmitted a probe and which paths have not. Without loss of generality, we start from path P_1. Because a zero binary vector will introduce ambiguity, $(1)_2$ is the smallest binary vector that we can use to denote the case where only path P_1 has successfully transmitted a probe. $(10)_2$ is the smallest binary vector that we can use to denote the case where only path P_2 has successfully transmitted a probe. Because $(11)_2$ denotes the case where both paths P_1 and P_2

have successfully transmitted a probe, $(100)_2$ is the smallest binary vector that we can use to denote the case where only path P_3 has successfully transmitted a probe. By induction, we can show that $(10\cdots0)_2$ of length $|\mathcal{P}(e)|$ is the smallest binary vector that we can use to denote the case where only path $P_{|\mathcal{P}(e)|}$ has successfully transmitted a probe. We modify the aforementioned binary vectors to vectors of length $|\mathcal{P}(e)|$, with zeros added to the left-hand side. Thus, for the probes that are transmitted in subgraph $\mathcal{G}_e$, we have $\ell_e \geq |\mathcal{P}(e)|$, and $q_e \geq 2^{|\mathcal{P}(e)|}$. ■

APPENDIX C
PROOF OF PROPOSITION 2

According to Theorem 1, it is necessary that the probes that are transmitted in each subgraph $\mathcal{G}_e$, where $e \in \mathcal{E}_R(\mathcal{G})$, have a size greater than $\ell_e \geq |\mathcal{P}(e)|$. Because network coding is applied, the probes that are transmitted on one subgraph should also have the same size. Moreover, if a link is shared between different subgraphs, the probes that are transmitted in the link should have the same size. Thus, for the probes that are transmitted in the subgraphs with overlapping links, the minimum probe size for each of the subgraphs is greater than the minimum probe size for all the subgraphs obtained in Theorem 1, i.e., $\ell_{\mathcal{G}} \geq \max_{e\in\mathcal{E}_R(\mathcal{G})} |\mathcal{P}(e)|$. ■

APPENDIX D
PROOF OF LEMMA 1

We prove by induction. We mention that the matrix $\mathcal{M}(|\mathcal{P}|)$ has binary entries and the column vectors are defined in a vector space over a finite field $\mathbb{F}_2$. It can be verified that $\mathcal{M}(2)$ has full rank. Assume that matrix $\mathcal{M}(k)$ also has full rank. That is, all 2^k-1 columns in $\mathcal{M}(k)$ are linearly independent. Thus, the modulo-2 summation of any m columns of this matrix, for $m = 2, \ldots, 2^k-1$, has at least one nonzero entry. Now, consider matrix $\mathcal{M}(k+1)$. This matrix can be represented as follows after row and column permutations:

$$\mathcal{M}(k+1) = \left[\begin{array}{c|c|ccc} 0 \cdots 0 & 1 & 1 & \cdots & 1 \\ \hline & 0 & & & \\ \mathcal{M}(k) & \vdots & & \mathcal{M}(k) & \\ & 0 & & & \\ \hline & 1 & 1 & \cdots & 1 \\ \mathcal{M}(k) & \vdots & \vdots & \ddots & \vdots \\ & 1 & 1 & \cdots & 1 \end{array}\right].$$

Permutations would not change its rank. The top row represents the newly added path, followed by two submatrices $\mathcal{M}_1 = [\mathcal{M}(k)\ \mathbf{0}\ \mathcal{M}(k)]$ and $\mathcal{M}_2 = [\mathcal{M}(k)\ \mathbf{1}]$, where $\mathbf{0}$ and $\mathbf{1}$ are columns of 0 and 1, respectively. We note that the path sets of $\mathcal{M}_1$ (rows in $\mathcal{M}_1$) do not include the newly added path, whereas the path sets of $\mathcal{M}_2$ all include it. Now, we show that the matrix $\mathcal{M}(k+1)$ has full rank. To do so, we show that the summation of all possible combinations of these $2^{k+1}-1$ columns in $\mathcal{M}(k+1)$ is a nonzero vector (i.e., there exists at least one nonzero entry in the summation vector).

First, the middle column $[1\ \mathbf{0}\ \mathbf{1}]^{\mathrm{T}}$ is included in the combination of the columns that we choose. Because the entries of the last row in $\mathcal{M}(k)$ are all ones, in the summation of the chosen vectors, at least one entry would be nonzero. This entry corresponds to the last row in $\mathcal{M}_1$ or in $\mathcal{M}_2$. Hereafter, we exclude the middle column from our choices.

Second, we choose the columns from either the 2^k-1 columns on the left-hand side or the 2^k-1 columns on the right-hand side (but not both at the same time). In this case, at least one entry in the summation vector would be nonzero, corresponding to the rows in $\mathcal{M}_1$. It is because of the linear independency of the columns in $\mathcal{M}(k)$.

Third, we choose the columns from both the 2^k-1 columns on the left- and right-hand sides. In this case, if an odd number of columns is chosen from the right-hand side, the entry of the summation vector that corresponds to the top row would be nonzero. However, if an even number of columns is chosen from the right-hand side, at least one entry of the summation vector that corresponds to the rows in $\mathcal{M}_2$ would be nonzero because of the linear independency of the columns in $\mathcal{M}(k)$. To this end, we have considered the modulo-2 summation for all possible combinations of the columns in matrix $\mathcal{M}(k+1)$, and there is always at least one nonzero entry in the summation vector. Therefore, all these $2^{k+1}-1$ column vectors in $\mathcal{M}(k+1)$ are linearly independent. ■

REFERENCES

[1] Y. Yardi, "Network tomography: Estimating source–destination traffic intensities from link data," *J. Amer. Stat. Assoc.*, vol. 91, no. 433, pp. 365–377, Mar. 1996.

[2] Y. Tsang, M. Coates, and R. Nowak, "Network delay tomography," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 51, no. 8, pp. 2125–2136, Aug. 2003.

[3] A. Chen, J. Cao, and T. Bu, "Network tomography: Identifiability and Fourier domain estimation," in *Proc. IEEE INFOCOM*, Anchorage, AK, May 2007, pp. 1875–1883.

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

[5] P. Sattari, A. Markopoulou, and C. Fragouli, "Multiple-source–multiple-destination topology inference using network coding," in *Proc. NetCod Workshop*, Lausanne, Switzerland, Jun. 2009, pp. 36–41.

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

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

[8] N. Duffield, J. Horowitz, F. LoPresti, and D. Towsley, "Multicast topology inference from measured end-to-end loss," *IEEE Trans. Inform. Theory*, vol. 48, no. 1, pp. 26–45, Jan. 2002.

[9] N. Duffield, "Network tomography of binary network performance characteristics," *IEEE Trans. Inform. Theory*, vol. 52, no. 12, pp. 5373–5388, Dec. 2006.

[10] Y. Tsang, M. Coates, and R. Nowak, "Passive unicast network tomography using EM algorithm," in *Proc. of IEEE Int. Conf. Acoust., Speech, Signal Processing*, Salt Lake City, UT, May 2001, pp. 1469–1472.

[11] J. Zhao, R. Govindan, and D. Estrin, "Computing aggregates for monitoring wireless sensor networks," in *Proc. of IEEE Int. Workshop on Sensor Network Protocols and Applications*, Anchorage, AK, May 2003, pp. 139–148.

[12] H. Nguyen and P. Thiran, "Using end-to-end data to infer lossy links in sensor networks," in *Proc. IEEE INFOCOM*, Barcelona, Spain, Apr. 2006, pp. 1–12.

[13] Y. Lin, B. Liang, and B. Li, "Passive loss inference in wireless sensor networks based on network coding," in *Proc. IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009, pp. 1809–1817.

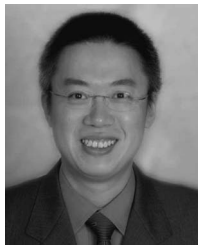
[14] M. Coates and R. Nowak, "Network loss inference using unicast end-to-end measurement," in *Proc. ITC Semin. IP Traffic, Meas. Modell.*, Monterey, CA, Sep. 2000, pp. 28-1–28-9.
[15] W. Zhu and Z. Geng, "A bottom–up inference of loss rate," *Comput. Commun.*, vol. 28, no. 4, pp. 351–365, Mar. 2005.
[16] H. Su, W. Chen, S. Lin, D. Jin, and L. Zeng, "The inference of link loss rates with internal monitors," in *Proc. IEEE GLOBECOM*, New Orleans, LA, Dec. 2008, pp. 1–6.
[17] T. Bu, N. Duffield, F. LoPresti, and D. Towsley, "Network tomography on general topologies," in *Proc. ACM SIGMETRICS*, Marina Del Rey, CA, Jun. 2002, pp. 21–30.
[18] R. Ahlswede, N. Cai, S. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.
[19] C. Fragouli, J. LeBoudec, and J. Widmer, "Network coding: An instant primer," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 1, pp. 63–68, Jan. 2006.
[20] C. Fragouli and A. Markopoulou, "A network coding approach to network tomography," in *Proc. 43rd Allerton Conf.*, Monticello, IL, Sep. 2005.
[21] C. Fragouli, A. Markopoulou, R. Srinivasan, and S. Diggavi, "Network monitoring: It depends on your points of view," in *Proc. Inf. Theory Appl. Workshop*, San Diego, CA, Jan. 2007.
[22] M. Gjoka, C. Fragouli, P. Sattari, and A. Markopoulou, "Loss tomography in general topologies with network coding," in *Proc. IEEE GLOBECOM*, Washington, DC, Nov. 2007, pp. 381–386.
[23] H. Yao, S. Jaggi, and M. Chen, "Network coding tomography for network failures," in *Proc. IEEE INFOCOM*, San Diego, CA, Mar. 2010, pp. 1–5.
[24] Y. Chen, D. Bindel, H. Song, and R. Katz, "Algebra-based scalable overlay network monitoring: Algorithms, evaluation, and applications," *IEEE/ACM Trans. Netw.*, vol. 15, no. 5, pp. 1084–1097, Oct. 2007.
[25] 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, Apr. 2005.
[26] B. Sun and Z. Zhang, "Probabilistic diagnosis of link loss using end-to-end path measurements and maximum likelihood estimation," in *Proc. IEEE ICC*, Dresden, Germany, Jun. 2009, pp. 1–5.
[27] R. Koetter and M. Medard, "Beyond routing: An algebraic approach to network coding," in *Proc. of IEEE INFOCOM*, New York, Nov. 2002, pp. 122–130.
[28] R. Myers, D. Montgomery, and G. Vining, *Generalized Linear Models: With Applications in Engineering and the Sciences*. Hoboken, NJ: Wiley, 2001.
[29] G. Grimmett and D. Stirzaker, *Probability and Random Processes*, 3rd ed. London, U.K.: Oxford Univ. Press, 2001.
[30] G. Golub and C. Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: The Johns Hopkins Univ. Press, 1996.
[31] G. W. Stewart, *Matrix Algorithms: Basic Decompositions*. Philadelphia, PA: SIAM, 1998.
[32] Internet2 network. [Online]. Available: http://www.Internet2.edu/network/
[33] BRITE. [Online]. Available: www.cs.bu.edu/brite/

Jiaqi Gui received the B.E. degree in information engineering from Shanghai Jiao Tong University, Shanghai, China, in 2008 and the M.A.Sc. degree in electrical and computer engineering from the University of British Columbia, Vancouver, BC, Canada, in 2010.

Vahid Shah-Mansouri (S'02) received the B.Sc. and M.Sc. degrees in electrical engineering from the University of Tehran, Tehran, Iran, in 2003 and the Sharif University of Technology, Tehran, in 2005, respectively. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of British Columbia (UBC), Vancouver, BC, Canada.

From 2005 to 2006, he was with the Farineh-Fanavar Company, Tehran. His research interests include the design and mathematical modeling of radio-frequency identification systems and wireless networks.

Mr. Shah-Mansouri received the UBC Four-Year Fellowship and the UBC Faculty of Applied Science Award.



Vincent W. S. Wong (SM'07) received the B.Sc. degree from the University of Manitoba, Winnipeg, MB, Canada, in 1994, the M.A.Sc. degree from the University of Waterloo, Waterloo, ON, Canada, in 1996, and the Ph.D. degree from the University of British Columbia (UBC), Vancouver, BC, Canada, in 2000.

From 2000 to 2001, he was a Systems Engineer with PMC-Sierra Inc. In 2002, he joined the Department of Electrical and Computer Engineering, UBC, and is currently an Associate Professor. His research interests include the protocol design, optimization, and resource management of communication networks, with applications to the Internet, wireless networks, smart grid, radio-frequency identification systems, and intelligent transportation systems.

Dr. Wong is an Associate Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and an Editor of the KICS/IEEE JOURNAL OF COMMUNICATIONS AND NETWORKS. He is a Symposium Co-chair of the 2011 IEEE Global Communications Conference (Globecom) Wireless Communications Symposium. He serves as a Member of Technical Program Committee of various conferences, including the IEEE International Conference on Computer Communications (Infocom) and the IEEE International Conference on Communications (ICC).