# ACCURATE AND EFFICIENT NETWORK MONITORING ON MESH TOPOLOGIES VIA NETWORK CODING

by

JIAQI GUI

B.E., Information Engineering, Shanghai Jiao Tong University, China, 2008

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA
(Vancouver)

June 2010

# Abstract

Accurate and efficient measurement of network-internal characteristics is critical for management and maintenance of large-scale networks. In this thesis, we propose a *linear algebraic network tomography* (LANT) framework for active inference of link loss rates on mesh topologies via 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 pre-determined 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 establish a tight lower bound on probe size which is necessary for establishing the mappings 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 lower bound on probe size is achieved. Furthermore, we propose a linear algebraic approach to developing consistent estimators of link loss rates, which converge to the actual loss rates as the number of probes increases. We show that using the LANT framework, 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. Simulation results show

that the LANT framework achieves better estimation accuracy than the belief propagation (BP) algorithm for large number of probe packets.

# Contents

# List of Figures

# List of Acronyms

| | |
|---|---|
| QoS | Quality of Service |
| MVWA | Minimum Variance Weighted Average |
| EM | Expectation-Maximization |
| LEND | Least-Biased End-to-end Network Diagnosis |
| MILS | Minimal Identifiable Link Sequence |
| LPRs | Link Pass Ratios |
| PPRs | Path Pass Ratios |
| LA | Linear Algebra |
| LANT | Linear Algebraic Network Tomography |
| BP | Belief Propagation |
| ML | Maximum Likelihood |
| RMSE | Root Mean Square Error |

| | |
|---|---|
| NE | Normal Equations |
| RS | Row Selection |

# List of Symbols

| | |
|---|---|
| $\lvert \cdot \rvert$ | Cardinality of a set |
| $n$ | Number of probe batches |
| $\mathcal{V}$ | Set of nodes |
| $\mathcal{S}$ | Set of source nodes |
| $\mathcal{R}$ | Set of receiver nodes |
| $\mathcal{E}$ | Set of links |
| $\mathcal{E}_I$ | Set of identifiable links |
| $\mathcal{E}_V$ | Set of virtual links |
| $\mathcal{E}_R$ | Set of end links |
| $\mathcal{P}$ | Set of monitored end-to-end paths |
| $\mathcal{P}(e)$ | Set of end-to-end paths that include link $e$ |
| $\mathscr{P}$ | Power set of $\mathcal{P}$, $\lvert\mathscr{P}\rvert = 2^{\lvert\mathcal{P}\rvert}$ |

| | |
|---|---|
| $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ | Directed acyclic graph |
| $\mathcal{G}_e$ | Subgraph consisting of the links and nodes in set $\mathcal{P}(e)$ |
| $\mathscr{G}$ | Set of subgraphs with overlapping links |
| $\mathcal{T}_e$ | Auxiliary tree corresponding to end link $e$ and subgraph $\mathcal{G}_e$ |
| $\mathcal{L}_e$ | Set of leaf nodes in $\mathcal{T}_e$ |
| $u_i(v)$ | The $i$th tree node that corresponds to the non-receiver node $v$ in the directed graph |
| $\varepsilon_i(e)$ | The $i$th tree link that corresponds to link $e$ in the directed graph |
| $\ell$ | Probe size, number of bits in each probe packet |
| $q = 2^\ell$ | Alphabet size of a probe packet |
| $\mathbf{M} = (m_{i,j})_{\lvert\mathcal{P}\rvert\times\lvert\mathcal{E}\rvert}$ | Path-link matrix |
| $\overline{\mathbf{M}} = (\overline{m}_{i,j})_{\lvert\mathcal{P}\rvert\times\lvert\mathcal{E}_I\cup\mathcal{E}_V\rvert}$ | Type 1 modified path-link matrix |
| $\widetilde{\mathbf{M}} = (\widetilde{m}_{i,j})_{(\lvert\mathscr{P}\rvert-1)\times\lvert\mathcal{E}_I\cup\mathcal{E}_V\rvert}$ | Type 2 modified path-link matrix |
| $\mathcal{M}(\lvert\mathcal{P}\rvert)$ | Auxiliary matrix of type 2 modified path-link matrix with path set $\mathcal{P}$ |
| $\alpha_j$ | Actual link success rate of $j$th link in $\mathcal{E}_I \cup \mathcal{E}_V$ |

| | |
|---|---|
| $\beta_i$ | Actual path success rate of $i$th path in $\mathcal{P}$ |
| $\theta_i$ | Actual path set success rate of $i$th path set in $\mathscr{P} \setminus \{\varnothing\}$ |
| $\mathbf{a} = (a_j)_{\lvert\mathcal{E}_I \cup \mathcal{E}_V\rvert \times 1}$ | Column vector, where $a_j = \log \alpha_j$ |
| $\mathbf{b} = (b_i)_{\lvert\mathcal{P}\rvert \times 1}$ | Column vector, where $b_i = \log \beta_i$ |
| $\mathbf{c} = (c_i)_{(\lvert\mathscr{P}\rvert - 1) \times 1}$ | Column vector, where $c_i = \log \theta_i$ |
| $\mu = \lvert\mathscr{P}\rvert - 1$ | Number of rows in $\widetilde{\mathbf{M}}$ |
| $\nu = \lvert\mathcal{E}_I \cup \mathcal{E}_V\rvert$ | Number of columns in $\widetilde{\mathbf{M}}$ |
| $\widetilde{\mathbf{M}}_1$ | Reduced $\nu \times \nu$ path-link matrix from $\widetilde{\mathbf{M}}$ after row selection |
| $\mathbf{c}_1$ | Reduced $\nu \times 1$ column vector from $\mathbf{c}$ after row selection |

# Acknowledgements

First, I would like to express my deepest gratitude to my supervisor, Dr. Vincent Wong, for his patient guidance, constant encouragement, and excellent advice throughout my graduate study. Through our weekly meeting, we generate new ideas, stimulate creativity and dig into technology details. Without his invaluable help, this work would not be possible.

A special thanks goes to my colleague, Vahid Shah-Mansouri, who provided me with precious assistance during the completion of this thesis. I am also thankful to all my colleagues in Dr. Wong's group: Amir Hamed Mohsenian-Rad, Vahid Shah-Mansouri, Man Hon Cheung, Keivan Ronasi, Ehsan Vahedi, Wei Bao, Xiaolei Hao, Wenbo Shi, and Jinbiao Xu as well as other friends in the data communications group, for sharing their experiences and knowledge during the time of my study.

Finally, I take this opportunity to express my profound gratitude to my beloved parents for their understanding, support, and endless love during my study in Canada. To them I dedicate this thesis.

This research is supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada under grant number STPSC 356767.

# Chapter 1

# Introduction

Accurate and efficient measurement of network-internal characteristics is critical for management and maintenance of large-scale networks. With accurate and timely performance estimates, more efficient traffic control protocols and dynamic routing algorithms can be designed. Quality-of-service (QoS) guarantees can be achieved if the available bandwidth can be gauged; the resulting service level agreements can be verified. Detecting anomalous or malicious behavior becomes a more achievable task [1].

Although network administrators can monitor local traffic conditions and detect congestion points in small-scale networks, most networks are not completely isolated. The user-perceived performance of a network thus depends heavily on the performance of the internetwork. The traditional approach for characterizing network performance is based on detailed queueing models at the individual router level, which requires access to a wide range of routers to obtain link-level statistics. However, the routers are operated by different companies or service providers, which makes it difficult to collect detailed information at individual devices. Alternatively, we can make useful end-to-end measurements that do not require widely cooperation from internal network devices. Subsequently, based on the end-to-end measurements, we can apply inference techniques to extract the hidden

information of interest.

Broadly speaking, large-scale network inference involves estimating network performance parameters based on traffic measurements at a limited subset of the nodes. Vardi was one of the first researchers to rigorously study this type of problem and he coined the term *network tomography* [2] due to the similarity between the inference of network characteristics and medical tomography. Three forms of network tomography have been addressed in the recent literature: (1) link-level parameter estimation based on end-to-end, path-level traffic measurements [3, 4], (2) sender-receiver path-level traffic intensity estimation based on link-level traffic measurements, and (3) the inference of network topology [5, 6]. Characterizing these parameters is critical for detecting congestion, faults and other anomalous behavior, ensuring compliance of service-level agreements with users, and management of overlay networks.

In link-level parameter estimation, the end-to-end measurements usually consist of the number of probe or data packets transmitted and received between the source and the receiver nodes or the delay between packet transmissions and receptions. The objective is to estimate the loss rate or the queuing delay of each link. Dropped packets on a link are usually due to overload of the finite output buffer of one of the routers encountered when traversing the link, but may also be caused by equipment downtime due to maintenance or power failures. The end-to-end delay is due to both propagation delay processing delay, queuing delay, and transmission delay. As assumed by most literature, occurrences of dropped packets and queuing delay is inherently random.

In path-level traffic intensity estimation, the measurements consist of the number of probe or data packets that transmitted through nodes in the network. In privately owned networks, the collection of such measurements is relatively straightforward. Based on these measurements, the goal is to estimate how much traffic originated from a specified node and was transmitted to a specified destination. The combination of the traffic intensities of all the origin-destination pairs forms the origin-destination traffic matrix. Both the node-level measurements and the parameter to be estimated are inherently random.

In the inference of network topology, the measurements usually consist of the number of probe or data packets transmitted and received between the source and the receiver nodes or the delay between packet transmissions and receptions. Some proposals require clock synchronization while other more practical ones do not. The physical network topology can be represented as a directed graph, where each vertex represents a physical device such as a router or a switch and the edges correspond to the communication links between those devices. Based on the end-to-end measurements, the logical topology can be determined.

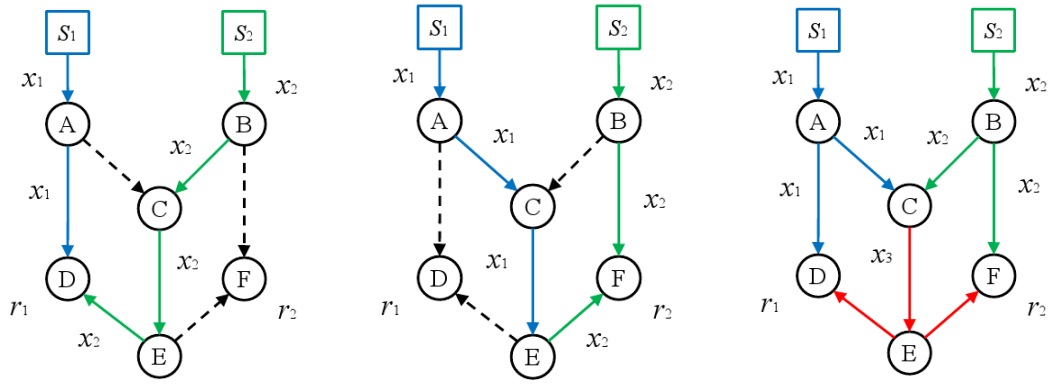
Network tomography can be performed either in an *active* or *passive* manner. Active network tomography refers to the case where probe packets are sent from the sources to the receivers located on the periphery of the network [7–9]. Using the end-to-end measurements generated by probe packets transmitted and received between the source and the receiver nodes, more informative and reliable path-level measurements are provided

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]. There is also an accelerating trend toward network security that will create a highly uncooperative environment for active tomography. For example, firewalls designed to protect information may not allow requests for routing information, special packet handling and other network transport protocols required by many active tomography techniques. This has prompted investigations into passive-based traffic monitoring techniques.

## 1.1 Network Coding Basics

Networked systems arise in various communication contexts such as telephone networks, the public Internet, peer-to-peer networks, ad-hoc wireless networks, and wireless sensor networks. An inherent premise behind the operation of all communication networks lies in the way information is treated. Recently, with the advent of network coding, the simple but important observation was made that in communication networks, intermediate nodes are allowed not only forward but also process the incoming independent information flows [14–16]. At the network layer, for example, intermediate nodes can perform binary addition of independent bitstreams, whereas, at the physical layer of optical networks, intermediate nodes can superimpose incoming optical signals. In other words, data streams that are independently produced and consumed do not necessarily need to




(a) Routing to $r_1$ (b) Routing to $r_2$ (c) Network Coding

**Figure 1.1:** The Butterfly network. Sources $s_1$ and $s_2$ multicast information $x_1$ and $x_2$ to receivers $r_1$ and $r_2$.

be kept separate when they are transported throughout the network. Combining independent data streams can better tailor the information flow to the network environment and accommodate the demands of specific traffic patterns. This shift in paradigm is expected to revolutionize the way we manage, operate, and understand the organization in networks, as well as to have a deep impact on a wide range of areas such as reliable delivery, resource sharing, efficient flow control, network monitoring, and security [17].

One essential benefit of network coding is in terms of throughput when multicasting. The following simple example from [14] illustrates the basic concepts in network coding and gives a preliminary idea of the expected benefits and challenges.

**Example 1:** Fig. 1.1 depicts a communication network represented as a directed graph where vertices correspond to terminals and edges correspond to channels. This

example is commonly known in the network coding literature as the butterfly network. Assume we can send one bit per time slot through each channel. We have two sources $s_1$ and $s_2$, and two receivers $r_1$ and $r_2$. Each source produces one bit per time slot which we denote by $x_1$ and $x_2$, respectively. If receiver $r_1$ uses all the network resources by itself, it could receive information from both sources. Indeed, we could route the bit $x_1$ from source $s_1$ along the path $\{AD\}$ and the bit $x_2$ from source $s_2$ along the path $\{BC, CE, ED\}$, as depicted in Fig. 1.1(a). Similarly, if the second receiver $r_2$ uses all the network resources by itself, it could also receive information from both sources. Indeed, we could route the bit $x_1$ from source $s_1$ along the path $\{AC, CE, EF\}$, and the bit $x_2$ from source $s_2$ along the path $\{BF\}$ as depicted in Fig. 1.1(b).

Now assume that both receivers want to simultaneously receive the information from both sources. We then have a contention for the use of edge $CE$, since we assume that each channel can only transmit one bit per time slot. Traditionally, information flow was treated like fluid through pipes, and independent information flows were kept separate. The simple but important observation made in the work by Ahlswede *et al.* is that we can allow intermediate nodes in the network to process their incoming information flows, and not just forward them. In Fig. 1.1(c), node $C$ can take bits $x_1$ and $x_2$ and XOR them to create a third bit $x_3 = x_1 + x_2$ which it can then send through edge $CE$ (the XOR operation corresponds to addition over the binary field). $r_1$ receives $\{x_1, x_1 + x_2\}$, and can solve this system of equations to retrieve $x_2$ by XORing $x_1$ and $x_1 + x_2$. Similarly, $r_2$ receives $\{x_2, x_1 + x_2\}$, and can solve this system of equations to retrieve $x_1$ by XORing

$x_2$ and $x_1 + x_2$. □

The previous example shows that if we allow intermediate node in the network to combine information streams and extract the information at the receivers, we can increase the throughput when multicasting.

Network coding can also be used to infer the loss rates of links in an overlay network. For conventional active tomography, packets are usually multicast to several receivers. After a sufficiently large number of probe packets, shared links and their loss rates can be inferred with reasonable accuracy. In such a setting, network coding can provide additional flexibility since probe packets are not only duplicated at branching points of the multicast tree, but may also be merged. If multiple senders transmit packets to a single receiver, and these packets are combined within the network, it allows to infer network parameters in much the same way as multicasting from one sender to multiple receivers [18]. Furthermore, if the network coding coefficients (i.e., the specific way in which packets are combined at the nodes) are known in advance, the received coded probe packets can provide additional information about which packets were lost in which part of the tree. By exploiting these features, it is possible to significantly reduce the number of active probes and the bandwidth usage generated by these probes, and thus increase bandwidth efficiency.

## 1.2 Related Work

In the area of loss tomography, extensive studies have been given to multicast tree topologies. In [7], Caceres *et al.* developed a maximum-likelihood estimator for loss rates on internal links based on losses observed by multicast receivers. It exploits the inherent correlation between such observations to infer the performance of paths between branch points in the tree spanning a multicast source and its receivers. The proposed method relies on the iterative approximation to identify the parameters that requires a long execution time. In addition, the parameters identified by this method may not be the true values of those parameters since the iterative procedure may trap into a local maximum. In [19], Zhu *et al.* proposed an estimate that is based on the correlation between a link and its sibling links to identify the loss rate of the link. The proposed method, instead of using an iterative approach to approximate the maximum, employs a bottom-up approach to identify the loss rates of the links of a network.

In contrast to multicast techniques, unicast inference based on multicast tree topologies can easily be performed on most networks. In [20], Duffield *et al.* designed experiments based on the notion of transmitting stripes of packets (with no delay between transmission of successive packets within a stripe) to two or more receivers. The purpose of these stripes is to ensure that the correlation in receiver observations matches as closely as possible what would have been observed if the stripe had been replaced by a multicast probe that followed the same paths to the receivers. In [10], Tsang *et al.* designed a measurement procedure for network loss inference based on end-to-end packet

pair measurements. Back-to-back packet pairs are the two packets that are sent one after the other by the source, possible destined for different receivers, but sharing a common set of links in their paths.

In [21], Padmanabhan *et al.* investigated the problem of identifying lossy links in the interior of the Internet by passively observing the end-to-end performance of existing traffic between a server and its clients. The key advantage of a passive approach is that it does not introduce additional traffic which might perturb the object of inference, i.e., the link loss rates. The techniques depend only on knowing the number of lost and successful packets sent to each client. While the accuracy of link loss rate inference may consequently suffer, the techniques can still pinpoint the trouble spots in the network (e.g., highly lossy links). They developed and evaluated three techniques for passive network tomography: random sampling, linear optimization, and Bayesian inference using Gibbs sampling.

To extend the existing multicast and unicast tomography approaches to general topologies, in [22], Bu *et al.* proposed an approach using multiple trees to cover a mesh topology and combine the inferred loss rates. They further proposed two algorithms to perform the link-level inferences. One, the minimum variance weighted average (MVWA) algorithm treats the trees separately and then average the results. The second, based on expectation-maximization (EM) merges all of the measurements into one computation. However, this approach may have *low bandwidth efficiency*, since those 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*, since it requires a large number of receivers to be deployed in each multicast tree.

The pioneering work in [14] showed that for multicast networks, if intermediate nodes can perform network coding, that is to perform simple local XOR-operations on incoming packets, one can achieve the min-cut throughput of the network to each receiver. Recent studies show that applying network coding in loss tomography can increase bandwidth efficiency [17, 23]. In a network which is capable of performing network coding in addition to multicast, the intermediate nodes linearly combine incoming probe packets and forward the coded probe packets to the outgoing links according to pre-determined coding coefficients. Results in [18] show that for active monitoring using network coding, appropriate selection of the number and location of sources and receivers can affect the accuracy of estimation in general tree topologies. The work in [24] 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. An example is illustrated in [18] such that each link in a mesh topology can be traversed in each time slot by exactly one probe.

In contrast to network coding with pre-determined coding coefficients, randomized network coding changes the fundamental connection between path and link loss probabilities such that new inference algorithms need to be developed. In [13], Lin *et al.* studied the loss inference problem in sensor networks with randomized network coding. As end-to-end data are not sufficient to compute link loss rates precisely, they proposed

inference algorithms based on Bayesian principles to discover the set of highly lossy links in sensor networks. The algorithms achieve high detection and low false-positive rates in extensive simulations. In [25], Yao *et al.* 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 setting 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. In [26], Mao *et al.* proposed a brief propagation (BP) algorithm, which is combined with the use of network coding in [24]. The BP algorithm is a low complexity algorithmic framework for link loss monitoring based on the recent modeling and computational methodology of factor graphs [27]. It iteratively updates the estimates of link losses upon receiving (or detecting the loss of) recently sent packets. The algorithm exhibits good performance and scalability, and can be easily adapted to different statistical models of networking scenarios. In particular, due to its low complexity, the algorithm is particularly suitable as a long-term monitoring facility.

In [28], Zhao *et al.* proposed a least-biased end-to-end network diagnosis (LEND) system for inferring link-level properties like loss rate. They defined a minimal identifiable link sequence (MILS) as a link sequence of minimal length whose properties can be uniquely identified from end-to-end measurements. They designed efficient algorithms to find all the MILSs and infer their loss rates for diagnosis. The LEND system works for any network topology and for both directed and undirected topologies. It gives highly

accurate estimates of the loss rates of MILSs and such diagnosis can be achieved with fine granularity and in near real-time even for reasonably large overlay networks. The LEND system can also supplement existing statistical inference approaches and provide smooth tradeoff between diagnosis accuracy and granularity.

In [29], Sun *et al.* focused on the problem of finding the link pass ratios (LPRs) when the path pass ratios (PPRs) of a set of paths are given. They proved the problem of finding the maximum-likelihood estimation of LPRs given PPRs is NP-hard, and then proposed a simple algorithm based on divide-and-conquer. It first estimates the number of faulty links on a path, then uses the global information to estimate assign LPRs to the links. It requires a priori probability distribution function on link loss rates and an assumption that the majority of links being lossless.

In [30], Chen *et al.* focused on overlay network monitoring, which enables distributed Internet applications to detect and recover from path outages and periods of degraded performance within seconds. For an overlay network with $n$ hosts, existing systems either require $\mathcal{O}(n^2)$ measurements, and thus lack scalability, or can only estimate the latency but not congestion or failures. They proposed an algebraic approach that selectively monitors $k$ linearly independent paths that can fully describe all the $\mathcal{O}(n^2)$ paths. The loss rates and latency of these $k$ paths can be used to estimate the loss rates and latency of all other paths.

## 1.3 Motivations and Objective

In this thesis, 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, 19, 31, 32], loss tomography on mesh topologies is still a challenging problem. Existing approaches have not exploited the inherent information in the end-to-end observations. As a result, the linear system of link loss rates and path loss rates usually has a coefficient matrix with deficient column rank[1], which makes it difficult to accurately infer the link loss rates [30].

In general, most of the previously proposed loss tomography approaches on mesh topologies in the literature have one or more of the following performance bottlenecks: (1) low bandwidth efficiency, (2) high monitoring cost, (3) estimation not being always accurate, and (4) requiring additional assumptions. In this thesis, we propose a *linear algebraic network tomography* (LANT) framework for 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 those on the different sets of paths, which is captured through network coding. We refer to probe packets and network coding schemes jointly

[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 full column rank; otherwise, the matrix has deficient column rank.

as *probe coding schemes*. In our LANT framework, a valid probe coding scheme enables us to establish the mappings 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. Since the number of all path sets can grow exponentially as the number of total 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.

## 1.4 Contributions

The main contributions of this thesis are as follows [33, 34]:

- We establish a tight lower bound on probe size, which is necessary for valid probe coding schemes when network coding is applied. Then, we develop algorithms to find a valid probe coding scheme such that the lower bound on probe size is achieved.

- We propose a linear algebraic (LA) approach to developing 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. Besides, the LA approach achieves better estimation accuracy than the BP algorithm, when the estimators converge. 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 number and location of sources may result in different number of identifiable links.

The framework we present in this thesis is unique when compared to the prior work done in the area of loss tomography using network coding. In terms of bandwidth efficiency, the work in [24] establishes a loose lower bound on probe size for valid probe coding schemes, while the problem of finding coding coefficients remains unexplored. Without efficient algorithms to find coding coefficients, the inference framework is incomplete and cannot be applied. We establish a tight lower bound and also develop algorithms to find a valid probe coding scheme such that the lower bound on probe size is achieved. In terms of inference approaches, the BP algorithm [26] 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 [29] requires additional assumptions such as a priori probability distribution function and the majority of links being lossless. The work in [28] only find the loss rate of a minimal identifiable link sequence. In contrast, our LA approach does not need extra assumptions while it can still obtain additional useful information, the losses on the different sets of paths. This information can only be obtained via probe coding schemes and cannot be achieved by routing probes in general. As a result, we obtain better estimation accuracy and obtain more identifiable links.

## 1.5 Structure of the Thesis

The rest of this thesis is organized as follows. In Chapter 2, we present the LANT framework, including system model, probe coding schemes, and an LA approach for the consistent estimation of link loss rates. Chapter 3 presents performance evaluation. Conclusions are given in Chapter 4.

# Chapter 2

# The LANT Framework

In this chapter, we present we present the LANT framework, including system model, probe coding schemes, and an LA approach for the consistent estimation of link loss rates.

## 2.1 System Model

We model the network as a directed acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consisting 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 $i$th path in set $\mathcal{P}$ includes the $j$th link in set $\mathcal{E}$, and is equal to 0 otherwise. As an example, the directed acyclic graph in Fig. 2.1 has

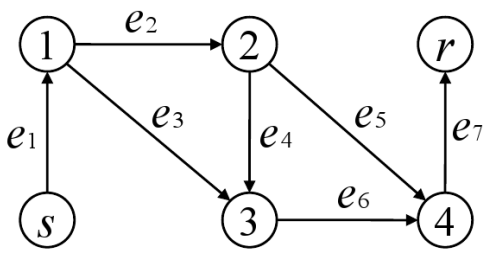



**Figure 2.1:** A directed acyclic graph with $\mathcal{V} = \{s, r, 1, 2, 3, 4\}$ and $\mathcal{E} = \{e_1, e_2, \ldots, e_7\}$. The set of monitored end-to-end paths $\mathcal{P} = \{P_1, P_2, P_3\}$, where $P_1 = \{e_1, e_2, e_5, e_7\}$, $P_2 = \{e_1, e_2, e_4, e_6, e_7\}$, and $P_3 = \{e_1, e_3, e_6, e_7\}$. For link $e_2 = (1, 2)$, we have $\mathcal{P}(e_2) = \{P_1, P_2\}$.

three paths from source $s$ to receiver $r$. Its path-link matrix is a 3 by 7 binary matrix as shown below:

$$\mathbf{M} = \begin{array}{c} \\ P_1 \\ P_2 \\ P_3 \end{array} \begin{array}{c} \begin{array}{ccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \end{array}. \tag{2.1}$$

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 in Fig. 2.1, links $e_2, e_3, \ldots, e_6$ are identifiable links, while links $e_1$ and $e_7$ are non-identifiable links since $\mathcal{P}(e_1) = \mathcal{P}(e_7)$. 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 condi-

tion 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.*

*Proof*: We prove it by contradiction. Suppose 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 them, i.e., $\mathcal{P}(e) = \mathcal{P}(e')$. If a probe packet is being 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. ■

We divide the set of non-identifiable 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 in Fig. 2.1, since $\mathcal{P}(e_1) = \mathcal{P}(e_7)$, 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 $\mathcal{E}_I = \{e_{v_1}\}$ and $\mathcal{E}_V = \{e_2, e_3, \ldots, e_6\}$. 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 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 $i$th path in set $\mathcal{P}$ includes the $j$th link in set $\mathcal{E}_I \cup \mathcal{E}_V$, and is equal to 0 otherwise. We refer to $\overline{\mathbf{M}}$ as type 1 modified path-link matrix. The type 1 modified path-link matrix for the graph in Fig.

2.1 is shown below:

$$\overline{\mathbf{M}} = \begin{array}{c} \\ P_1 \\ P_2 \\ P_3 \end{array} \begin{array}{c} \begin{array}{cccccc} e_{v_1} & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array}. \tag{2.2}$$

We model the loss of packets on different links by a set of mutually independent Bernoulli processes. Losses are therefore spatial and temporal independent. This model is commonly used in the literature [7–9, 19, 31, 32] for network tomography. We define $\alpha_j \in (0, 1]$ as the *link success rate* of the $j$th link in set $\mathcal{E}_I \cup \mathcal{E}_V$, which is the probability that a packet can be successfully transmitted on the $j$th link. Thus, $1 - \alpha_j$ denotes the loss rate of the $j$th 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 $i$th path in set $\mathcal{P}$, which is the probability that a probe packet can be successfully transmitted on the $i$th path in set $\mathcal{P}$.

Unlike data packets, probe packets would not be retransmitted if being dropped. Thus, we have

$$\prod_{j=1}^{|\mathcal{E}_I \cup \mathcal{E}_V|} (\alpha_j)^{\overline{m}_{i,j}} = \beta_i, \quad i = 1, \ldots, |\mathcal{P}|. \tag{2.3}$$

Taking logarithm on both sides of (2.3), we can reformulate it as linear equations:

$$\sum_{j=1}^{|\mathcal{E}_I \cup \mathcal{E}_V|} \overline{m}_{i,j} \log \alpha_j = \log \beta_i, \quad i = 1, \ldots, |\mathcal{P}|, \tag{2.4}$$

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

$$\sum_{j=1}^{|\mathcal{E}_I \cup \mathcal{E}_V|} \overline{m}_{i,j} a_j = b_i, \quad i = 1, \ldots, |\mathcal{P}|. \tag{2.5}$$

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

$$\overline{\mathbf{M}}\mathbf{a} = \mathbf{b}. \tag{2.6}$$

Equation (2.6) 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 $\hat{\mathbf{b}}$ while $\hat{\mathbf{a}}$ remains unknown. Thus, equation (2.6) becomes a system of $|\mathcal{P}|$ equations with $|\mathcal{E}_I \cup \mathcal{E}_V|$ unknowns as:

$$\overline{\mathbf{M}}\hat{\mathbf{a}} = \hat{\mathbf{b}}. \tag{2.7}$$

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, (2.7) is under-determined. 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 $\mathcal{P}$. $n$ 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 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 details.

## 2.2 Probe Coding Schemes

In Subsection 2.2.1, we establish a tight lower bound on probe 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 in Subsection 2.2.2.

### 2.2.1 Lower Bound on Probe Size

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 [15] that are sufficient for our task.

A probe packet is a binary vector $(\cdot)_2$ of length $\ell$, 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 $\ell$ is desired to be as small as possible, since 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. 2.1, receiver $r$ receives coded packets that are combined from packets on three different paths. Using one-bit probe packets, we are not able to distinguish which of these three paths have successfully transmitted a probe packet. In this case, we need probe packets with at least three bits for valid probe coding schemes while smaller probe sizes cannot constitute valid probe coding schemes.

Before we find a lower bound on probe size, which is necessary 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 $\mathcal{G}$ consisting 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 $\ell_e$ denote the alphabet size and the length of the probe packets transmitted on subgraph $\mathcal{G}_e$, respectively. The following theorem presents a loose lower bound on probe packet size for valid probe coding schemes.

**Theorem 1** *For the probes 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., $q_e \geq 2^{|\mathcal{P}(e)|}$), in order to obtain valid end-to-end observations.*

*Proof*: For each end link $e \in \mathcal{E}_R$, let $\mathcal{P}(e) = \{P_1, P_2, \ldots, P_{|\mathcal{P}(e)|}\}$. As for 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 $P_1$. Since a zero binary vector will introduce

ambiguity, $(1)_2$ is the smallest binary vector we can use to denote the case where only path $P_1$ has successfully transmitted a probe. $(10)_2$ is the smallest binary vector we can use to denote the case where only path $P_2$ has successfully transmitted a probe. Since $(11)_2$ denotes the case where both paths $P_1$ and $P_2$ have successfully transmitted a probe, $(100)_2$ is the smallest binary vector 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 we can use to denote the case where only path $P_{|\mathcal{P}(e)|}$ has successfully transmitted a probe. We modify the above binary vectors to vectors of length $|\mathcal{P}(e)|$ with zeros added to the left-hand side. Thus, for the probes transmitted in subgraph $\mathcal{G}_e$, we have $\ell_e \geq |\mathcal{P}(e)|$, and $q_e \geq 2^{|\mathcal{P}(e)|}$. ■

Although Theorem 1 provides lower bounds on probe size for the probes transmitted on different subgraphs separately, some lower bounds may not be achieved. For example, in Fig. 2.2, 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 $\mathcal{G}_{e_7}$ such as links $e_1$, $e_2$ and $e_3$. In this case, a valid probe size should be 4. Let $\mathscr{G}$ denote a set of subgraphs with overlapping links. The probes transmitted on these subgraphs should have the same size. Let $\ell_{\mathscr{G}}$ denote the size of such probes. Correspondingly, the set of end links in the subgraph set $\mathscr{G}$ is denoted by $\mathcal{E}_R(\mathscr{G})$. The following proposition presents an improved lower bound on probe size for valid probe coding schemes.

**Proposition 2** *For the probes transmitted on subgraph set $\mathscr{G}$, the probe size should satisfy $\ell_{\mathscr{G}} \geq \max_{e\in\mathcal{E}_R(\mathscr{G})} |\mathcal{P}(e)|$, in order to obtain valid end-to-end observations.*

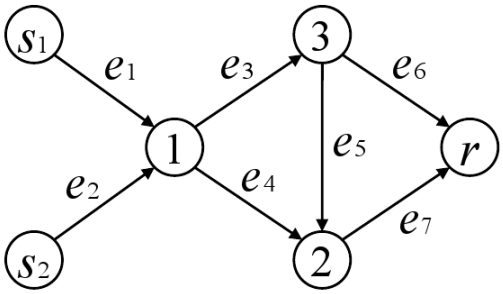



**Figure 2.2:** A directed acyclic graph with two end links $e_6$ and $e_7$.

*Proof*: According to Theorem 1, for the probes transmitted in each subgraph $\mathcal{G}_e$ where $e \in \mathcal{E}_R(\mathcal{G})$, there exists a lower bound $\ell_e \geq |\mathcal{P}(e)|$. Since network coding is applied, the probes transmitted on one link should have the same size. Similarly, the probes transmitted on one subgraph should also have the same size. Thus, for the probes transmitted in the subgraphs with overlapping links, the lower bound on probe size is the maximum value of the lower bounds obtained from Theorem 1. That is, $\ell_{\mathcal{G}} \geq \max_{e \in \mathcal{E}_R(\mathcal{G})} |\mathcal{P}(e)|$. ■

### 2.2.2 Algorithms for Finding a Valid Probe Coding Scheme

We propose an approach to find a valid probe coding scheme, such that the improved lower bound obtained from Proposition 2 is achieved. The approach is divided into three processes, described in this section.

**Constructing Auxiliary Trees**

For each end link $e = (h, r) \in \mathcal{E}_R$, we introduce an auxiliary tree topology $\mathcal{T}_e$. There is a one-to-many mapping from the nodes and the links in the original graph $\mathcal{G}$ to the nodes and the links in the auxiliary tree $\mathcal{T}_e$. 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 $i$th tree node that corresponds to the non-receiver node $v$ in graph $\mathcal{G}$. Similarly, we use $\varepsilon_i(e)$ to denote the $i$th tree link that corresponds to link $e$ in graph $\mathcal{G}$.

Algorithm 1 shows how to construct the auxiliary trees corresponding to each end link in set $\mathcal{E}_R$. For each end link $e = (h, r) \in \mathcal{E}_R$, nodes $u_0(r)$, $u_1(h)$ 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 defined 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.

**Selecting Coding Coefficients**

The coding coefficients are readily obtained based on the auxiliary trees. The non-receiver nodes with multiple incoming links in $\mathcal{G}$ are the nodes that perform network coding. The

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

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

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 to select coding coefficients. For each node $u_k(v)$ in $\mathcal{T}_e$, suppose it has a set of $t(u_k(v))$ incoming links $\{\varepsilon^1_{\text{in}}(e^1_{\text{in}}), \varepsilon^2_{\text{in}}(e^2_{\text{in}}), \ldots, \varepsilon^{t(u_k(v))}_{\text{in}}(e^{t(u_k(v))}_{\text{in}})\}$ and one outgoing link $\varepsilon_{\text{out}}(e_{\text{out}})$. Then, node $v$ has coding coefficients $[\delta(e^1_{\text{in}}, e_{\text{out}}), \delta(e^2_{\text{in}}, e_{\text{out}}), \ldots, \delta(e^{t(u_k(v))}_{\text{in}}, e_{\text{out}})]$. Suppose tree links $\varepsilon^1_{\text{in}}(e^1_{\text{in}}), \varepsilon^2_{\text{in}}(e^2_{\text{in}}), \ldots, \varepsilon^{t(u_k(v))}_{\text{in}}(e^{t(u_k(v))}_{\text{in}})$ have $n_1$, $n_2$,

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

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

$\ldots$, $n_{t(u_k(v))}$ corresponding leaf nodes, respectively. Then, we choose the values of coding coefficients as $[2^0, 2^{n_1}, 2^{n_1+n_2}, \ldots, 2^{n_1+n_2+\cdots+n_{t(u_k(v))-1}}]$.

### Designing Probe Packets

The path-link matrix $\mathbf{M}$ can be easily obtained based on the paths from the leaf nodes to the destination node in each auxiliary tree according to Algorithm 3. Now, we show how to 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 $\mathscr{G}$ with its end-link set $\mathcal{E}_R(\mathscr{G})$, according to Proposition 2, we calculate the tight lower bound on probe size $\ell_{\mathscr{G}} = \max_{e\in\mathcal{E}_R(\mathscr{G})} |\mathcal{L}_e|$. Thus, probes as $(0\cdots01)_2$ of length $\ell_{\mathscr{G}}$ are sent from the sources to the outgoing links in $\mathscr{G}$.

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 content of a received probe that denotes the case where only path $P_i$ has successfully transmitted a probe. In this way, we can establish the mappings 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.

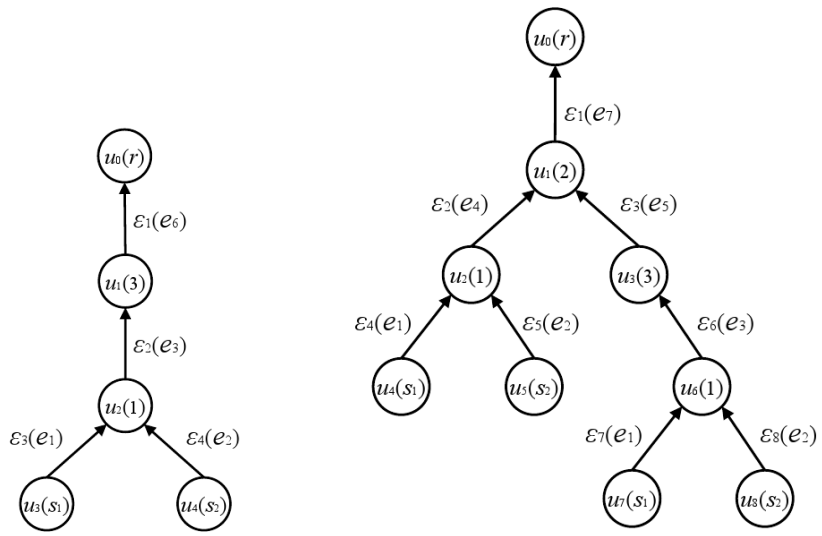
**Example 2:** We consider a directed acyclic graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ depicted in Fig. 2.2. We use Algorithms 1-2 to obtain a valid probe coding scheme with the minimum probe size. First, we construct two auxiliary trees $\mathcal{T}_{e_6}$ and $\mathcal{T}_{e_7}$, as depicted in Fig. 2.3. Second, we choose nodes 1 and 2 as the nodes that perform network coding, since they are the non-receiver nodes with multiple incoming links. Based on the auxiliary tree $\mathcal{T}_{e_6}$, we have $|\mathcal{L}(\varepsilon_3(e_1))| = 1$ (Algorithm 2, Steps 5-6). Thus, some of the coding coefficients of node 1 are obtained as $[\delta(e_1, e_3), \delta(e_2, e_3)] = [1, 2]$ (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 $[\delta(e_4, e_7), \delta(e_5, e_7)] = [1, 4]$, since $|\mathcal{L}(\varepsilon_2(e_4))| = 2$.

**Algorithm 3** Algorithm for constructing path-link matrix. Assume the auxiliary trees are given.

Initialize an empty path-link matrix $\mathbf{M}$
**for** each auxiliary tree $\mathcal{T}_e$ **do**
  $i \leftarrow 1$
  **for** each tree path $P_i$ in $\mathcal{T}_e$ **do**
    **for** each $m_{i,j}$, $j = 1, 2, \ldots, |\mathcal{E}|$ **do**
      If $\exists \varepsilon_k(e_j) \in P_i$, $m_{i,j} = 1$; otherwise, $m_{i,j} = 0$.
    **end for**
    Row vector $\omega \leftarrow (m_{i,j})_{1\times|\mathcal{E}|}$
    Update $\mathbf{M} \leftarrow \begin{bmatrix} \mathbf{M} \\ \omega \end{bmatrix}$
    $i \leftarrow i + 1$
  **end for**
**end for**

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

$$\mathbf{M} = \begin{array}{c} \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array}. \tag{2.8}$$




(a) Auxiliary tree $\mathcal{T}_{e_6}$ (b) Auxiliary tree $\mathcal{T}_{e_7}$

**Figure 2.3:** Two auxiliary trees $\mathcal{T}_{e_6}$ and $\mathcal{T}_{e_7}$, corresponding to end links $e_6$ and $e_7$, respectively.

The top two rows represents the two paths in subgraph $\mathcal{G}_{e_6}$, and the bottom four rows represents the four paths in subgraph $\mathcal{G}_{e_7}$. Since 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 $\mathscr{G} = \{\mathcal{G}_{e_6}, \mathcal{G}_{e_7}\}$. Counting the number of leaf nodes in each auxiliary tree, we obtain $|\mathcal{P}(e_6)| = 2$ and $|\mathcal{P}(e_7)| = 4$. Thus, $\ell_{\mathscr{G}} = \max\{2, 4\} = 4$ and probes as $(0001)_2$ are sent from sources $s_1$ and $s_2$ to outgoing links $e_1$ and $e_2$, respectively. □

## 2.3 Linear Algebraic Approach

As described previously, the system in (2.7) has $|\mathcal{P}|$ equations with $|\mathcal{E}_I \cup \mathcal{E}_V|$ unknowns. However, $|\mathcal{P}|$ may be less than $|\mathcal{E}_I \cup \mathcal{E}_V|$, such as the topologies in Figs. 2.1 and 2.2, and thus $\hat{\mathbf{a}}$ in (2.7) cannot be uniquely determined. Even when $|\mathcal{P}| \geq |\mathcal{E}_I \cup \mathcal{E}_V|$, it does not ensure that $\hat{\mathbf{a}}$ can be determined. In this section, we propose a linear algebraic (LA) approach using the observations from coding operations. We show that $\hat{\mathbf{a}}$ can be determined using the method of least-squares [35]. Then, we combine the methods of normal equations and row selection with the LA approach and analyze the computational complexity.

### 2.3.1 Least-squares Solutions

By inspecting the contents of the received coded probe packets at the destinations, we can estimate not only the success rate of a single path, but also the success rate of a set of paths. As the main 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 is unique to the networks which use probe coding schemes and cannot be achieved by routing probes in general.

We denote the power set[1] of $\mathcal{P}$ by $\mathscr{P}$. Thus, $|\mathscr{P}| = 2^{|\mathcal{P}|}$. Each element of $\mathscr{P}$ is a subset of $\mathcal{P}$, which can be used to represent a unique combination of paths. Let $\theta_i$ denote

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

the *path set success rate* of the $i$th path set in $\mathscr{P} \setminus \{\varnothing\}$, which is the probability that a batch of probes can be successfully transmitted on all the paths in the $i$th path set. We define a path set success rate except for $\varnothing \in \mathscr{P}$, 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 $\widetilde{\mathbf{M}} = (\widetilde{m}_{i,j})_{(|\mathscr{P}|-1)\times|\mathcal{E}_I\cup\mathcal{E}_V|}$ as follows: The element $\widetilde{m}_{i,j}$ is equal to 1 if there exists a path in the $i$th path set in set $\mathscr{P} \setminus \{\varnothing\}$ which includes the $j$th link in set $\mathcal{E}_I \cup \mathcal{E}_V$, and is equal to 0 otherwise. We refer to $\widetilde{\mathbf{M}}$ as type 2 modified path-link matrix.

The type 2 modified path-link matrix for the graph in Fig. 2.1 is shown below:

$$\widetilde{\mathbf{M}} = \left[\begin{array}{cccccc} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ \hdashline 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right]. \tag{2.9}$$

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

We define a column vector, $\mathbf{c} = (c_i)_{(|\mathscr{P}|-1)\times 1}$, where $c_i = \log\theta_i$. The column vector $\mathbf{a}$ is defined as in Section 2.1. Thus, we have a linear system

$$\sum_{j=1}^{|\mathcal{E}_I\cup\mathcal{E}_V|} \widetilde{m}_{i,j}a_j = c_i, \quad i = 1, \ldots, |\mathscr{P}| - 1 \tag{2.10}$$

or in the matrix form as

$$\widetilde{\mathbf{M}}\mathbf{a} = \mathbf{c}. \tag{2.11}$$

For each path set in $\mathscr{P} \setminus \{\varnothing\}$, the $n$ probe batches sent from the sources can be considered as a binomial experiment consisting 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 links) whose contents represent that all the paths in the $i$th path set have successfully transmitted a probe packet.

The sample proportion $\hat{\theta}_i = X_i/n$ is a maximum likelihood (ML) estimator of $\theta_i$ [36] (or an ML estimate resulting from end-to-end measurement $x_i$ substituted in the 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 (2.7) to a system of $|\mathscr{P}| - 1$ equations with $|\mathcal{E}_I \cup \mathcal{E}_V|$ unknowns, as follows:

$$\widetilde{\mathbf{M}}\hat{\mathbf{a}} = \hat{\mathbf{c}}. \tag{2.12}$$

The linear system in (2.12) has more equations than unknowns, i.e.,

$$|\mathscr{P}| - 1 \geq |\mathcal{E}_I \cup \mathcal{E}_V|. \tag{2.13}$$

This is because for every pair of links $e, e' \in \mathcal{E}_I \cup \mathcal{E}_V$, $\mathcal{P}(e)$ is different from $\mathcal{P}(e')$, while $|\mathcal{P}|$ paths can have at most $2^{|\mathcal{P}|} - 1$ different combinations of paths. The inequality (2.13) is a necessary condition for $\hat{\mathbf{a}}$ to be determined.

To show that $\hat{\mathbf{a}}$ in (2.12) can be determined by the least-squares, we introduce a $(|\mathscr{P}| - 1) \times (|\mathscr{P}| - 1)$ 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 to $\widetilde{\mathbf{M}}$, $\mathcal{M}(|\mathcal{P}|)$ has additional column vectors and has $|\mathscr{P}|-1$ column vectors in total. The $|\mathscr{P}|-1$ column vectors in the top $|\mathcal{P}|\times(|\mathscr{P}|-1)$ submatrix can represent all non-zero vectors in the vector space $\mathbb{F}_2^{|\mathcal{P}|}$. The bottom part of the additional columns are obtained according to the relation between the paths and path sets. An example of $\mathcal{M}(3)$ for $\widetilde{\mathbf{M}}$ in (2.9) is as follows:

$$\mathcal{M}(3)=\left[\begin{array}{ccccccc}1&1&0&0&1&0&1\\1&1&0&1&0&1&0\\1&0&1&0&0&1&1\\\hdashline 1&1&0&1&1&1&1\\1&1&1&0&1&1&1\\1&1&1&1&0&1&1\\1&1&1&1&1&1&1\end{array}\right]. \tag{2.14}$$

**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, 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.*

*Proof*: We prove it 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 non-zero entry. Now, consider matrix $\mathcal{M}(k+1)$. This matrix can be represented as follows after row and

column permutations:

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

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 new added path, while those 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 non-zero vector (i.e., there exists at least one non-zero 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. Since 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 non-zero. This entry corresponds to the last row in $\mathcal{M}_1$ or in $\mathcal{M}_2$. From now on, 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 non-zero 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-hand side and on the right-hand side. In this case, if an odd number of columns is chosen from the the right-hand side, the entry of the summation vector corresponding to the top row would be non-zero. However, if an even number of columns is chosen from the right-hand side, at least one entry of the summation vector corresponding to the rows in $\mathcal{M}_2$ would be non-zero, 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 non-zero entry in the summation vector. Therefore, all these $2^{k+1}-1$ column vectors in $\mathcal{M}(k+1)$ are linearly independent.

■

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, 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 $\mathcal{M}(|\mathcal{P}|)$. From 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, 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., $|\mathscr{P}| - 1 = |\mathcal{E}_I \cup \mathcal{E}_V|$, $\widetilde{\mathbf{M}}$ is a square matrix. Theorem 2 ensures that $\widetilde{\mathbf{M}}$ is invertible. Thus, $\hat{\mathbf{a}}$ can be determined as

$$\hat{\mathbf{a}} = \widetilde{\mathbf{M}}^{-1}\hat{\mathbf{c}}. \tag{2.15}$$

When the number of equations is greater than the number of unknowns, i.e., $|\mathscr{P}|-1 > |\mathcal{E}_I \cup \mathcal{E}_V|$, the system is over-determined. We can apply least-squares [35] to obtain an approximate solution which 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

$$\hat{\mathbf{a}} = (\widetilde{\mathbf{M}}^{\mathrm{T}}\widetilde{\mathbf{M}})^{-1}\widetilde{\mathbf{M}}^{\mathrm{T}}\hat{\mathbf{c}}. \tag{2.16}$$

We note that (2.15) is a special case of (2.16). ■

Corollary 1 gives an analytical solution of $\hat{\mathbf{a}}$ using least-squares. 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 $e_j \in \mathcal{E}_I \cup \mathcal{E}_V$, $1 - \hat{\alpha}_j$ is a function of $\hat{\alpha}_j$, while $\hat{\alpha}_j$ is a function of $\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_{|\mathscr{P}|-1}$. Since $\hat{\theta}_i \xrightarrow{p} \theta_i$, the continuous mapping theorem and Slutsky's theorem [36] yield that $1 - \hat{\alpha}_j \xrightarrow{p} 1 - \alpha_j$, where $\xrightarrow{p}$ denotes convergence in probability. ■

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

*Proof*: Theorem 3 shows that the loss rates of all identifiable links can be consistently 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 together prove this proposition. ■

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

### 2.3.2 Method of Normal Equations

**Algorithm 4** Method of Normal Equations [37]

1: Calculate the symmetric matrix $\widetilde{\mathbf{M}}^{\mathrm{T}}\widetilde{\mathbf{M}}$

2: Calculate Cholesky decomposition $\widetilde{\mathbf{M}}^{\mathrm{T}}\widetilde{\mathbf{M}} = \mathbf{L}\mathbf{L}^{\mathrm{T}}$

3: Calculate $\mathbf{d} \leftarrow \widetilde{\mathbf{M}}^{\mathrm{T}}\hat{\mathbf{c}}$

4: Use forward substitution to solve $\mathbf{L}\mathbf{y} = \mathbf{d}$ for $\mathbf{y}$

5: Use back substitution to solve $\mathbf{L}^{\mathrm{T}}\hat{\mathbf{a}} = \mathbf{y}$ for $\hat{\mathbf{a}}$

The most common technique to solve a full rank least-squares problem is the method of normal equations [37]. We define $\mu = |\mathscr{P}| - 1$ 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 requires $\mu\nu^2$ flops[2]. The second step is to calculate the Cholesky decomposition $\widetilde{\mathbf{M}}^{\mathrm{T}}\widetilde{\mathbf{M}} = \mathbf{L}\mathbf{L}^{\mathrm{T}}$ requiring $\nu^3/3$ flops. The third step is to calculate $\widetilde{\mathbf{M}}^{\mathrm{T}}\hat{\mathbf{c}}$ requiring $2\mu\nu$ flops. The fourth and fifth steps are to solve $\mathbf{L}\mathbf{y} = \widetilde{\mathbf{M}}^{\mathrm{T}}\hat{\mathbf{c}}$ for

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

$y$ 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 $\nu^2$ flops. Considering $\mu \geq \nu$ by (2.13), 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 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 $\kappa$ times in a monitoring period. Thus, the LA approach using the method of normal equations has a complexity of $\mathcal{O}(\mu\nu^2 + \mu n\kappa + \mu\nu\kappa)$ in practice.

### 2.3.3 Method of Row Selection

Since $\mu = 2^{|\mathcal{P}|} - 1$, the number of path sets $\mu$ grows exponentially 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, $\mathrm{rank}(\widetilde{\mathbf{M}}) = \nu$. This means there exist $\nu$ linearly independent path sets out of $\mu$ path sets which are sufficient to determine $\hat{\mathbf{a}}$.

To select $\nu$ linearly independent path sets, we modify the row selection algorithm proposed in [30], and obtain a reduced linear system as below:

$$\widetilde{\mathbf{M}}_1\hat{\mathbf{a}} = \hat{\mathbf{c}}_1, \tag{2.17}$$

where $\widetilde{\mathbf{M}}_1 \in \{0,1\}^{\nu\times\nu}$ and $\hat{\mathbf{c}}_1 \in \mathbb{R}^{\nu}$ consists of $\nu$ rows of $\widetilde{\mathbf{M}}$ and $\hat{\mathbf{c}}$, respectively. Algorithm 5 shows the modified row (path set) selection algorithm. This algorithm incrementally

**Algorithm 5** 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**

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 initial setup with a complexity of $\mathcal{O}(\mu\nu^2)$.

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 above steps for $\kappa$ times in a monitoring period. Thus, the LA approach using the method of row selection has a lower complexity of $\mathcal{O}(\mu\nu^2 + \nu n\kappa + \nu^2\kappa)$ in practice.

# Chapter 3

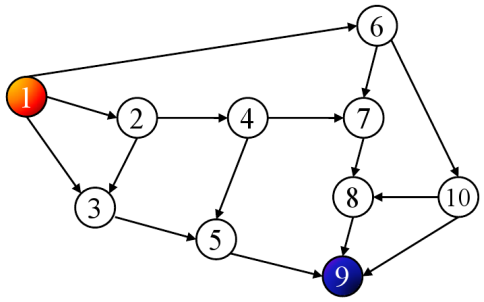
# Performance Evaluation

In this chapter, we access the performance of the LANT framework by simulations.

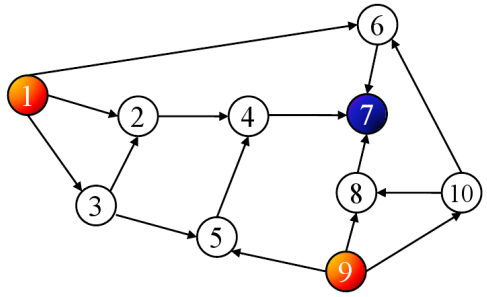
## 3.1 Simulation Setup

For the network topology, we first consider the Internet2 network map [38], which is a high-performance backbone network created by the Internet2 community. The topology is modified as the one used in [24], consisting of 10 nodes and 15 links. We apply the orientation algorithm [24] that converts the modified topology with selected sources to three directed acyclic graphs with different number of sources in Fig. 3.1, where all links are identifiable. Destinations are determined by the orientation algorithm.

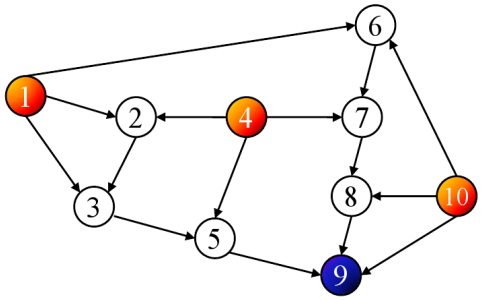
To consider larger networks, we use BRITE [39] to generate three router-level undirected network topologies with Waxman model. BRITE is a universal topology generater which improves the state of the art and is based on design principles which include representativeness, inclusiveness, and inter-operability. Representativeness leads to synthetic topologies that accurately reflect many aspects of the actual Internet topology (e.g. hierarchical structure, degree distribution, etc.). Inclusiveness combines the strengths of as




(a) One source.




(b) Two sources.




(c) Three sources.

**Figure 3.1:** Directed acyclic graphs with different number 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).

many generation models as possible in a single generation tool. Inter-operability provides interfaces to widely-used simulation applications such as ns, SSF and OmNet++ as well as visualization applications. In the three large network topologies generated by BRITE, the number of nodes are chosen as 20, 100, and 500. The number of links is twice the number of nodes in each topology.

A random link loss rate $1 - \alpha_j$ is assigned to the $j$th link $e_j \in \mathcal{E}_I \cup \mathcal{E}_V$, where the link success rate $\alpha_j$ is uniformly distributed within $[\alpha_{ave} - 0.05, \alpha_{ave} + 0.05]$. The value of $\alpha_{ave}$ 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 traversing 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 $j$th 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

$$\text{RMSE} = \left( \sum_{j=1}^{|\mathcal{E}_I \cup \mathcal{E}_V|} \frac{|\alpha_j - \hat{\alpha}_j|^2}{|\mathcal{E}_I \cup \mathcal{E}_V|} \right)^{1/2}. \tag{3.1}$$

We briefly summarize the belief propagation (BP) algorithm [26] and compare our LANT framework with the BP algorithm for loss rate inference through simulations in Section 3.2. Following the approach in [26], the first step is to create the factor graph from the original graph. The factor graph is a bipartite graph: on one side there are the links (variable nodes), whose loss rates we aim 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. We note that, unlike tree topologies considered in [26], in general topologies there might exist multiple paths for every source-receiver pair. The second step is to perform belief propagation. Each received probe triggers message passing in the factor graph and results in an estimate of link loss probabilities. Then, the estimates from different probes are combined, using standard methods [26], to obtain an estimate $1 - \hat{\alpha}_j$ of the actual link loss rate $1 - \alpha_j$ for the $j$th link $e_j \in \mathcal{E}_I \cup \mathcal{E}_V$.

The results are averaged over 100 simulations to eliminate possible random effects, where each simulation has new loss rate assignments and new loss processes.

## 3.2 Simulation Results

First, we investigate the influence of different methods adopted by LA approach on the estimation accuracy, based on the graph with one source in Fig. 3.1(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 5 to build a square matrix $\widetilde{\mathbf{M}}_1$, where each row (path set) includes 1 or 2 paths. This case is denoted by LA-RS. Fig. 3.2 shows the RMSE of 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 that of the LA-RS algorithm when $n = 50$. Such behavior is reasonable since 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

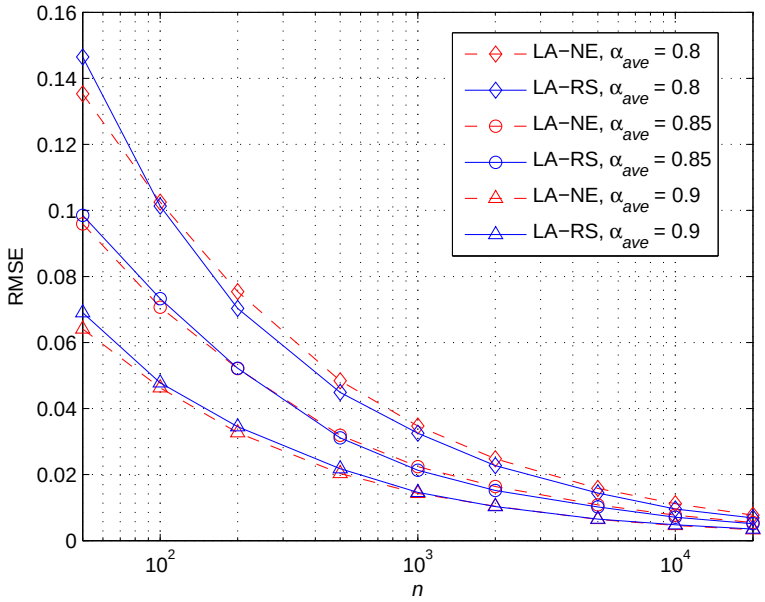



**Figure 3.2:** The RMSE of LA using the methods of normal equations (LA-NE) and row selection (LA-RS), versus the number of probe batches $n$.

number of probes increases. Therefore, for large number of probes, the performance of the LA-RS algorithm and the LA-NE algorithm is similar while the LA-RS algorithm outperforms the LA-NE algorithm in terms of the computational complexity.

Second, we compare the estimation accuracy of the BP algorithm and the LA-RS algorithm, based on the graph with one source in Fig. 3.1(a). Fig. 3.3 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 the LA-RS algorithm achieves better accuracy, after sending reasonably sufficient probe batches ($n > 400$). This is because the LA-RS algorithm exploits the losses on the different combinations of paths, while the BP algorithm only utilizes the losses on paths. Fig. 3.4

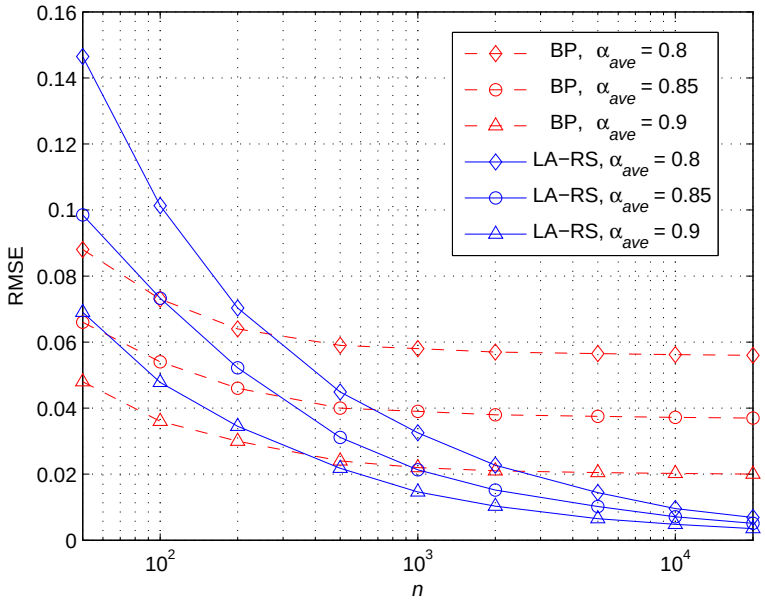



**Figure 3.3:** The RMSE of the BP algorithm and the LA-RS algorithm, versus the number of probe batches $n$, for different average success rate $\alpha_{ave}$.

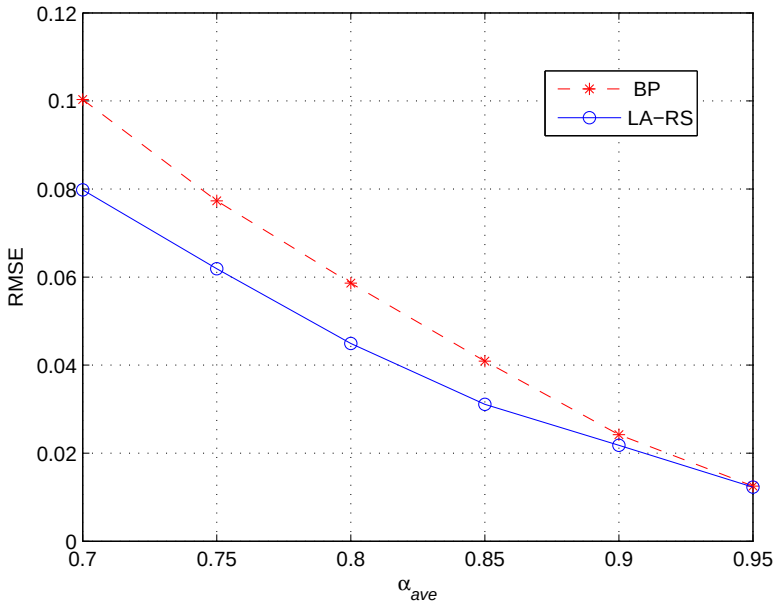



**Figure 3.4:** The RMSE of the BP algorithm and the LA-RS algorithm, versus the average success rate $\alpha_{ave}$ ($n = 500$).

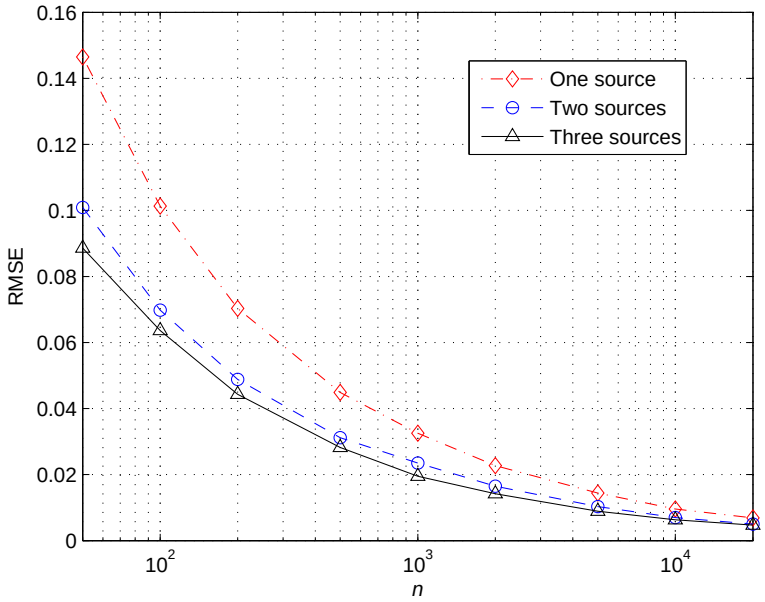



**Figure 3.5:** The RMSE of the LA-RS algorithm, versus the number of probe batches $n$, for different number of sources ($\alpha_{ave} = 0.8$).

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. 3.3. 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 $> 6\%$, $n = 20,000$), while the LA-RS algorithm would still achieve satisfactory accuracy (RMSE $< 1\%$, $n = 20,000$).

Third, for the networks with different number of sources in Fig. 3.1, Fig. 3.5 shows the RMSE as a function of the number of probe batches, and Fig. 3.6 shows the RMSE as a function of the average success rate. We compare the relative position of the three curves in Figs. 3.5 and 3.6, and obtain the following observation: The graph with more

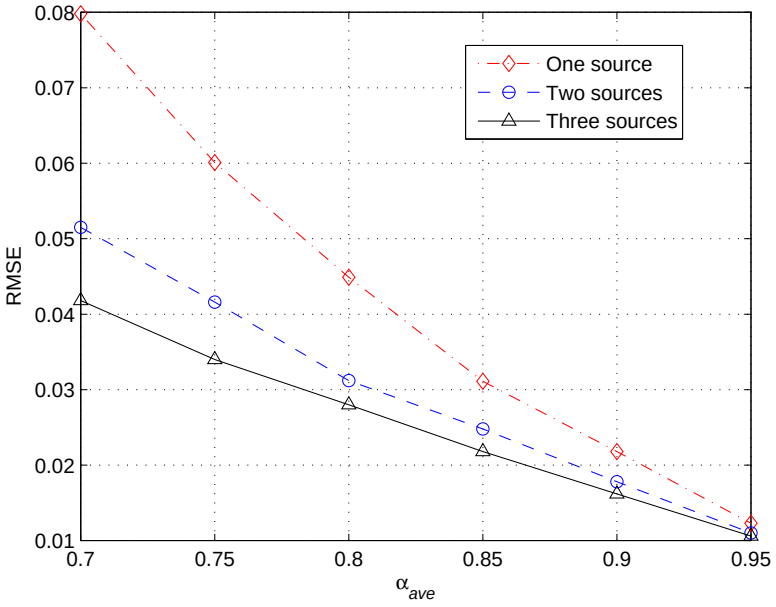



**Figure 3.6:** The RMSE of the LA-RS algorithm, versus the average success rate $\alpha_{ave}$, for different number of sources ($n = 500$).

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.

Finally, we investigate the performance of the LA-RS algorithm in three larger networks generated by BRITE. We randomly pick a part of nodes as source nodes in each network for 10 times. We pick 4 source nodes for the 20-node network, and 20 source nodes for the 100-node and the 500-node network. The orientation algorithm is applied to obtain directed acyclic graphs. There are 68.25% identifiable links on average in the directed graph for the 20-node network (40 links), 63.7% identifiable links on average in the the directed graph for 100-node network (200 links), and 26.17% identifiable links

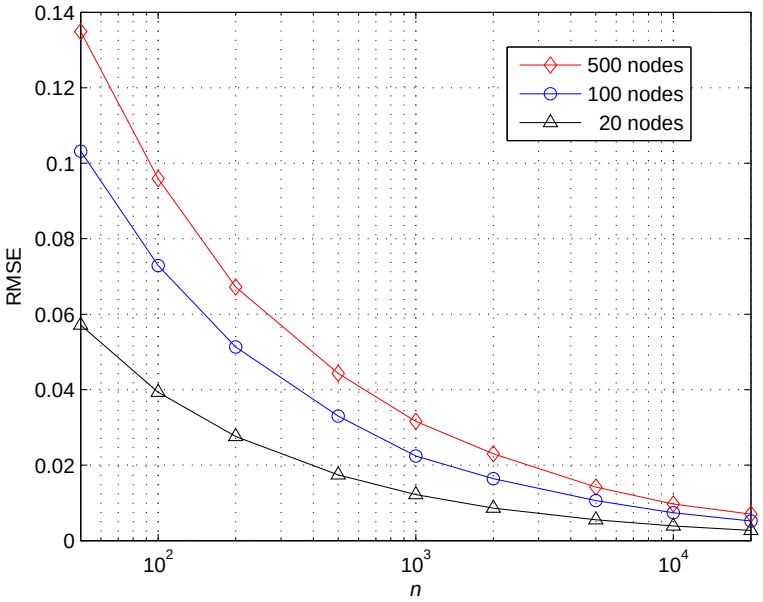



**Figure 3.7:** The RMSE of the LA-RS algorithm, versus the number of probe batches $n$, for networks of different sizes ($\alpha_{ave} = 0.9$).

on average in the directed graph for the 500-node network (1,000 links). 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 number and location of sources may result in different number of identifiable and virtual links. Fig. 3.7 shows the RMSE as a function of the number of probe batches for the networks of different sizes. As expected, the LA-RS algorithm still achieves satisfactory accuracy (RMSE $< 1\%$, $n = 20,000$), while more probe batches are needed to achieve the same accuracy in larger networks.

# Chapter 4

# Conclusions and Future Work

## 4.1 Conclusions

In this thesis, we studied the problem of link loss tomography on mesh topologies using network coding. We first provided an overview of the area of network tomography in communication networks. Accurate and efficient measurement of network-internal characteristics is critical for management and maintenance of large-scale networks. Recently there have been attempts to apply network coding in loss tomography in order to increase bandwidth efficiency. We introduced different inference techniques in the related works and explained the various performance bottlenecks such as (1) low bandwidth efficiency, (2) high monitoring cost, (3) estimation not being always accurate, and (4) requiring additional assumptions.

Then, we proposed a linear algebraic network tomography (LANT) framework for active inference of link loss rates. We first established a tight lower bound on the probe size for valid end-to-end observations when network coding is applied. Then, we developed algorithms to find a valid probe coding scheme, such that the lower bound on probe size is always achieved. Furthermore, we proposed a linear algebraic (LA) approach and devel-

oped consistent estimators of link loss rates. We also demonstrated that the complexity of LA using the method of row selection is lower than that 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.

We investigated the performance of the LANT framework under different simulation scenarios. Results showed that, for large number of probes, the performance of the LA-RS algorithm and the LA-NE algorithm is similar while the LA-RS algorithm outperforms the LA-NE algorithm in terms of the computational complexity. Moreover, the LA-RS algorithm achieves better estimation accuracy than the belief propagation (BP) algorithm when the estimators converge. 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 number and location of sources may result in different number of identifiable and virtual links.

## 4.2 Future Work

One possible extension of this work is to minimize the number of nodes performing network coding. In the current work, we choose all non-receiver nodes with multiple incoming links as the nodes that perform network coding. In this way, we can easily obtain a valid coding scheme. One possible solution is to choose the nodes next to the source nodes, since these nodes performing network coding can sufficiently distinguish different information flows, such that the intermediate nodes do not need to perform

network coding.

Another extension is design an algorithm to choose the number and location of the source and receiver nodes, such that all the links in the given network are identifiable. In the current work, we use the orientation algorithm [24] with selected sources to find a directed acyclic graph. Using this algorithm, we cannot control the number and location of the receiver nodes. To solve this problem, a new algorithm to find a directed acyclic graph with selected sources and receivers is necessary.

# Bibliography

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

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

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

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

[5] G. Sharma, S. Jaggi, and B. Dey, "Network tomography via network coding," in *Proc. of Information Theory and Applications Workshop*, San Diego, CA, Jan. 2008.

[6] P. Sattari, A. Markopoulou, and C. Fragouli, "Multiple source multiple destination topology inference using network coding," in *Proc. of NetCod Workshop*, Lausanne, Switzerland, June 2009.

[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'l Conf. Acoust., Speech, Signal Processing*, Salt Lake City, UT, May 2001.

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

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

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

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

[15] R. Koetter and M. Medard, "Beyond routing: An algebraic approach to network coding," in *Proc. of IEEE INFOCOM*, New York, NY, Nov. 2002.

[16] T. Ho and D. Lun, *Network Coding: An Introduction*. Cambridge University Press, 2008.

[17] C. Fragouli, J. LeBoudec, and J. Widmer, "Network coding: An instant primer," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 63–68, Jan. 2006.

[18] C. Fragouli, A. Markopoulou, R. Srinivasan, and S. Diggavi, "Network monitoring: It depends on your points of view," in *Proc. of Information Theory and Applications Workshop*, San Diego, CA, Jan. 2007.

[19] W. Zhu and Z. Geng, "A bottom-up inference of loss rate," *Computer Communications*, vol. 28, no. 4, pp. 351–365, Mar. 2005.

[20] N. Duffield, F. LoPresti, V. Paxson, and D. Towsley, "Inferring link loss using striped unicast probes," in *Proc. of IEEE INFOCOM*, Anchorage, AK, Apr. 2001.

[21] V. Padmanabhan, L. Qiu, and H. Wang, "Passive network tomography using bayesian inference," in *Proc. of the 2nd ACM SIGCOMM Workshop on Internet Measurement*, Marseille, France, Nov. 2002.

[22] T. Bu, N. Duffield, F. LoPresti, and D. Towsley, "Network tomography on general topologies," in *Proc. of ACM SIGMETRICS*, Marina Del Rey, CA, June 2002.

[23] C. Fragouli and A. Markopoulou, "A network coding approach to network tomography," in *Proc. of 43rd Allerton Conf.*, Monticello, IL, Sept. 2005.

[24] M. Gjoka, C. Fragouli, P. Sattari, and A. Markopoulou, "Loss tomography in general topologies with network coding," in *Proc. of IEEE GLOBECOM*, Washington, DC, Nov. 2007.

[25] H. Yao, S. Jaggi, and M. Chen, "Network coding tomography for network failures," in *Proc. of IEEE Infocom (Mini-Conference)*, San Diego, CA, March 2010.

[26] Y. Mao, F. Kschischang, B. Li, and S. Pasupathy, "A factor graph approach to link loss monitoring in wireless sensor networks," *IEEE J. Select. Areas Commun.*, vol. 23, no. 4, pp. 820–829, Apr. 2005.

[27] F. Kschischang, B. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[28] Y. Zhao, Y. Chen, and D. Bindel, "Towards unbiased end-to-end network diagnosis," *IEEE/ACM Trans. Networking*, vol. 17, no. 5, pp. 1724–1737, Dec. 2009.

[29] B. Sun and Z. Zhang, "Probabilistic diagnosis of link loss using end-to-end path measurements and maximum likelihood estimation," in *Proc. of IEEE ICC*, Dresden, Germany, June 2009.

[30] Y. Chen, D. Bindel, H. Song, and R. Katz, "Algebra-based scalable overlay network monitoring: Algorithms, evaluation, and applications," *IEEE/ACM Trans. Networking*, vol. 15, no. 5, pp. 1084–1097, Oct. 2007.

[31] M. Coates and R. Nowak, "Network loss inference using unicast end-to-end measurement," in *Proc. of ITC Seminar on IP Traffic, Measurement and Modelling*, Monterey, CA, Sept. 2000.

[32] H. Su, W. Chen, S. Lin, D. Jin, and L. Zeng, "The inference of link loss rates with internal monitors," in *Proc. of IEEE GLOBECOM*, New Orleans, LA, Dec. 2008.

[33] J. Gui, V. Shah-Mansouri, and V. Wong, "A linear algebraic approach for loss tomography in mesh topologies using network coding," in *Proc. of IEEE ICC*, Cape Town, South Africa, May. 2010.

[34] ——, "Accurate and efficient network tomography via network coding," submitted to *IEEE Trans. on Vehicular Technology*, 2010.

[35] R. Myers, D. Montgomery, and G. Vining, *Generalized Linear Models: With Applications in Engineering and the Sciences*. John Wiley & Sons, Inc., 2001.

[36] G. Grimmett and D. Stirzaker, *Probability and Random Processes*, 3rd ed. Oxford University Press, 2001.

[37] G. Golub and C. Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, 1996.

[38] "Internet2 network," http://www.internet2.edu/network/.

[39] "BRITE," www.cs.bu.edu/brite/.