INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
RICE UNIVERSITY

Loss Inference in Unicast Network Tomography
Based on TCP Traffic Monitoring

by

Yau-Yau Yolanda Tsang

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE
Master of Science

APPROVED, THESIS COMMITTEE:

Dr. Robert Nowak, Chair
Assistant Professor of Electrical and
Computer Engineering

Dr. Richard G. Baraniuk
Professor of Electrical and Computer
Engineering

Dr. Edward W. Knightly
Assistant Professor of Electrical and
Computer Engineering

Houston, Texas

April, 2001
Loss Inference in Unicast Network Tomography Based on TCP Traffic Monitoring

Yau-Yau Yolanda Tsang

Abstract

Network tomography is a promising technique for characterizing the internal behavior of large-scale networks based solely on end-to-end measurements. Despite the efficiency of active probing in most network loss tomography methods, these measurements impose an additional burden on the network in terms of bandwidth and network resources. They can therefore cause the estimated performance parameters to differ substantially from losses suffered by existing TCP traffic flows. In this thesis, we propose a promising passive measurement framework based on the sampling of existing TCP flows. We demonstrate its performance using extensive ns-2 simulations. We observe accurate estimates of link losses (with 2% mean absolute error). We also describe the Expectation-Maximization (EM) algorithm in solving the Maximum Likelihood (ML) Estimates in terms of individual link loss rates as an incomplete data problem. Finally, we present a new method for simultaneously visualizing the network connectivity and the network performance parameters.
Acknowledgments

I would like to express my deep gratitude to all my committee members, Dr. Robert Nowak, Dr. Richard G. Baraniuk and Dr. Edward W. Knightly. Dr. Nowak has been an excellent advisor in guiding me through the research and a supportive friend. I am grateful to Dr. Baraniuk and Dr. Knightly for their enlightening discussions which led to the enrichment of the thesis.

I also want to thank all my friends, the Rice Digital Signal Processing Group members. I am especially grateful to Dr. Mark Coates for his constructive advices and numerous insightful discussions.

Finally, my heartfelt thanks go to my parents for their care and support throughout these years.
# Contents

Abstract \hspace{1em} ii
Acknowledgments \hspace{1em} iii
List of Illustrations \hspace{1em} vi

## 1 Introduction

1.1 Contribution \hspace{1em} 2
1.2 Related Work \hspace{1em} 4

## 2 Unicast Network Tomography

2.1 Unicast Tomography \hspace{1em} 6
2.2 Loss Modeling \hspace{1em} 8
2.3 Passive Measurement Framework
  2.3.1 Transmission Control Protocol (TCP) \hspace{1em} 10
  2.3.2 Measurement Scheme \hspace{1em} 11

## 3 Maximum Likelihood Network Tomography

3.1 Measurement framework \hspace{1em} 13
  3.1.1 Single Packet Measurement \hspace{1em} 14
  3.1.2 Back-to-Back Packet Pair Measurement \hspace{1em} 14
3.2 Likelihood Estimates \hspace{1em} 15

## 4 Expectation-Maximization Algorithm

4.1 Review of the Incomplete Data Problem \hspace{1em} 18
4.2 Unobserved Data Likelihood \hspace{1em} 19
5 Performance Visualization

5.1 Motivation for Performance Visualization

5.2 Visualization Tool

5.3 Example of Visualization Tool

6 ns-2 Simulation Experiment

6.1 Simulation Framework

6.2 Simulation Results

7 Discussion and On-going Work

8 Conclusion

Bibliography
Illustrations

2.1 An example of a depth-4 logical tree with one source (node 0), four internal routers (nodes 1 to 4) and seven receivers (nodes 5 to 11)............. 6

2.2 Pseudo-code for passive sampling of the existing traffic. .................. 12

4.1 Path from source to receiver $i = 8$ with unobserved data at each internal router. ................................................. 20

4.2 A path consisting of three internal nodes. ................................. 21

4.3 Pseudo-code for the EM algorithm. ..................................... 25

5.1 Network connectivity graphs in (a) standard representation with loss rates indicated next to the link and in (b) proposed visualization graph. .... 27

5.2 Visualization of loss rates of a 32-receivers network with a binary tree structure in %. ................................................................. 28

5.3 (a) Ratios of received packet counts (raw data) for each receiver pair (in %). (b) Reconstructed loss rates for each link for network connectivity shown in Fig. 5.1 (a). (c) Absolute error between true and estimated loss rates for each link. ............................................. 30

6.1 Simulation Results. True and estimated link-level success rates of TCP flows from source to receivers for several traffic scenarios: (a) Heavy losses on link 4-8, (b) Heavy losses on links 1-2, 2-5, (c) Heavy losses on links 1-2, 4-8, and (d) Traffic mixture with medium losses. . 33
6.2 The performance error (mean absolute error averaged over all links) versus measurement period. ........................................... 34

7.1 An example of 'lumping' receivers to reduce network complexity. . . . 36
Chapter 1

Introduction

Characterizing the network parameters becomes crucial for the development of the current Internet. The knowledge of network parameters allows network engineers to improve the network design and control. Network managers can use this information to monitor and manage the network. This information is also valuable for Internet Service Providers (ISP) to determine pricing policies. They can also use it to determine bandwidth, number of routers, and exchange points needed to meet the Service Level Agreements (SLA) with their customers. Moreover, the customers can use this information to decide which ISPs do the best job and whether the qualities of services (QoS) are being delivered.

This kind of information can be obtained by setting up internal monitors at all routers to gather statistics. Unfortunately, this requires the modification of network architecture, which is an onerous and expensive task. Even if we assume that the statistics are available, the diversity in network management can prevent any individual or organization from collecting relevant statistics. Furthermore, as the complexity of the Internet grows, the composition of all this data into a comprehensive picture is not well understood.

An alternative method is to infer the internal network behavior from end-to-end measurements: the so-called network tomography problem. Early efforts [1, 3, 4, 12, 15, 16] have focused on probing the network to collect correlated statistics in terms of losses/delays at the multicast receivers (where packets are broadcast from a
source to a group of receivers). If a packet reaches one of the receivers, it provides correlation to nodes beyond it. These multicast-based algorithms are impressive, but a significant portion of the current Internet does not support multicast routings. The majority of traffic is unicast in nature, e.g. Telnet and File Transfer Protocol (FTP), where packets are sent from a source targeted at one particular receiver. Moreover, the routers treat multicast and unicast differently. Therefore, inference made by multicast receivers may poorly reflect the actual network performance.

As a result, strategies based on unicast routings are proposed [6, 11, 17] to avoid limitations caused by multicast based inference. This is a much more difficult problem because unicast packets do not have the correlations as experienced by multicast packets. To address this difficulty, authors use closely time-spaced packet pairs (back-to-back packets) to mimic the correlation in multicast.

1.1 Contribution

In this thesis, we adapt and extend the unicast inference techniques proposed in [6, 7]. The Expectation-Maximization (EM) algorithm [8, 14] is used in solving the maximum likelihood estimation problem by taking advantage of the complete data likelihood. Because of the graphical nature of the network, probabilistic modeling together with the upward-downward algorithm [10] simplifies the expectation step in the EM algorithm. It is shown to be robust for any given tree-structured network.

In [7, 11], the authors suggest that both active probing and passive sampling of existing traffic can be used in the collection of statistics, but a number of issues regarding the passive sampling have not been addressed. Active probing refers to insertion of packets into the network for the sole purpose of collecting data while passive sampling is based on the existing traffic.
The motivation for passive sampling is two-fold. Insertion of a large number of probe packets into the network traffic imposes the risk of changing the performance of the network traffic. Moreover, the measurements based on the losses experienced by the probe packets might be substantially different from the existing traffic flows. Extra bandwidth and network resources will be consumed. We propose a new, truly passive methodology for unicast network tomography. We evaluate the feasibility and performance of our approach using ns-2 [2] simulation environment. Performance of the inference algorithm based on the sampled traffic will be quantified in terms of mean absolute error.

The other contribution is a new method for simultaneously visualizing network connectivity and network performance parameters. Upon estimating the internal link loss probability, we need a method to present this estimated information. However, current tools on the market can only depict directly measured performance parameters. A method which conveys spatial connectivity as well as the estimated parameters, e.g. loss rates, is proposed. In this thesis, we focus on visualizing link loss rates, but extensions to delay statistics and bandwidth are also possible.

In Chapter 2, we review the basic unicast network tomography problem, examine the technical issues involved, and describe the passive measurement framework. In Chapter 3, we pose unicast network tomography as a Maximum Likelihood Estimation (MLE) problem. In Chapter 4, we describe the EM algorithm which takes advantage of the incomplete data likelihood to solve for the MLE problem. In Chapter 5, we propose and illustrate a visualization tool for reporting the estimated link loss rate data. In Chapter 6, we examine the performance of our passive measurement framework through extensive simulations in ns-2 environment. In Chapter 7, we provide a discussion of our simulation results and present on-going work. Concluding
remarks are made in Chapter 8.

1.2 Related Work

There exist several unicast network tomography based methods for characterizing link-level behavior. One of the first schemes using back-to-back packet pairs is proposed in [12] to identify bottleneck bandwidth. Even though the measurement metric is different from the loss rates, the idea of introducing correlation in unicast probe packets is the same. Other network probing techniques using back-to-back packets can be found in [5, 11, 15]. Recently, authors in [9] proposed an alternative strategy based on sending multiple probe packets (more than 2 closely time spaced packets) to improve the observed correlations, and then applied the multicast-based algorithm of the MINC project [1] for loss rate estimation.

Our work is an extension of the work presented in [6, 7] and is closely related to [4, 5, 11]. The authors in [4] presented extensive performance analysis of multicast loss inference techniques. In this thesis, we aim to perform a similar analysis for the passive unicast inference.

In [5, 11], the authors applied packet pairs in detecting shared losses using unicast probes. In a two-receiver tree-structured network, they were interested in detecting whether or not losses in two flows suffering similar losses are occurring on the shared path. The authors in [11] applied Bayesian probing to the network. They mentioned the possibility of passively recording the statistics but they applied only active probing to their simulations. The measurements they made were spaced at a fixed probing interval. Their work was an extension of [5] in which Markovian probing technique was used. In our work, we estimate individual link statistics, and thereby the congestion level of all links in a larger network, in which there are more than two receivers.
Chapter 2

Unicast Network Tomography

Today, most of the Internet Protocol based applications are based on unicast delivery. These applications include HyperText Transfer Protocol (HTTP), FTP, Simple Mail Transport Protocol (SMTP), Telnet and others [13], in which a connection involves only one source and one destination. The other common method of delivering data-grams is through the use of multicast. In multicast delivery, the same piece of data is delivered from one source or multiple sources to many destinations. However, the scalability and routing policy issues have prevented multicast from being deployed in the current Internet. One of the issues concerning routing protocols is that multicast routers have to maintain the per flow state, keeping track of all forwarding information as well as the source locations. Besides, unlike unicast, multicast offers no policy control mechanisms between peers. Multicast packets are treated differently from the unicast ones. Moreover, the existence of non-multicast-capable routers have made the multicast study even more difficult. For this reason, we focus our attention on unicast inference. In this chapter, we provide a brief overview of the unicast inference problem, describe the loss measurement framework and review the passive monitoring scheme.
2.1 Unicast Tomography

Network tomography is a study of the internal behavior of a network through end-to-end measurements. Current effort has been made to study loss rates, delay statistics, bandwidth information, topology and other service related parameters. In this thesis, we will focus on the estimation of link loss rates.

The topology of the network depends on the connectivity between senders and receivers. It varies as the routing table updates. For inference purposes, we assume that all measurements are made during the stationary period, i.e. while the routing table, and thus the topology, is fixed.

![Diagram of a depth-4 logical tree with one source (node 0), four internal routers (nodes 1 to 4) and seven receivers (nodes 5 to 11).]

Figure 2.1: An example of a depth-4 logical tree with one source (node 0), four internal routers (nodes 1 to 4) and seven receivers (nodes 5 to 11).

We consider a scenario where the receivers are connected to a source with some common paths. From the perspective of the source, it is a tree-structured network. More specifically, we are interested in the logical tree structured network where all nodes (except the source and receivers) have one parent and at least two children. Since all measurements are edge based, we cannot resolve individual links in isolated subpaths (subpaths where nodes have only one child). Therefore, we group them
together into a single composite link. Thus, for a packet traversing between two nodes, for example from node 1 to 2, it is possible for it to pass through a number of single-chained routers. Figure 2.1 depicts an example of a depth-4 logical tree, a topology of a network consisting of one source (node 0), four internal routers (nodes 1 to 4) and seven receivers (nodes 5 to 11). To distinguish between a link and a path, we define a 'link' as the connection between two adjacent nodes in the logical tree and a 'path' as the connection between the source and a receiver. In this thesis, we will only focus on single source scenarios. However, extensions to cases of multiple sources are possible.

When we say all measurements are assumed to be edge-based, we imply that we can only measure the number of packets sent by the source and received by each receiver. This information can be obtained via Transmission Control Protocol's (TCP) acknowledgement system (see Section 2.3.1).

Given those measurements, it is straightforward to measure the path success probability. However, this information is not sufficient to resolve individual link success probabilities. There is no unique mapping of success probabilities because more than one configuration can give rise to the same path level measurements. To overcome this difficulty, a methodology based on back-to-back packet pairs measurements is used. Back-to-back packet pairs measurement was first used in [12] to refer to two closely time-spaced packet pairs, possibly destined for different receivers but sharing a set of common links in their paths. Given the information of one packet, we expect the other packet within a pair to have similar experience through the shared link. If the packet pairs are sent across a link and one of them is received, then it is highly likely that the other one will also be received. That is, conditioning on the success transmission of a packet in the pair, the success probability of the other packet is very
close to one. This observation has been verified in real network experimentally [3, 15]. Assuming total correlation between the packets in the pair, if one of the packets is dropped, then losses can only occur on unshared links. Collecting all possible combinations of these packet pair measurements, we should be able to resolve link loss probabilities. Unfortunately, the correlation on the shared subpaths are not perfect. Therefore, a more sophisticated method is necessary, as described next.

2.2 Loss Modeling

In this section, we define the relevant parameters in tomographic loss rate estimation. Throughout the remainder of the thesis, we work with 'success probability'. It eases the mathematical parameterization of the problem and the probability of loss is simply one minus the success probability.

For single packet transmission, we define the unconditional success probability of link $i$ (the link into node $i$), $\alpha_i$, as

$$\alpha_i \equiv Pr(\rho(i) \rightarrow i),$$

where $\rho(i)$ denotes the index of the parent node of node $i$ (e.g. $\rho(1) = 0$) and $\rho(i) \rightarrow i$ denotes the successful transmission of a packet from $\rho(i)$ to $i$. The packet is successfully sent from $\rho(i)$ with probability $\alpha_i$ and is dropped with probability $1 - \alpha_i$.

For back-to-back packets measurements, we define the conditional success probability, $\gamma_i$, of the link as

$$\gamma_i \equiv Pr(1st \text{ packet } \rho(i) \rightarrow i \mid 2nd \text{ packet } \rho(i) \rightarrow i).$$

The 'first' and 'second' refer to the temporal order of the two packets. That is, given that the second packet of the pair is received on a link, the success probability of
the first packet on the same link is $\gamma_i$ and dropped with probability $1 - \gamma_i$. We expect $\gamma_i$ to be very close to one if the background traffic between the two packets under consideration is small. The actual value (the degree of correlation) depends on the number of other arrival and service events in the background traffic between the arrival of the pair into the queue. The two back-to-back packets become less correlated ($\gamma_i < 1$) as the number of inter-arrival events increases.

We model all these arrivals and services of packets having the same unit, thus an arrival event increments the queue length by one and a service event decrements the length by one. We also assume the queueing policy to be independent of the length of the queue. For example, the packet is dropped only when the queue is full using the DropTail policy. In the next section, we detail the passive traffic monitoring scheme for TCP-based traffic.

2.3 Passive Measurement Framework

Most network tomography studies have used active probing in verifying their inference schemes, e.g. [3, 4, 15, 19]. Active probing perturbs the existing traffic by inserting probe packets into the network traffic for the purpose of measurements only. Large number of probe packets are inserted to ensure sufficient measurements are made. Because of this, measurements using these probe packets might not reflect the real network traffic and lead to biased estimates of true loss. Instead of active probing, we propose a passive measurement scheme which takes advantage of the existing traffic and does not consume additional network resources. We are mainly interested in Transmission Control Protocol (TCP) traffic. The details of the passive measurement scheme will be introduced after a brief overview of how TCP functions.
2.3.1 Transmission Control Protocol (TCP)

TCP is one of the Internet's transport layer protocols. It provides a reliable data-transfer service by ensuring the data streams are not corrupted, missing or duplicated. This reliable data transfer depends on the acknowledgement segment (ACK) field and the sequence number field in the TCP segment header. When a connection is established between two hosts, a unique sequence number will be assigned and inserted into the header such that the receiving end will know exactly what it is expecting next. If the receiver end discovers an out of order sequence number, it will generate a duplicated ACK for the last in-order byte of data received. When the sender receives the third duplicated ACK, it knows that the segment after the ACKed segment is lost and it will perform a fast retransmit of the missing packets.

Another important component of TCP is its congestion control mechanism. It controls by the congestion window and the threshold. The size of the congestion window determines the Maximum Segment Size (MSS) that can be transmitted in single transmission. The threshold affects how the congestion window grows. When the network is not congested, the number of packets sent across the link, increases exponentially. When the window size reaches the threshold, the number of packets sent will grow linearly instead, to avoid congestion. However, once packet loss (missing ACK) is detected, the congestion window size diminishes to allow only one MSS and the threshold will be reset to half of the MSS reached previously. For this reason, TCP sometimes is known to be an additive increase, multiplicative decrease algorithm.

Assuming the ACK packets will never be dropped in the network, they provide a means of measurement for our passive measurement framework. We want to make measurements such that they are a subsampled version of the traffic. When the traffic is heavy, we want to make more measurements, and when it is light, just a few of
them. The other constraint is to have sufficient back-to-back measurements. These packets have to be closely time spaced within pairs and the inter-pairs have to be spaced sufficiently far enough to approximate the inter-pair temporal independence. Thus, the measurement scheme has to fulfill the above requirement, as described next.

2.3.2 Measurement Scheme

In the measurement scheme, we consider numerous short TCP sessions randomly generated between the sender and the receivers throughout the measurement period, together with varying TCP and User Datagram Protocol (UDP) [13] background traffic. To ensure a sufficient number of back-to-back packet measurements, we scan through the traffic trace to locate packet pairs which are less. We identify the first packet pair by locating two packets sent from the source within a time threshold, $\delta_t$. Then we skip forward by a $\Delta_t >> \delta_t$ time interval to locate another pair. The whole procedure repeats until the end of the measurement period. Single packets which do not violate the time separation requirements are included afterwards. Figure 2.2 illustrates the pseudo-code used in these passive measurement framework.

With all these measurements in hand, we have to define the statistics for our loss inference scheme. We count the measurements for each possible combination: the number of packets sent, and among those, the number of packets received. In the next chapter, we formally describe the types of measurements and detail the inference task.
Passive sampling methodology

1. Set $\delta_t =$ packet pair spacing;
2. Set $\Delta_t =$ inter pairs spacing;
3. Record the time stamps and sequence numbers at the source;
4. while (not end of trace) {
5. \hspace{1cm} Compare the time difference between current and previous packet time stamp;
6. \hspace{1cm} if (time difference $\leq \delta_t$) AND (the packet has not been paired)
7. \hspace{1cm} Record current sequence number and receiver id and group the packets as a pair;
8. \hspace{1cm} }
9. \hspace{1cm} Update the current time by $\Delta_t$;
10. }
11. for (each pair) {
12. \hspace{1cm} if (packet conditioned on is received) {
13. \hspace{1cm} Increment associated packet pairs measurements;
14. \hspace{1cm} }
15. \hspace{1cm} else {
16. \hspace{1cm} Increment associated single packets measurements;
17. \hspace{1cm} }
18. }

Figure 2.2: Pseudo-code for passive sampling of the existing traffic.
Chapter 3

Maximum Likelihood Network Tomography

In this chapter, we formally describe the two types of measurements we require for the inference task. Using these measurements, we form the likelihood functions for unknown parameters, the conditional and unconditional probabilities associated with each link.

We now list the assumptions made in the measurement framework:

- Bernoulli loss models are assumed. There are two and only two possible outcomes; that is, packets can be either received or lost at receivers.

- Packet pair data on a particular link is independent of and identically distributed to data for other packet pairs on the same link.

- Spatial correlation is negligible. Losses across different paths are mutually independent.

3.1 Measurement framework

In the previous chapter, we introduced the two types of measurements: those for single packets and back-to-back packets. These measurements are affected by \( \alpha_i \) and \( \gamma_i \), the two probabilities associated with each link, as described next.
3.1.1 Single Packet Measurement

Each single packet transmission is modeled as an independent Bernoulli trial. The path success probability is given by $p_i$. We record the outcome of each trial, where 1 denotes a successful transmission and 0 a packet loss. Suppose $n_i$ packets are sent to receiver $i$ and among those, $m_i$ are received with probability $p_i$. Since the sum of independent Bernoulli trials is Binomial, the likelihood function of $m_i$ given $n_i$ and $p_i$ is thus Binomial and is given by

$$l(m_i \mid n_i, p_i) = \binom{n_i}{m_i} p_i^{m_i} (1 - p_i)^{n_i - m_i},$$

where $p_i = \prod_{j \in \mathcal{P}(0, i)} \alpha_j$ is the product of all the unconditioned probabilities along the path and $\mathcal{P}(0, i)$ denotes the sequence of nodes in the path from the source 0 to receiver $i$. For example, $\mathcal{P}(0, 9) = 1, 3, 4, 9$ in Fig. 2.1 and $p_9 = \prod_{j \in \mathcal{P}(0, 9)} \alpha_j = \alpha_1 \alpha_3 \alpha_4 \alpha_9$.

3.1.2 Back-to-Back Packet Pair Measurement

In back-to-back packets measurements, given the receipt of the 2nd packet in a pair to node $j$, we model the 1st packet to node $i$ as a Bernoulli trial with success probability $p_{i,j}$. Suppose the source sends out a large number of closely time spaced packet pairs to receivers $i$ and $j$. Let $n_{i,j}$ be the number of $(i, j)$ pairs in which the 2nd packets reaches receiver $j$, and among those, let $m_{i,j}$ be the number of packets successfully received by both $i$ and $j$. Let $k_{i,j}$ be the node where the paths $\mathcal{P}(0, i)$ and $\mathcal{P}(0, j)$ diverge; then the two packets share the common path $\mathcal{P}(0, k_{i,j})$. For example, refer to Fig. 2.1, let $i = 9$ and $j = 10$, then $k_{9,10} = 3$. The likelihood of $m_{i,j}$ given $n_{i,j}$ is given by

$$l(m_{i,j} \mid n_{i,j}, p_{i,j}) = \binom{n_{i,j}}{m_{i,j}} p_{i,j}^{m_{i,j}} (1 - p_{i,j})^{n_{i,j} - m_{i,j}},$$
where

\[ p_{i,j} = \prod_{q \in P(0, k_{i,j})} \gamma_q \prod_{s \in P(k_{i,j}, j)} \alpha_s. \]

We collect an assortment of measurements, where single packets are sent to different receivers and back-to-back packets to a combination of receivers. We define the collection as

\[ \mathcal{M} \equiv \{m_i\} \cup \{m_{i,j}\} \]

\[ \mathcal{N} \equiv \{n_i\} \cup \{n_{i,j}\}, \]

where the index \( i \) alone runs over all receivers and the indices \( i, j \) run over all pairwise combinations of receivers in the network.

### 3.2 Likelihood Estimates

Denote the collections of the unconditional and conditional link success probabilities as \( \alpha \) and \( \gamma \), respectively. The joint likelihood of all mutually independent measurements is given by

\[ l(\mathcal{M} | \mathcal{N}, \alpha, \gamma) = \prod_i l(m_i | n_i, p_i) \prod_{i,j} l(m_{i,j} | n_{i,j}, p_{i,j}). \]

The maximum likelihood estimates (MLE) of \( \alpha \) and \( \gamma \) are defined as

\[ (\hat{\alpha}, \hat{\gamma}) = \arg \max_{\alpha, \gamma} l(\mathcal{M} | \mathcal{N}, \alpha, \gamma), \]

or equivalently,

\[ (\hat{\alpha}, \hat{\gamma}) = \arg \max_{\alpha, \gamma} \log l(\mathcal{M} | \mathcal{N}, \alpha, \gamma). \]

The aim is to estimate \( \hat{\alpha}, \hat{\gamma} \). Unfortunately, the dimension and the coupling effect of the unknown probabilities impose time and computational burdens on the direct
computation of maximum likelihood estimates or marginal likelihood functions. The computation is formidable. The major difficulty here is to separate the coupling of link success probabilities with the measurements. As one can see from the likelihood function, it involves a product of subsets of $\gamma$ and/or $\alpha$ probabilities. Thus the probabilities involved are non-linear functions of $\gamma$ and/or $\alpha$. Furthermore, there is no closed form solution for the maximum likelihood problem. To overcome this difficulty, we take advantage of a common device known as *unobserved/incomplete* data, as described in the next chapter.
Chapter 4

Expectation-Maximization Algorithm

In the previous chapter, we formulated the loss rate inference as a Maximum Likelihood Estimation (MLE) problem. In our problem setting, we want to compute the maximum likelihood estimates of $\alpha$ and $\gamma$, but the coupling of subsets of $\alpha$ and $\gamma$ in the joint likelihood function imposes difficulty in direct maximization. However, if the packet counts at each node along the path (the complete data likelihood) are known, then the problem is greatly simplified. The joint likelihood given the complete data likelihood is given by the products of likelihood functions involving only one unknown variable. Note that since the complete data cannot be measured, the likelihood function can only be estimated.

Expectation-Maximization (EM) algorithm is an attractive approach to compute the MLE from incomplete data. The algorithm, as the name suggests, involves two steps - the Expectation (E) step and the Maximization (M) step. In the E-step, we estimate the complete data using the estimates of the parameters. In the M-Step, we use the estimated complete data to estimate the parameters. These two steps iterate until convergence.

Before we proceed further, we review the Expectation-Maximization (EM) algorithm [8, 14] as a solution to an incomplete data problem. We then formulate our problem as an unobserved data likelihood. Probability propagation and the upward-downward algorithm [10] will be introduced to further simplify the E-step computation.
4.1 Review of the Incomplete Data Problem

Denote the individual link probabilities along a path as $\theta_i$. We wish to compute the MLE of $\theta$,

$$\hat{\theta} = \arg\max_{\theta} l_y(\theta),$$

where $\theta$ is defined on sample space $\mathcal{Y}$ and $y$ is an observation from $\mathcal{Y}$. The density of $\mathcal{Y}$ is given by $g(y; \theta)$, thus, $l_y(\theta)$ is the log-likelihood function defined by

$$l_y(\theta) = \log g(y; \theta).$$

Assume another more informative sample space $\mathcal{X}$ is available with density $f(x; \theta)$ and there exists a many-to-one mapping from $\mathcal{X}$ to $\mathcal{Y}$. We can only measure the incomplete data $y$ and $y = t(x)$ where $t$ is a non-invertible function. Thus, $y$ can be considered as a compressed version of $x$. We refer $x$ as the complete data. If we have had measured $x$, the ML problem would have become easier to solve. The density of $Y$ is related to that of $X$ by

$$g(y; \theta) = \int_{t^{-1}(y)} f(x; \theta) dx.$$ 

Knowing the observed sample $y$ does not provide us information about the complete data $x$. Therefore, the complete data likelihood can only be estimated. We first initialize the parameters in some ad hoc manner. Then we estimate the missing data (the incomplete data). The parameters are then updated with the estimates. The procedure iterates until convergence.

To illustrate the monotonicity of the algorithm, we introduce notation $k(x \mid y; \theta)$ for the conditional density of $x$ given $y$ and $\theta$,

$$k(x \mid y; \theta) = \frac{f(x; \theta)}{g(y; \theta)}.$$
We can rewrite

\[ l_y(\theta) = \log g(y; \theta) \]
\[ = \log f(x; \theta) - \log k(x | y; \theta) \]
\[ = Q(\theta, \theta) - H(\theta, \bar{\theta}), \]

where

\[ H(\theta, \bar{\theta}) = E[\log k(x | y; \theta) | y; \bar{\theta}]. \]

As a consequence of Jensen's inequality,

\[ H(\theta, \bar{\theta}) \leq H(\theta, \theta). \]

This ensures the monotonicity of the EM algorithm. The likelihood increases in each iteration. For interested readers, please refer to [8, 14] for details.

Suppose that \( \theta^{(s)} \) denotes the current estimates of \( \theta \) after \( s \) cycles, starting with an initial guess \( \theta^{(0)} \). The EM algorithm can be generalized as follows:

**E-Step:** Compute \( Q(\theta, \theta^{(s)}) = E[\log f(x; \theta) | y; \theta^{(s)}] \)

**M-Step:** \( \theta^{s+1} = \arg \max_{\theta} Q(\theta, \theta^{(s)}) \)

In our problem setting, \( \mathcal{M}, \mathcal{N} \), and the internal packet counts form the complete data.

### 4.2 Unobserved Data Likelihood

Recall that in single packet measurements, the likelihood is found to be \( l(m_i | n_i, p_i) \) where \( p_i \) is the path success probability (product of link success probabilities). The maximization task will be simplified if we can decouple the likelihood such that we can rewrite the likelihood function containing only the subset of the paths. If we
have all the packet counts at each node $u_{j,i}$, $j \in P(0,i)$, $j \neq i$, then we can form the complete data likelihood function as

$$l(u_{j,i} | n_i, \rho_i) = \prod_{j \in P(0,i)} \left( \frac{u_{\rho(j),i}}{u_{j,i}} \right)^{u_{j,i}} (1 - \alpha_j)^{u_{\rho(j),i} - u_{j,i}},$$

where $\rho(j)$ again denotes the parent of node $j$. Also, since we are able to measure at the source and receiver, in the expression above we set $u_{0,i} = n_i$ and $u_{i,i} = m_i$. The example in Figure 4.1 illustrates the notion of unobserved data. In a similar fashion, we introduce unobserved data for all measurements (including packet pairs). The key feature of the complete data likelihood function is that it factors into a product of individual binomial likelihood functions, each involving just a single success probability. Thus, the complete data likelihood function is a trivial multivariate function, and the effects of the individual link probabilities are easily separated. Furthermore, a closed form MLE is achievable.

![Diagram](figure4.1.png)

Figure 4.1: Path from source to receiver $i = 8$ with unobserved data at each internal router.

In the next section, we describe the use of probability propagation with the upward-downward algorithm in the E-Step to compute the conditional expectation of the likelihood function given the measurements in an effective manner.
4.3 E-Step

In the E-Step, we need to estimate the complete data by computing the conditional expectation of the likelihood function given the packet counts. This can be calculated from the probability distribution of each link. The probability computations can become very tedious as the number of unknown parameters increases. However, this can be greatly simplified using probability propagation and the upward-downward algorithm.

4.3.1 Probability Propagation

Probability propagation is used to infer exact distributions over each variable if the model has only one single, acyclic path. The idea is to remove a variable at each stage, such that global sum can be decomposed into local sum and provide a tractable computation. Figure 4.2 illustrates an example of a path consisting of four links. The probability of link $i$ (the link into node $i$) is denoted as $p_i$ and the packet counts are indicated as $n_i$. The number of packets sent is $n_0$, and $n_4$ of those are received according to the probability distribution. We are interested in finding the expected value of $n_1$ through $n_3$, given $n_0$, $n_4$ and the $p_i$'s.

![Diagram of a path consisting of three internal nodes.](image)

Figure 4.2: A path consisting of three internal nodes.

We first consider

$$Pr(n_0, n_1, n_2, n_3, n_4) = Pr(n_4|n_3, n_2, n_1, n_0) Pr(n_3|n_2, n_1, n_0) Pr(n_2|n_1, n_0) Pr(n_1|n_0)$$
\[ = Pr(n_4|n_3)Pr(n_3|n_2)Pr(n_2|n_1)Pr(n_1|n_0), \]

The second equality holds because the packet counts at each node cannot exceed the packet counts of the previous node. It is only conditionally dependent on the previous node.

Suppose we want to compute \( Pr(n_3|n_0, n_4) \). We notice that we can compute \( Pr(n_0, n_3, n_4) \) and then normalize to get \( Pr(n_3|n_0, n_4) \). Therefore,

\[ Pr(n_3|n_0, n_4) = Pr(n_0, n_3, n_4)/\sum_{n_3} Pr(n_0, n_3, n_4). \]

To compute \( Pr(n_0, n_3, n_4) \), it can either be derived from \( \sum_{n_1, n_2} Pr(n_0, n_1, n_2, n_3, n_4) \) or we can apply the probability propagation technique which is more effective,

\[ Pr(n_0, n_3, n_4) = Pr(n_4|n_3) \sum_{n_2} Pr(n_3|n_2) \sum_{n_1} Pr(n_2|n_1)Pr(n_1|n_0). \]

4.3.2 Upward-Downward Algorithm

One might note that the computation can be repetitive along the shared paths. Instead of computing all possible paths separately, we can take advantage of the Upward-Downward algorithm [10]. This algorithm is also known as the forward-backward algorithm or message passing algorithm. The information or the messages are passed from the leaves towards the root level by level. Generally speaking, at each level, information is marginalized and multiplied. The information is stored with each node for future use. When the information eventually reaches the root node, the information will be passed back to the leaves level by level. After completion of message passing, conditional probabilities associated with each link given the number of packets sent and received will be computed.

Let’s refer back to our previous example in Fig. 4.2. In the upward steps, we pass
the message from the receiver and propagate it to the sender,

\[ q(n_3 \mid n_2) = p(n_3 \mid n_2) \sum_{n_4} p(n_4 \mid n_3) \]

\[ q(n_2 \mid n_1) = p(n_2 \mid n_1) \sum_{n_3} q(n_3 \mid n_2) \]

\[ q(n_1 \mid n_0) = p(n_1 \mid n_0) \sum_{n_2} q(n_2 \mid n_1). \]

When the message reaches the root node, we know \( p(n_1 \mid n_4) \propto q(n_1 \mid n_0) \). Therefore, we can apply the result towards the leaf nodes through the downward step to compute the conditional probability of the remaining internal nodes.

\[ p(n_2 \mid n_4) \propto \sum_{n_1} p(n_1 \mid n_4)q(n_2 \mid n_1) \]

\[ p(n_3 \mid n_4) \propto \sum_{n_2} p(n_2 \mid n_4)q(n_3 \mid n_2). \]

We have illustrated how the Upward-Downwards algorithm functions on a single path. For a tree-structured network, we combine all messages on the shared path by multiplying the distribution for all paths sharing the link. Then similarly, we propagate the messages from receivers to the source in the upward step.

Once the conditional probabilities are found, we can derive the conditional expectation for each link. Therefore, the complexity of the E-step is \( O(NL) \) where \( N \) is the number of possible measurements that we can make and \( L \) is the number of levels involved.

### 4.4 M-Step

After estimating the complete data likelihood, the maximization is greatly simplified because the likelihood function involves only one unknown parameter. Thus, M-Step becomes very trivial. It only involves a normalizing stage for each unknown probability. The probability associated with each link is the ratio of packet counts
received on that link to the number of packets sent. The complexity of the M-
Step is $O(L)$ and thus the overall complexity of the EM algorithm becomes $O(NL)$.
The pseudo-code of the EM algorithm using the above simplifications is included in
Figure 4.3.
Expectation-Maximization Algorithm

1. Initialize $\alpha_i$ and $\gamma_i$ to 0.5, $\forall i$;
2. Initialize diff to an arbitrary big number;
3. Set threshold $= 10^{-3}$;
4. Input connectivity information;
5. Construct data structure for storing packet counts for each parent-children pair;
6. \[ \text{while (diff} > \text{threshold)} \{ \]
7. \[ \text{for (each possible path) } \{ \]
8. \[ \text{E-Step: compute complete data likelihood} \]
9. \[ \text{Input current estimates and packet counts for upward-downward algorithm;} \]
10. \[ \text{Update parent-children packet counts;} \]
11. \[ \} \]
12. \[ \text{M-Step: Normalizing packet counts} \]
13. \[ \text{for (each link) } \{ \]
14. \[ \text{Divide packet counts at children node by its parent counts;} \]
15. \[ \} \]
16. \[ \text{diff} = \text{mean absolute error between current and previous } \alpha \text{ estimates;} \]
17. \[ \} \]

Figure 4.3: Pseudo-code for the EM algorithm.
Chapter 5

Performance Visualization

In previous chapters, we described how to perform end-to-end inference on network parameters, specifically on loss statistics inference. With all this data in hand, one might ask what is the best way to convey this information effectively. In this chapter, we suggest a method which answers this question. We first motivate the need for network performance visualization. Then we describe the visualization tool that we propose. Finally, we illustrate the use of the visualization tool with our link loss statistics.

5.1 Motivation for Performance Visualization

One might notice that there are numerous network management and visualization tools available on the market. However, those tools are used for directly measurable performance parameters. Besides, networks are commonly depicted using a connectivity graph, in which nodes of the graph represent routers or end-systems and the edges of the graph represent links (wire connections) between nodes. The graph of a network consisting of only one sender and several receivers is tree-structured, as depicted in Fig. 2.1. Performance information for each link can be presented in a numeric form near the corresponding link. In Fig. 5.1 (a) we indicate the loss rates next to each link. Alternatively, the information can be provided in a simple tabular format.
These standard presentation schemes are effective for small networks, but as the network complexity and size increase, such a representation becomes very difficult to interpret. When human interaction and interpretation are required in the system, we need to have an alternative method that depicts the performance information more effectively. For example, spatial relationships between losses in a network may be difficult to detect from a tabular or graphical presentation.

The goal of estimated parameter visualization is to prompt network managers and engineers to rapidly respond to changes in the network conditions by reallocating resources and exercising control options. The visualization tool has to convey network connectivity as well as the parameters estimated (or measured). The information has to allow rapid and global human interpretation.

Figure 5.1: Network connectivity graphs in (a) standard representation with loss rates indicated next to the link and in (b) proposed visualization graph.

5.2 Visualization Tool

The visualization tool we propose is an alternative representation for network performance visualization and is better suited to rapid and global human interpretation.
The tree structured graph can be related to a particular tiling of the plane, which expresses the spatial connectivity of the network. Connectivity between links corresponds to vertical adjacency between tiles. The gray scale value assigned to each tile represents a performance parameter value (e.g., loss rate, average delay) for a link. Each column of the 'image' corresponds to a path from the source to a particular receiver, and the number of tiles intersected by the column indicates the number of hops on the path from the sender. This accounts for the non-regularity of the partition depicted in Figure 5.1, for example. The performance over each path can be calculated, in the case of loss statistics, by vertical multiplication over tiles for path loss rates.

![Graph](image)

Figure 5.2: Visualization of loss rates of a 32-receivers network with a binary tree structure in %.

In Fig. 5.1 (a), we illustrate the tree-structured connectivity graph representing a single-source, multiple-receiver network. We indicate the loss rate (in %) next to each link. The same representation in 5.1 (b) uses the proposed visualization graph.
Each block represents the loss rate experienced by the link. The block on the top represents the estimate for the root. The lowest level represents those for the receivers. In Fig. 5.2, we illustrate the loss rates for a larger network whose topology is given by a binary tree structure with 32 receivers. The congested link(s) can be easily identified using our method.

5.3 Example of Visualization Tool

In this section, we illustrate the visualization tool by evaluating the performance of the algorithm in the case of loss rate mapping. We apply the EM-algorithm to estimate the loss rate on individual links given by Fig. 5.1 (a). Figure 5.3 (a) depicts the raw data in terms of the ratio of received packet counts to the packet sent for each receiver pair (path loss rates). Figure 5.3 (b) shows the estimated link loss rates for the same network and Fig. 5.3 (c) displays the absolute errors between the true and estimated loss rates for each link. The average absolute error per link is less than 2%, compared to the average loss rate per path of approximately 12%.
Figure 5.3: (a) Ratios of received packet counts (raw data) for each receiver pair (in %). (b) Reconstructed loss rates for each link for network connectivity shown in Fig. 5.1 (a). (c) Absolute error between true and estimated loss rates for each link.
Chapter 6

ns-2 Simulation Experiment

In this chapter, we evaluate the measurement framework developed earlier (Section 2.3) using the ns-2 simulation environment. This allows us to evaluate the performance of our inference scheme (with the EM algorithm) in the passive measurement framework. For accurate inference, we also vary the length of the measurement period to collect sufficient statistics. In the following sections, we introduce the simulation scenario, including the topology and traffic generation, followed by the simulation results.

6.1 Simulation Framework

We use the same topology in Fig. 2.1 in all simulations. The bandwidth on each link (indicated next to the link) is intended to reflect the nature of current Internet to a certain extent. Typically, the links at end-hosts are slower than the Internet backbones. The number of hops (links) from the sender to receivers is intended to illustrate the idea of distance in order to study the effect of fan-out. However, as mentioned in Section 2.1, the number of links from the sender to receivers do not represent neither the physical distance nor the number of routers that the packets pass through. Moreover, in all simulations, queue length is set to allow a maximum number of 35 packets at one time. The queueing policy is fixed to be DropTail. Thus, packets are dropped when the queue reaches its capacity of 35 packets.
In the simulations, we focus on packets sent from the source to receivers using TCP protocols. Thus, we assume the TCP connections between the source and receivers last for the entire measurement period for a random on and off period. Other than these connections, we also include background traffic composed of a mixture of short duration TCP and User Datagram Protocol (UDP) connections. On average, there are thirty TCP and thirty UDP connections operating in the network at any given time. The utilization of each link is high, which allows us to study the losses, and thereby perform loss inference.

The loss inference statistics are collected from the network trace using the procedures as described in Section 2.3.2. Ten simulations are carried out for each scenario (as described later) for a 300 second interval. We set the maximum spacing within a pair to be $\delta_t = 1$ms and the minimum spacing between pairs to be $\Delta_t = 10$ms. The four scenarios are described as follows.

**Traffic Scenarios 1-3: Heavy losses on 1 or 2 links** These scenarios are set up to access our ability in identifying where significant losses occur in the network. To simulate heavy losses, we add substantial exponential on-off UDP traffic as well as short duration TCP connections to simulate heavy link traffic.

By having heavy losses on one link, our intention is to test the ability of the framework in localizing losses at links near the receivers, i.e. heavy losses at links connecting node 4 and 8 (link 4-8) in Fig. 2.1. The scenario in which losses occur on two links is to assess the ability in identifying distributed losses (occurring in different parts of the network) and cascaded losses in the network. These two congested link scenarios are simulated as substantial loss on links 1-2, 4-8 and links 1-2, 2-5 respectively.

**Traffic Scenario 4: Mixed traffic with medium losses** In this scenario, we
distribute losses over all links caused by a mixture of TCP and UDP with varying connection times. We also insert extra links to internal nodes to develop cross traffic with different round trip times. This allows us to test the accuracy of the loss inference scheme on all links.

Figure 6.1: Simulation Results. True and estimated link-level success rates of TCP flows from source to receivers for several traffic scenarios: (a) Heavy losses on link 4-8, (b) Heavy losses on links 1-2, 2-5, (c) Heavy losses on links 1-2, 4-8, and (d) Traffic mixture with medium losses.
Figure 6.2: The performance error (mean absolute error averaged over all links) versus measurement period.

6.2 Simulation Results

Ten independent simulations were conducted for each scenario for a measurement period of 300 seconds. Fig 6.1 depicts the simulation results. In each subfigure, the two panels display values for each link 1-11 (horizontal axis): the top panel is a plot of true and estimated success rates and the bottom panel is a plot of the mean absolute error between the estimated and true link success rates over 10 trials. The results show that our estimation is in good agreement with the true losses, with a maximum mean absolute error of about two percent. The passive measurement framework is thus capable of identifying the bottleneck as well as giving a good estimate of the true loss on each link.

In Figure 6.2, we also examine the relationship between the average percentage error and the length of the measurement period using one scenario. As expected, the error decreases as the measurement period increases. Note that even for a 60 second
simulation time, the mean absolute error is less than 0.6%.

We also want to draw attention to the fact that the measured losses for TCP flows can be very different from those of the UDP traffic. For one simulation of the mixed traffic scenario, we observe an average TCP loss of 3% on link 2-6 and link 3-10, whereas the UDP traffic has a loss rate of nearly 20%. This shows that active probing may provide a poor indication of losses in existing TCP connections.
Chapter 7

Discussion and On-going Work

One crucial step in our methodology is the sampling (or 'mining') of those back-to-back packets from the existing traffic. As described earlier (see Section 2.3), the passive measurement framework gives rise to a fast extraction algorithm depending on the throughput to receivers. Unfortunately, due to the nature of TCP connections, there are large number of packet pairs where both packets are targeted at the same receiver. This is due to the congestion window of the TCP connection in which groups of packets from the same source are transmitted. Thus, packet pairs involving different receivers are more rare. This happens only when the source switches from one connection to the other. The information contained in packet pairs targeted at different receivers (cross-packets) allows isolation of individual link success probabilities and provides more information for inference, even though fewer pairs like this exist.

Figure 7.1: An example of ' lumping ' receivers to reduce network complexity.
An alternative approach is to include as many informative packet pairs as possible by first filtering out all the cross-packets. Since they usually spread out for a longer time than the required $\Delta_t$ separation, this does not violate our assumption of independent measurements between packet pairs. If the spacing between pairs is less than the $\Delta_t$ separation requirement, we have to decide which set of closely-time spaced cross-packet pair to eliminate. The elimination process should try to remove those with the least representation in the current set of included pairs. Then we carefully include those packet pairs sent to the same receivers without violating the inter packet spacing requirement. Finally, we include single packet measurements, again maintaining the $\Delta_t$ time separation requirement. This approach will allow us to extract more informative data and should provide a better estimate.

A more aggressive approach in collecting informative statistics could involve an alternating service strategy at the source. The goal is to generate more cross pairs. Other sophisticated passive measurement schemes as well as the one described above are currently under investigation.

Scalability becomes another issue in passive framework, especially when the network size increases. As the number of nodes grow in the network, the potential for collecting sufficient number of cross pairs diminishes. It will be difficult to involve all possible combinations of receivers in packet pairs measurements. This results in a 'data starvation' problem in which we have insufficient measurements for the inference task. It limits ones ability to infer internal link loss in the passive measurement scheme. One possible solution is to group links into clusters and thereby abstract the actual network to a smaller size network. This can be done by replacing the cluster by an 'effective node'. As a result, packet pairs and single packets can be shared among the receivers in the cluster and a reliable estimate can be made on links close
to the source.

For the simplest case, we consider the estimate of $\alpha_1$ on a complete binary tree structured network with and without topology abstraction/lumping. We assume that there is total correlation between packet pairs, thus $\gamma_i = 1 \forall i$ and $n_i$ is large. We want to show that by lumping, we reduce the error bound, which turns out to be the covariance of the measurements on the estimated link(s) of interest.

First, we consider the single packet measurements. Recall in Chapter 3, for $n_i$ packets sent to receiver $i$, $m_i$ of those packets are received, and the product of path success probability is $p_i = \prod_{j \in P(i)} \alpha_j$ and $m_i = n_i p_i$. We model $m_i$ as $n_i$ independent, identically distributed Bernoulli trials with success probability $p_i$ and the likelihood function of $m_i$ is Binomial. Since $n_i$ is large, there are enough values of $m_i$ to make an approximation of a continuous random distribution. We assume $p_i$ is not 0 or 1, such that the distribution is nearly symmetric. By Central Limit Theorem, the distribution can be approximated with that of a normal random variable with mean $\mu_i = n_i p_i$ and variance $\sigma^2_i = n_i p_i (1 - p_i)$, that is,

$$m_i \sim N(\mu_i, \sigma^2_i).$$

If we approximate $x_i \approx p_i \equiv \frac{m_i}{n_i}$, we obtain the distribution of $x_i$ as

$$x_i \sim N\left(\frac{\mu_i}{n_i}, \frac{\sigma^2_i}{n_i^2}\right).$$

Note that if $X$ is normally distributed, $X \sim N(\mu, \sigma^2)$, then $\log X$ is also asymptotically normal $\log X \sim N(\log |\mu|, \sigma^2/\mu^2)$ [18]. This can extend to packet pairs measurements, $x_{i,j}$.

Define $\mathcal{X} \equiv \{x_i\} \cup \{x_{i,j}\}$ and $N$ to be the total number of measurements made. To formulate it as an estimation problem, we rewrite the measurements in logarithm
form and express it as

\[ Y \approx A\theta + \log \epsilon, \]

where \( Y \) is a \( N \times 1 \) vector of \( \log \mathcal{X} \). \( A \) is the transformation matrix consisting of elements in \( \{0, 1\} \) indicating the link probability involved. \( \theta \) is a vector of \( \log \alpha_i \) and \( \epsilon \) is the measurement error, \( N(0, \Sigma) \). Since all measurements are mutually independent, \( \Sigma \) has only non-zero diagonal elements which are variances for each element of \( \mathcal{X} \).

Now we have to solve for a \( N \)-dimensional parameter \( \theta = [\theta_1, ..., \theta_N]^T \) given an observation vector \( Y \) which is normally distributed with density function \( f_Y(y; \theta) \). The Cramer-Rao lower bound on \( \widehat{\theta}_1 \) is the upper-left \((1, 1)\) element of the covariance of the unbiased estimator of \( \widehat{\theta} \), which is also the inverse of Fisher Information Matrix. Thus, we can compare the upper-left \((1, 1)\) elements for both the lumped and non-lumped case.

Numerical analysis as well as \texttt{ns-2} simulations have shown promising preliminary results. The error bound of \( \alpha_1 \) in the lumped case is lower than that in the non-lumped case, and thus variance of the estimate is closer to the true value. We are currently investigating this approach through a theoretical analysis for the study of error bound on any arbitrary link.
Chapter 8

Conclusion

In this thesis, we present a passive unicast network tomography methodology for inferring individual link-loss rates. The framework relies on passive sampling of the existing TCP traffic flows, based on the sequence numbers and the ACK packets received. The inference scheme is based on computing maximum likelihood estimates from incomplete data using Expectation-Maximization algorithm. The maximum likelihood estimate is greatly simplified using the complete data likelihood. Moreover, a visualization tool is illustrated, to convey both network connectivity and estimated parameters from our inference scheme.

From ns-2 simulations, we show that sufficient data for accurate inference can be collected within a short measurement period. We also address issues that arise from passive measurement, such as scalability and the 'data-starvation' problem. In the case of scalability, alternative sampling schemes are described to increase the measurements of the more informative data, the 'cross-pairs'. For the 'data-starvation' problem, we discuss a method in abstracting the network into a smaller size network to focus on a subset of interesting links. Preliminary results show promise in reducing the error bound on the links to non-lumped nodes. Both of these topics are currently under investigation.
Bibliography


[2] The network simulator-2. For more information, see http://www.isi.edu/nsnam/ns/.


