Noname manuscript No.
(will be inserted by the editor)

A Two-Stage Approach For Network Monitoring

Linda Bai · **Sumit Roy**

Received: date / Accepted: date

Abstract A goal of network tomography is to infer the status (e.g. delay) of *congested* links internal to a network, through end-to-end measurements at boundary nodes (end-hosts) via insertion of probe signals. Because a) probing constitutes traffic overhead, and b) in any typical scenario, the number of congested links is a *small fraction* of the total number in the network, a desirable design objective is to identify those (few) congested links using a *minimum number of probes*. In this paper, we make a contribution to solving this problem, by proposing a new *two-stage* approach for this problem.

First, we develop a binary observation model linking end-to-end observations with individual link statuses and derive necessary and sufficient conditions for whether at least one link in the network is congested. Stage I of the proposed method shows that achieving 1-identifiability with a *minimum number of probes* is equivalent to the familiar *minimum set covering* problem that can be efficiently solved via a greedy heuristic. A *sequential* algorithm is described, leading to a significantly lowered computational complexity vis-a-vis a batch algorithm. Next, a binary splitting algorithm originally developed in group testing is used to identify the location of the congested links. The proposed scheme is evaluated by simulations in OPNET and experiments on the PlanetLab testbed to validate the advantages of our 2-stage approach vis-a-vis a conventional (batch) algorithm.

Keywords Network Tomography · Group Testing · Link Monitoring · Active Probing

Linda Bai · Sumit Roy
Department of Electrical Engineering, University of Washington, Seattle, WA, 98195, USA
Tel.: +1-206-616-9249

Linda Bai
E-mail: lyb3@u.washington.edu

Sumit Roy
E-mail: roy@ee.washington.edu

1 Introduction

As the Internet grows in size and diversity (types of host nodes), monitoring link-level properties (e.g. delay, loss rate) of interest becomes a growing challenge as a key component of an overall network management approach. Since access to any internal nodes in a network is often infeasible, one must presume that only a set of end-hosts is available for end-to-end measurements. Inferring characteristics of interior links based on such end-to-end measurements is a fundamental problem of great practical interest, and is broadly referred to as the *network tomography* problem [1].

If the traffic rate on a link approaches its available bandwidth, packets on this link will experience large delays and ultimately, loss. Hence probe packets sent between end-hosts along a route that contains at least one such congested link will experience significant end-to-end delay, signifying congestion onset. However, from such end-to-end measurements, it is not always feasible to uniquely identify *which* link(s) are congested - we refer to this as the *identifiability* problem.

Identifiability has been bypassed in network monitoring by resorting to stochastic approaches that estimate the properties of *all* links [1,2]. Stochastic approaches presume that the link delays are a random variable, specified by a suitable probability model such as a Gaussian or an exponential distribution as in [1]. The tradeoff between stochastic models and deterministic models is summarized in Table.1. Link delay estimators based on deterministic models reflect the randomness in the observed data that is not modeled. Hence, a stochastic model with a good prior distribution for link delays will typically lead to lower variance estimates than deterministic ones, while difference between the *assumed* distribution and the true distribution will result in estimator bias. Note, when a model is correct or approximates reality well, a random model is not biased. Since link delays are assumed to be randomly time-varying in the stochastic regime, the preferred method of probing uses packet pairs (generalizable to a packet train) that are sent back-to-back, i.e. within the coherence time of the network flows. In this manner, the relative delay observed at the receive end-hosts between the packet pairs serves as the measurement of interest. For the deterministic model where the link delay is modeled as an unknown constant (at least over the duration of the measurements), such pairwise probing is not needed. A single packet with a time-stamp entered by the source node suffices to provide a measurement of the path latency, assuming all nodes in the network are synchronized. Finally, and perhaps most pertinently, over-parametrization in the stochastic model results in solutions with significantly more computational complexity than deterministic approaches. In most scenarios, only a few links are congested, and an efficient link monitoring method should seek to *directly* identify only the congested links (and estimate the parameter of interest such as delay) instead of jointly estimating delays on *all* links only to identify the few congested ones.

Model	Accuracy	Measurements	Computation
Deterministic	Low bias, High variance	No special need	Less intensive
Stochastic	Low variance	Needs correlated measurements	intensive

Table 1 Trade-off between deterministic and stochastic models

Approach	Model	Measurements	Link attributes
Maximum likelihood tree[1]	Stochastic	Active	Delay and topology
Expectation maximization[2]	Stochastic	N/A	Delay
Minimum variance weighted average[3]	Stochastic	Active	Loss rates
Expectation maximization[4]	Stochastic	Active	Loss rates
Factor graph framework[5,6]	Stochastic	Active	Link status
Smallest consistent failure set[7]	Deterministic	Passive	Link status
Vector span based probe selection[8,9]	Deterministic	N/A	Link status
Maximum A Posteriori estimation[10]	Deterministic	Active	Link status
Minimum hitting set heuristic[11]	Deterministic	Active	Link status

Table 2 Previous works in network tomography

1.1 Deterministic Measurement Model

Accordingly, in this work we adopt a *deterministic* observation model due to its potential for yielding more efficient network monitoring algorithms. A brief comparative summary of the relevant literature inclusive of both stochastic and deterministic approaches is provided in Table.2. As we later discuss in detail, the main weakness of the deterministic approach is the issue of *identifiability* - without progress on this front, these will continue to remain of academic interest. Accordingly, in this work, we provide several new results concerning identifiability.

Assume we are given a set of end-hosts and paths between them for probing, where the paths are equivalently the routing matrix [1]. Our monitoring scheme operates in the following stages:
A. In the first stage, we seek to answer the following binary hypothesis: is there *at least one* congested link in the network?
B. If a positive determination is made in the above, then proceed to estimate the *number* of congested links in the network and identify their locations.

A given network is modeled as a graph $G(V,L)$ with a set of vertices V and edges/links L. The graph is presumed to have a set of defined *boundary nodes*; probes will be exchanged among pairs of boundary nodes along routes that are predetermined. The total number of links in the network (or cardinality of $|L|$) equals n, and a total of m routes between the boundary nodes is chosen (implying that a total of m end-to-end measurements are obtained). If a link l_i belongs to a path ϕ_j, the corresponding (j,i)-th entry in the routing matrix $\mathbf{A}$ equals unity, otherwise the remaining entries are zero.

The end-to-end (delay) measurements and individual link delays are then assumed to be related by the following linear model in the absence of any measurement noise:

$$\mathbf{Y} = \mathbf{AX} \tag{1}$$

where $\mathbf{Y}$ is the m-vector of real end-to-end measurements, $\mathbf{A}$ is the $m \times n$ binary routing matrix, and $\mathbf{X}$ is the n-vector of unknown link delays. In this model, the link

[1] The choice of these paths is themselves a design variable, however consideration of this is beyond the scope of this work.

delay is considered constant; in practice, these naturally vary over time. As a result, the deterministic model is assumed to apply during a period of 'stationarity', i.e., when network conditions are relatively stable.

To address the binary hypothesis problem in the first stage (does the network have a congested link?), we convert the above measurement model to an equivalent purely binary model via thresholding the end-to-end measurements. Based on available side information, the threshold is suitably chosen so as to classify all routes as normal or congested, resulting in a *binary* m-vector of measurements $\mathbf{Y}$. The corresponding binary n-vector $\mathbf{X}$ thus represents the status of individual links in the network, also classified as normal or congested.

1.2 End-to-end Measurements

The end-to-end measurements needed may be obtained via active or passive probes. Numerous tools exist for active and passive end-to-end measurements that can measure and report the delay and the packet loss rate on end-to-end routes [12].

For active probing, the end-hosts insert separate probe packets within data and measure the round trip time (RTT). Since probing constitutes overhead, sequential probing instead of batch is preferrable for various reasons, as discussed in Sec.3. Moreover, by using the information from results of previous probes, the number of probes needed to identify the congested links can be minimized via sequential approaches as compared to batch. For passive measurements, the end-hosts collect and analyze the existing data packets in the network. Most works in network tomography using passive measurements monitor the traffic between a server and its clients, which requires cooperation with the server [7, 13, 14]. On the other hand, to obtain delay and packet loss rates from passive measurements requires clock synchronization among nodes. If synchronization is achieved by, say, inserting an NTP timestamp into the header of data packets, using a sequential approach to process the collected data yields more computational efficiency than batch processing. While our method is agnostic to the mode of how end-to-end measurements are obtained, we assume active probing is employed.

1.3 Contributions

In this paper, we first develop a binary deterministic model for our network tomography approach in Sec. 2. Then Sec. 3 derives a sequential algorithm that requires fewer measurements than traditional batch algorithms. Sec. 4 defines the notion of identifiability as a necessary pre-requisite for our method and derives necessary and sufficient conditions. An algorithm is given to check identifiability of the sequential scheme on a given network. The results are verified by simulation in Sec. 5. Finally, the paper concludes with Sec. 6 and all proofs are deferred to the Appendix.

1.4 Related Works

Network tomography from end-to-end measurements is a well-known problem, dating back to [15]. In previous works summarized in Table 2 and in [12], end-to-end measurement results are obtained by sending probes simultaneously. However, this results in a sharp increase of network traffic that can alter the congestion profile in the network. Based on the binary deterministic model, a sequential algorithm is proposed that mitigates this problem.

Considerations of identifiability for tomography methods based on stochastic models have focussed on the distribution of $\mathbf{X}$ in multicast trees [1,3,16]. In [2], the issue of identifiability in deterministic models was discussed; however, it presumed that probes could be initiated from any node and use any route in the network, which is generally impractical. Our approach is similar in principle to that in [7] where the notion of "separability" was proposed in a binary model. A network is separable when a path is bad if and only if one of the links on this path is bad. This property ensures a. deterministic: the binary performance (good/bad) is experienced by all the packets on this link/path; b. binary: a bad path cannot arise because of a number of 'partially' bad links. It is proved that the performance models, e.g. general loss model and delay spike model in [14], can be binarized via a threshold to separable deterministic models. Thus, the binary deterministic model is practical. However, only the tree structure is considered in [7], and the sufficient and necessary conditions for methods using this model to correctly identify all the congested links are not given.

2 Model

Assume a network $G(V,L)$ with routing matrix $\mathbf{A}$. Denote the status of the link l_i to be

$$I_{l_i} = \begin{cases} 1 & if\ link\ l_i\ congested \\ 0 & if\ link\ l_i\ non-congested \end{cases} \tag{2}$$

Denote the status of a route ϕ_i to be

$$I_{\phi_i} = \begin{cases} 1 & if\ route\ \phi_i\ congested \\ 0 & if\ route\ \phi_i\ non-congested \end{cases} \tag{3}$$

Denote the set of all the links in the network as L, and denote the set of all the end-to-end routes (obtained with shortest-path algorithm) in the network as Φ.

We adopt the following assumptions, similar to the ones in [7]:

1. A route is bad if and only if at least one link on the route is congested. That is,

$$I_{\phi_i} = 1 \Longleftrightarrow \exists l_\alpha \in \phi_i, I_{l_\alpha} = 1 \tag{4}$$

In other words, we have

$$I_{\phi_i} = I_{l_{j_1}} + I_{l_{j_2}} + ... + I_{l_{j_k}}, \phi_i = \{l_{j_1}, l_{j_2}, ..., l_{j_k}\} \tag{5}$$

where '+' is the logic 'OR' operation. This leads to a binary deterministic model

$$\mathbf{Y} = \mathbf{AX} \tag{6}$$

where $\mathbf{Y} = [I_{\phi_1}, I_{\phi_2}, ..., I_{\phi_m}]^T$ is an $m \times 1$ binary vector obtained from probe measurements, $\mathbf{A}$ is the binary $m \times n$ routing matrix, and $\mathbf{X} = [I_{l_1}, I_{l_2}, ..., I_{l_n}]^T$ is an $n \times 1$ binary vector representing link status. In (6), the '$+$' is the logic 'OR' operation, and the '$*$' is the logic 'AND' operation.

2. All the links in the network are monitored by end-to-end measurements, i.e.,

$$\forall l_\alpha \in L, \exists \phi_i \in \Phi, s.t. l_\alpha \in \phi_i \tag{7}$$

In other words, there is no link that is not monitored via the set of routes chosen, implying there is at least one '1' entry in each column of $\mathbf{A}$.

3 Sequential Algorithm

In this section, a sequential measurements method using the model in Sec. 2 is proposed to minimize the overhead caused by probing. Our sequential technique consists of a two-stage approach. In the first stage, the algorithm detects if there exists at least one congested link in the network using a minimum number of probes or routes. To cover every link with a minimum number of measurements is shown to be the well-known 'minimum set cover' problem. When network congestion is detected, the second stage is invoked to determine the location of one of the congested links with the fewest number of measurements. This problem is solved using group testing methods. Then the algorithm moves back to the first stage until all the links are classified. The locations of all congested links are determined recursively, one at a time. The number of congested links detected is defined as the number of links being classified as congested when the algorithm terminates.

3.1 First Stage

In this stage, the method determines if there is a congested link in the network using a greedy algorithm for the set cover problem defined next.

Definition 1 (**Minimum Set Cover**) Given a universal set U of n elements and a collection of subsets of U denoted by $S = \{S_1, ..., S_k\}$, a cost function $c: S \to Q^+$, find a minimum cost sub-collection of S that covers all elements of U [17].

In the case of network tomography, the set U corresponds to the set of all (n) links and the collection of subsets S are all possible end-to-end measurement routes. The cardinality of this collection (the total number of all possible end-to-end routes) is assumed to be k. The cost function is $c: S \to Q^+$ is $c(S_i) = 1, \forall i \in \{1, 2, ..., k\}$, and denotes the number of probes sent per route (one). We seek a minimum number of subsets that covers all elements of U corresponding to a minimum cost sub-collection.

A suitable heuristic for this problem is a greedy algorithm, as shown in the flowchart in Figure 1. The heuristic first searches for a route that covers the *largest*

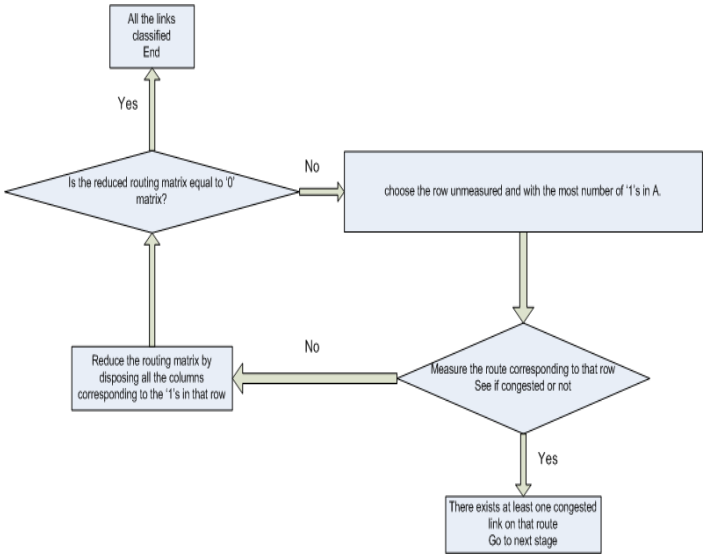


Fig. 1 Flow chart of the greedy algorithm

number of (remaining or uncovered) links. A probe is sent on this route; if the end-to-end delay measurement indicates that this route ϕ_c is good, all the links on this route are *classified* as non-congested and are removed from subsequent consideration by eliminating the corresponding columns in the routing matrix **A**. The algorithm iterates by searching for the next route that covers the largest number of uncovered links using the reduced routing matrix. If on the other hand, the result of the initial end-to-end probe measurement shows that this route ϕ_c is bad, it indicates at least one congested link l_c. The algorithm then goes onto the second stage to detect the location of the congested links.

As a baseline, the complexity of the greedy algorithm is shown in Figure.2 compared to a batch via a Matlab simulation for a grid network with no congested links. All nodes on the boundary of the grid are potential end-hosts and the routing table **A** is generated using the shortest path algorithm. The result shows the significant complexity improvement via the greedy algorithm that uses only $O(log(N))$ measurements [18] compared to the N measurements in the batch, where $N = \binom{N_0}{2}$, with N_0 being the number of end-hosts in the network.

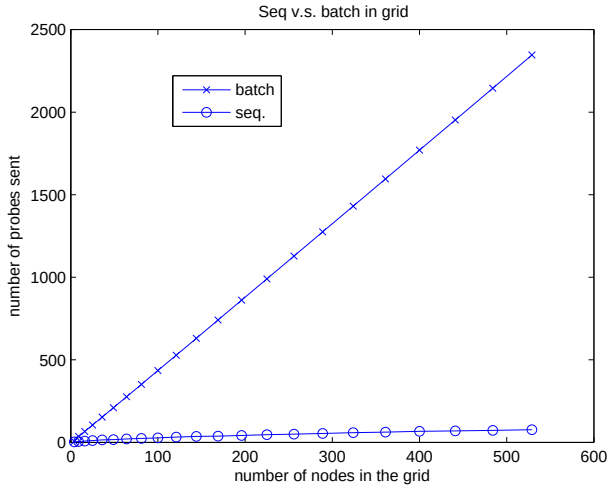


Fig. 2 The number of measurements in sequential compared to batch in grid networks: no congested links

3.2 Second Stage

If no congested route is detected in Stage I, the end-hosts stop any further probing. On the other hand, the algorithm moves into the second stage as soon as a bad route is detected in the first stage; the goal of the algorithm is to detect the location of one of the congested links with a minimum number of measurements. We show how the group testing technique can be applied to this problem.

In the first stage of the algorithm, a congested route is detected; assume there are n_0 links on this route ϕ_c. Then at least one of the n_0 links is congested. The n_0 links constitute the set of n_0 items in the group testing among which we wish to identify the $d \ll n_0$ "defectives" or the set of congested links. Each end-to-end measurement corresponds to a binary hypothesis test in the group testing framework with two possible outcomes: a) a good route (all the links on the route are non-congested); or b) a bad route (at least one link on the route is congested). The binary splitting algorithm in group testing can now be applied to find the location of the congested links efficiently, as described next.

A flow chart of the binary splitting algorithm is shown in Figure 3. The Stage I sequential measurements are conducted until a congested route is detected. Until then, the routing matrix $\mathbf{A}$ of the full network is successively reduced everytime the end-to-end measurements indicate a good route, i.e. all links on this route are non-congested and the corresponding columns in the routing matrix $\mathbf{A}$ are removed. All links not on this route ϕ_c are temporarily considered as not congested and the respective columns are removed from the reduced routing matrix $\mathbf{A_r}$. The remaining columns in $\mathbf{A_r}$ correspond to the links that are potentially congested. The splitting algorithm selects a sub-route ϕ within ϕ_c that covers approximately half the links and sends a probe for an end-to-end measurement. The columns corresponding to the links on ϕ are not to be removed if ϕ is found congested. Otherwise, the columns corresponding to the

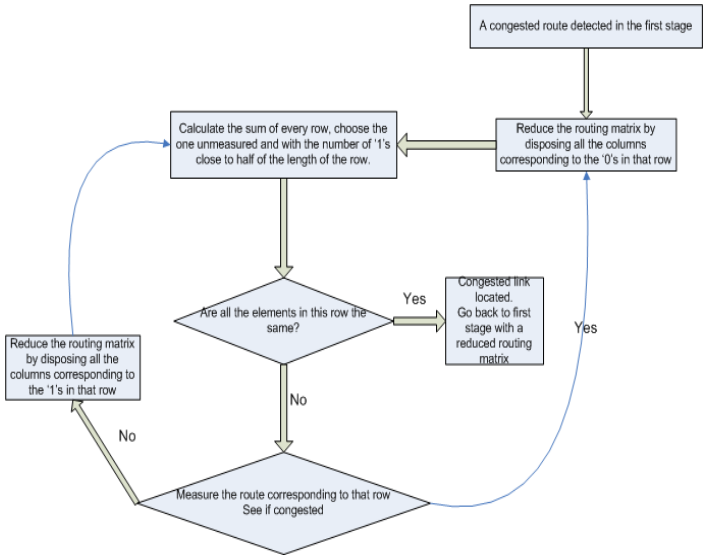


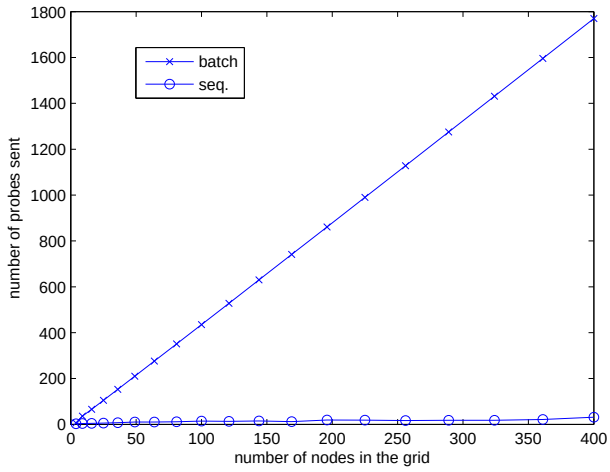
Fig. 3 Flow chart of the binary splitting algorithm

remaining links (those not on ϕ) are eliminated from $\mathbf{A_r}$, in order to first locate the congested links within ϕ. This process iterates until all the rows in $\mathbf{A_r}$ are either all '1's or all '0's. The all '0's rows correspond to the routes that do not cover the congested link l_c. And the all '1's rows correspond to the routes that cover the congested link l_c. The links corresponding to the columns left in $\mathbf{A_r}$ are *classified* as congested. The number of measurements needed to find d congested links is $d\lceil log_2(n_s)\rceil$ [19], where n_s is the number of links on the route ϕ_c.

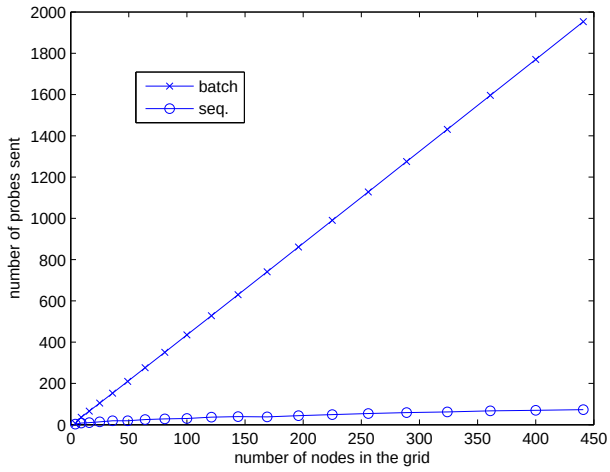
After this step, the algorithm reduces the routing matrix $\mathbf{A}$ to account for processing of ϕ_c and the process iterates until all the links in the network are classified, OR all the possible routes have been probed. If all the possible routes have been probed and some links $\{l_n\}$ are not classified, these are determined to be congested. Because if $\exists \phi \in \Phi, s.t. I_\phi = 0, l_n \in \phi$, then l_n is classified as non-congested. Therefore, $\forall \phi \in \Phi, s.t. l_n \in \phi \Longrightarrow I_\phi = 1$ and the remaining l_n should be classified as congested.

3.3 Number of Measurements

The performance of the algorithm with two stages on grid networks and with a single congested link in the network, is depicted in Figure 4 (a). The same grid network and routing matrices, as in the first stage, are generated in Matlab. A link in the network is randomly chosen to be congested according to a uniform distribution. The proposed algorithm is used to identify the location of the congested link. For each grid network,



(a)



(b)

Fig. 4 (a) Number of measurements needed for one random link failure in the grid networks (b)Number of measurements needed for two random link failures in the grid networks

the experiment is repeated ten times and the average number of measurements needed to identify the congested link (over ten tests) is determined. The performance of the algorithm for two random link failures is shown in Figure 4 (b). The number of probes needed is nearly the same for the one and two link failures in grid networks. A demo of the experiments is available @ http://www.ee.washington.edu/research/funlab/network_coding/.

3.4 Examples

Example 1 Consider the four nodes network in Figure 5 with three boundary nodes. The routing matrix is shown in Figure 5 (b). This network is sufficiently simple that the performance of the sequential algorithm can be analyzed explicitly.

Scenario 1 - No congestion: In this case, the sequential algorithm only executes the first stage. The greedy algorithm chooses the route between End1 and End2, since all the routes contain an equal number of links. A probe is sent on this route and since the result of the measurement is "not congested", l1 and l2 are classified as non-congested. The routing matrix $\mathbf{A}$ is then reduced to $\begin{matrix} 1\to 2 \\ 1\to 3 \\ 2\to 3 \end{matrix}\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$, and the only column left in the matrix corresponds to l3. Then the route between End1 and End3 is chosen, since it is one of the longest routes left (containing one link). After measuring this route, l3 is classified as non-congested, completing the algorithm. In summary, two probes are sent in the sequential method instead of three being sent in the batch method and concludes that no links in the network is congested.

Scenario2 - l1 congested: In this case, the process is shown in Figure 6. The sequential method uses three probes to identify l1 as the only congested link. Similarly, for the scenarios where l2 or l3 is congested, we can obtain the correct location of the congested link, using two probes instead of three in batch.

Scenario3 - l1 and l2 congested: In this case, the same steps are executed as in Scenario 2, except that for the last measurement on the route between End2 and End3, the result obtained is "congested". Therefore, the method classifies all the three links as congested, which is incorrect. This indicates a limitation of our method, i.e., whenever two links are congested in this network, the sequential method is unable to correctly identify the links. The reason for this is discussed in Sec.4.

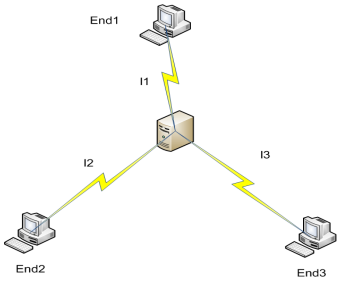
Scenario 4 - all links congested: The same steps are executed in this scenario as in Scenario 3. The sequential method correctly identifies all links as congested.

Example 2 The topology of the University of Washington's Electrical Engineering network is shown in Figure 7. The sequential method uses 7 probes instead of 78 probes in batch to correctly identify the 'no congestion' scenario. It also uses less probes to correctly identify all possible 'one congested link' scenarios.

4 Identifiability

As shown in the previous section, whether the sequential algorithm can correctly identify the number and locations of congested links depends on the topology and the number of congested links in the network. Clearly, the *identifiability* of individual link status from end-to-end measurements is an important consideration that is explored next [3]. It is proved that the false negative rate of the sequential algorithm is zero. Necessary and sufficient conditions on any approach to achieve a zero false positive rate are also given. Finally, an algorithm to check if the sequential algorithm can achieve a zero false positive rate for a given number of congested links is proposed.

For the binary observation model in Eq. 6, we next define the identifiability.



(a)

$$\begin{array}{c} 1\to 2 \\ 1\to 3 \\ 2\to 3 \end{array}\begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

(b)

Fig. 5 A 4-nodes network (a)Network topology (b)Routing matrix

$$\begin{array}{c} \\ End1\to End2 \\ End1\to End3 \\ End2\to End3 \end{array}\begin{array}{c} \begin{array}{ccc} l1 & l2 & l3 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \end{array}$$

End1 → End2 congested

⇒

move into 2nd phase

$$\begin{array}{ccc} l1 & l2 \\ \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \end{array}$$

End1 → End3 congested

⇒

$l1$ congested, move back to 1st phase

$$\begin{array}{c} \\ End2\to End3 \end{array}\begin{array}{c} \begin{array}{cc} l2 & l3 \end{array} \\ \begin{bmatrix} 1 & 1 \end{bmatrix} \end{array}$$

End2 → End3 normal

⇒ all links classified

l2 and l3 are non - congested

Fig. 6 Process of Scenario2: l1 congested

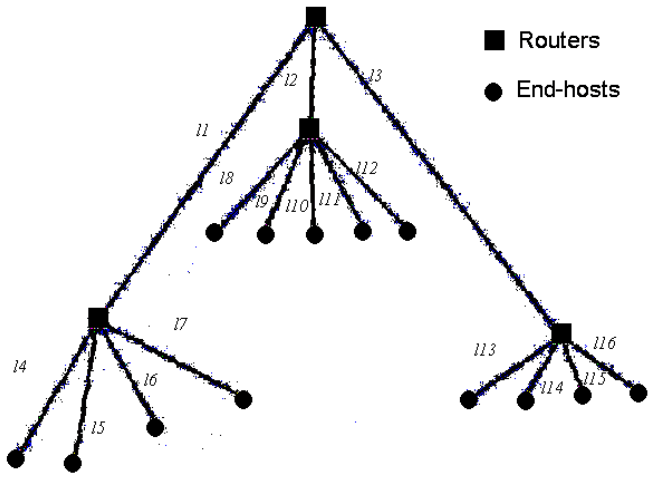


Fig. 7 Topology of the University of Washington's Electrical Engineering Network

Definition 2 (Identifiability)
A network with routing matrix **A** is *i*-identifiable, if and only if

$$\forall \mathbf{X}_1, \mathbf{X}_2, \; s.t. \; <\mathbf{X}_1> = <\mathbf{X}_2> = i \Rightarrow \mathbf{A}\mathbf{X}_1 \neq \mathbf{A}\mathbf{X}_2, \tag{8}$$

where $<\mathbf{X}>$ is the cardinality (the number of non-zero elements) in **X**. **Y**,**A**,**X** are all binary, as defined in (6).

The definition states that if *i* congested links can be uniquely identified from the end-to-end measurements, then the network is *i*-identifiable. Clearly, checking identifiability is a combinatorial problem that is NP-hard. An example of a 1-identifiable network is shown in Figure 5. As shown in Figure 8, if any one of the three links is congested, individual link status can be uniquely determined from the end-to-end measurements. However, as shown in Figure 9, the scenario where l_1 and l_2 are congested gives the same measurement result as the scenario where l_1 and l_3 are congested. Therefore, we cannot tell which two links are congested according to the end-to-end measurements. So this network is not 2-identifiable, because $\mathbf{X}_1 = [1\ 1\ 0]^T, \mathbf{X}_2 = [1\ 0\ 1]^T, <\mathbf{X}_1> = <\mathbf{X}_2> = 2$, but $\mathbf{A}\mathbf{X}_1 = \mathbf{A}\mathbf{X}_2$. This is the reason why both sequential and batch methods fail to identify the congested links, as shown in the previous section.

Any method based on the binary deterministic model in Sec.2 requires the network to be i-identifiable, in order to correctly identify the number and location of *i* congested links. Therefore, the sufficient and necessary conditions for a network to be i-identifiable are of great interest.

Suppose we can only measure the routes between an arbitrary pair of end-hosts, and the end-hosts are the boundary nodes of the network, whereas other nodes are intermediate nodes (routers). From (6), whether we can obtain the link status from the results of measurements depends on the routing matrix **A**. Thus, whether a network is

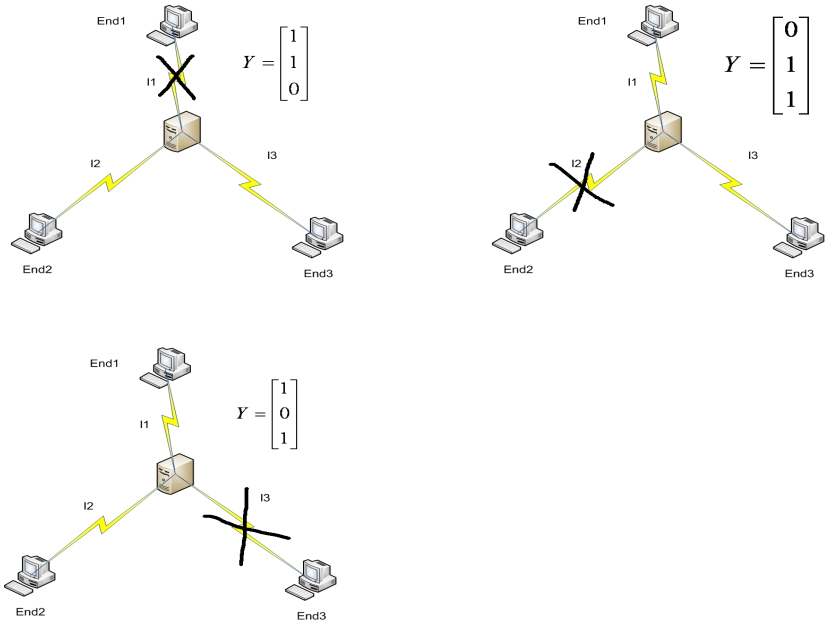


Fig. 8 A 1-identifiable network: three conditions of 1 link congested

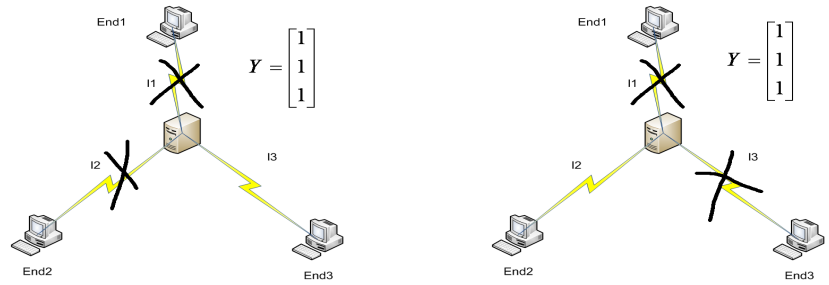


Fig. 9 The network is not 2-identifiable

i-identifiable relies on the properties of $\mathbf{A}$. We next provide some insight into how the structure of the routing matrix $\mathbf{A}$ impacts i-identifiability, via the following necessary condition.

Theorem 1 A network with routing matrix $\mathbf{A}$ is i-identifiable only if all $i+1$ columns of $\mathbf{A}$: $a_{j_1}, a_{j_2}, ..., a_{j_{i+1}}$ are linearly independent under modulo 2 operations.

A set of i linearly independent columns of the routing matrix $\mathbf{A}$ can result in a degree i node in the network, which immediately leads to the following:

Corollary 1 A network with a degree i intermediate node is not (i-1)-identifiable.

This suggests an easier method to check network identifiability - by looking at the degree of the intermediate nodes. For example, in Figures 8 and 9, the network has a degree 3 intermediate node and is thus not 2-identifiable.

Theorem 2 If a network is not i-identifiable, then it is not j-identifiable, $\forall j \geq i, j \neq n$, where n is the total number of links in the network.

By Theorem 2, any algorithm for the binary deterministic model in (6) can detect (at most) the i congested links if the network is i-identifiable but not $i+1$-identifiable. For any j congested links with $j < i$, the algorithm can identify all the congested links correctly.

Combining Theorem 2 and Corollary 1, the following can be derived, which serves as an upper bound on the identifiability of a network.

Corollary 2 Any network with an intermediate node is not (n-1)-identifiable, where n is the total number of links in the network.

Since in most cases, congested links are rare compared to the total number of links, it is useful to focus on i-identifiability with $i \ll n$. Necessary and sufficient conditions for 1-identifiability can be derived and is given in the following theorem.

Theorem 3 A network $G(V,L)$ is 1-identifiable if and only if the routing matrix $\mathbf{A}$ that corresponds to the chosen routes does not contain any identical columns.

For 1-identifiability, there exists an interesting connection to the error control coding theory. Rewrite (6) as $\mathbf{Y}^T = \mathbf{X}^T\mathbf{A}^T$ and treat the rows of the $\mathbf{A}^T$ matrix as binary codewords, However, the dimension m of the codewords is (in contrast to traditional error control coding) smaller than the dimension n of the information vector. 1-identifiability is now seen as requiring that the Hamming distance between the rows of $\mathbf{A}^T$ be at least 1, so as to allow detectability of all 1-error patterns. This is equivalent to "no 2 rows should be identical" as in theorem 3. However, for $i \geq 2$, this connection no longer holds, since the '+' and '*' in (6) are logic "OR" and "AND" instead of modulo 2 operations.

Note that a network without degree 2 nodes is *not* always 1-identifiable as claimed in [1]. The following necessary condition can be proved for 1-identifiability.

Theorem 4 Any graph $G(V,L)$ has at least one common node with degree 2 in all the multicast trees rooted in the end-hosts, whenever the routing matrix $\mathbf{A}$ has identical columns.

This is illustrated in Figure 10. Although the network itself does not contain any degree 2 nodes, it is not 1-identifiable, because node 4 is a degree 2 node in all the multicast trees rooted in end-hosts.

After examining the necessary and sufficient condition for 1-identifiability, we give the following theorems as necessary and sufficient conditions for i-identifiability in aspect of the topology of the network, $\forall i \geq 1$.

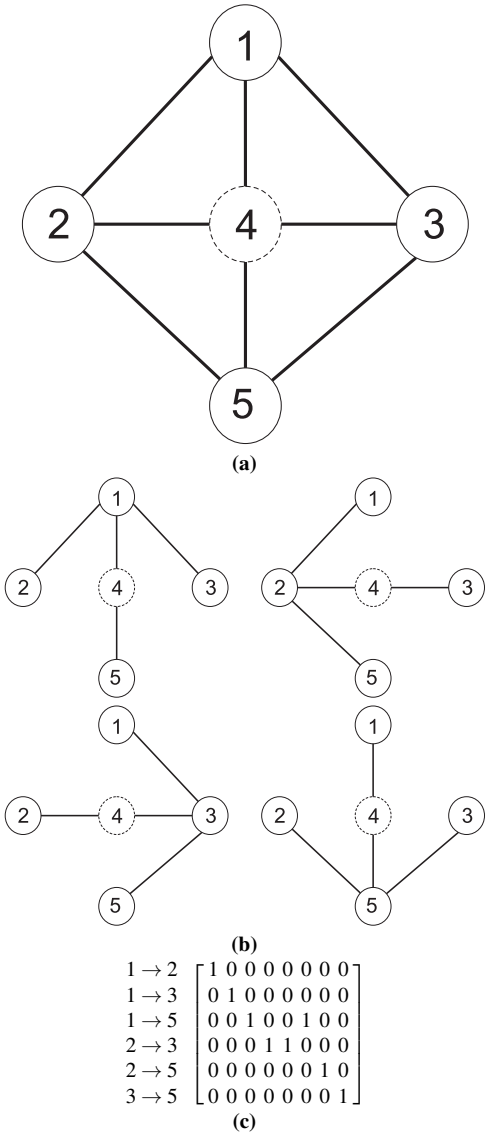


Fig. 10 An example of a network that is not 1-identifiable (a)Network topology (b)Multicast trees rooted at boundary nodes (c)Routing matrix.

Theorem 5 A network is i-identifiable if:

$$\forall l_{j_1}, l_{j_2}, ..., l_{j_{i+1}} \in L, \exists \phi \in \Phi, s.t. l_{j_1} \in \phi, l_{j_2}, ..., l_{j_{i+1}} \notin \phi \tag{9}$$

This sufficient condition means that for any i congested links L_i, for any link $l \notin L_i$, there is a route that covers l, but does not cover any links in L_i. Therefore,

the link l can be separated from the i links in L_i. Otherwise, because link l cannot be measured without measuring links in L_i, if all the i links in L_i have failed, the status of l can no longer be measured. Using our algorithm in Sec. 3, l will be classified as congested. When l is not congested, the proposed scheme will give a false alarm.

This condition is in accordance with the d-disjunct definition in group testing [19]. By this definition, the routing matrix $\mathbf{A}$ of a network $G(V,L)$ is d-disjunct if for any d+1 links $l_0, l_1, ..., l_d$ of $G(V,L)$, there exists a row, or a route, containing l_0, but not $l_1, ..., l_d$. A d-disjunct matrix can identify all congested links in a network with (at most) d congested links in a very simple way that a link is not failed if and only if it is covered in a good route [20]. Therefore, for the sequential algorithm, if a link is congested, it is always classified as congested. In other words, the false negative rate for the sequential algorithm is 0. For the false positive rate,

$$n_{detected} \geq n_{congested} \tag{10}$$

where $n_{detected}$ is the number of congested links detected in the sequential method, and $n_{congested}$ is the number of congested links in the network. The equality is reached when the network is $n_{congested}$-identifiable.

As shown in Figure 9, $\forall \phi, l_1 \in \phi \Longrightarrow l_2 \in \phi$ or $l_3 \in \phi$. Therefore, the 2-identifiability sufficient condition is not satisfied for this network.

The necessary condition for i-identifiability can be expressed in a similar way as Theorem 5.

Theorem 6 A network is i-identifiable only if:

$$\begin{aligned} &\forall l_{j_1}, l_{j_2}, ..., l_{j_{i+1}} \in L, \exists \phi \in \Phi, \\ &s.t. l_{j_1} \notin \phi, \exists l \in \{l_{j_2}, ..., l_{j_{i+1}}\} \in \phi \end{aligned} \tag{11}$$

The flow chart for checking the sufficient condition for a known routing matrix $\mathbf{A}$ is shown in Figure 11. First, $i+1$ columns should be selected from $\mathbf{A}$. Second, select i columns out of the $i+1$ columns and add them with the logic OR operation. Third, compare the sum with the column remaining after the second step. If the location of all the non-zero elements in the sum is a subset of the location of the non-zero elements in the non-selected column, the network does not satisfy the sufficient condition. This process iterates until all the possible combinations of the $i+1$ columns are checked.

A special case of the necessary and sufficient conditions given in Theorem 6 and 5 arises for 1-identifiability, since these can be combined to yield necessary and sufficient conditions. According to the theorems, a network is not 1-identifiable if and only if

$$\exists l_\alpha, l_\beta \in L, s.t. l_\alpha \in \phi_i \Longleftrightarrow l_\beta \in \phi_i \tag{12}$$

This can be proved using Theorem 3 and yields the following.

Corollary 3 A network satisfies (12) if and only if there exists at least two identical columns in the routing matrix $\mathbf{A}$.

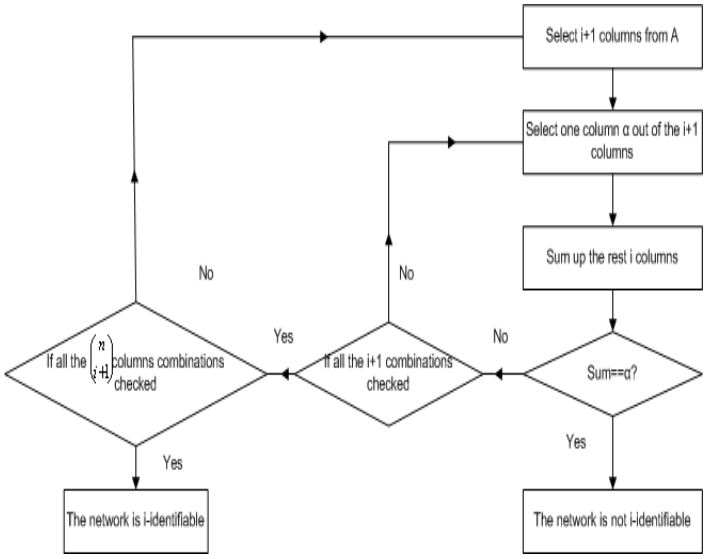


Fig. 11 Flowchart: check the identifiability of a network

Using the necessary and sufficient conditions allows us to check the identifiability of a network. For example, the necessary and sufficient condition for a network $G(V,L)$ with routing matrix $\mathbf{A}$ to be 2-identifiable is

$$\begin{aligned} \forall a_{j_1}, a_{j_2}, a_{j_3}, a_{j_4}, a_{j_1} + a_{j_2} \neq a_{j_3} + a_{j_4}, \\ \textit{and } a_{j_1} + a_{j_2} \neq a_{j_2} + a_{j_3}, \end{aligned} \tag{13}$$

where a_j is a column in $\mathbf{A}$, and '+' is the logic 'OR' operation. Thus, every two columns in the routing matrix need to be summed up and the resultant compared to decide if the network is 2-identifiable. The number of operations needed in calculation is $\binom{n}{2} * m + \left(\binom{n}{2} \right) * m = O(n^4 m)$. Checking the necessary and sufficient conditions given in Theorem 6 and Theorem 5 requires $\binom{n}{3} * m * m = O(nm^2)$ operations. Hence, when $m \approx n$, checking identifiability of a network with the latter conditions offers significant reductions in the number of computations needed.

5 Simulations and Experiments

In this section, we implement the sequential scheme described in Sec. 3 using $OPNET^{TM}$ 14.5. The simulation results from applying the proposed sequential scheme to the test scenario in Figure 7 are presented. Two sets of scenarios are used for comparison: scenarios with different links congested; and scenarios with different background traffic loads. Experiments on the PlanetLab Testbed are used to validate the proposed scheme in real networks.

5.1 Simulation Environment

The proposed scheme is applied to the topology as shown in Figure 12 and resembles the University of Washington's Electrical Engineering network. The figure shows thirteen end-hosts that are all connected through 100 Mbps Ethernet links to backbone routers. OPNET is used for testing the identifiability, which can be done for small networks. This topology is 1-identifiable but not 2-identifiable, as there is a degree 3 intermediate node (router) in the network. The proposed detection scheme is implemented in the application layer at the hosts. Each probe is defined as a "phase" in custom applications deployment in OPNET at the end-hosts. In each phase, a probe using IP packets is sent from a sender to a receiver, and the receiver returns a response packet to the sender. A time-out of five seconds is set for each phase. If a phase experiences a time-out, the route for the probe is identified as "bad"; otherwise, it is classified as "good". The delay of the probes is obtained in OPNET using the Discrete Event Simulation (DES) statistics variable: custom application phase response time. The background traffic is set to be full-mesh IP unicast between all the nodes in the network. The delay of the background traffic is tracked by the DES statistics variable "background traffic delay" that captures the average end-to-end delay of background traffic.

5.2 Validation

Figure 13 shows the average background traffic delay (obtained from successfully received packets) when none of the links are congested and the background traffic rate between any two nodes in the network is the value on the x-axis. The batch measurement is set to be IP unicast packets between any two end-hosts. As shown in the figure, the background traffic delay with batch measurements is higher than with sequential measurements. First, it takes less time for the routers to process the probes, since fewer probes are sent in the sequential mode. Second, because the probes are sent sequentially, the impact of probe insertion on the background traffic delay is less than in batch mode, where all the probes are sent simultaneously. This difference is expected to scale with network size. Also, as the background traffic reaches the link capacity, the delay begins to increase sharply, as the links between the routers become congested.

The detection results for the topology in Figure 7 is listed in Table 3. Background traffic with a rate higher than the link capacity is created on both directions of a link

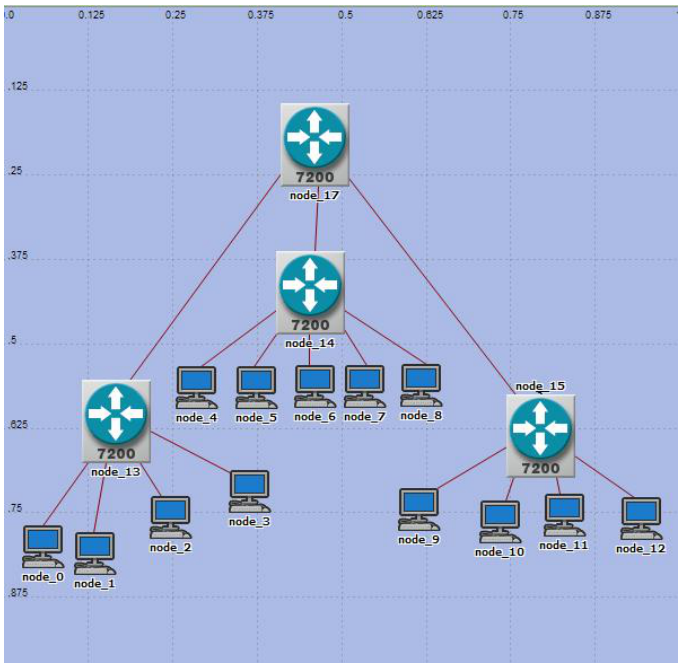


Fig. 12 OPNET implementation of the proposed link failure monitoring scheme for the UWEE network in Figure 7

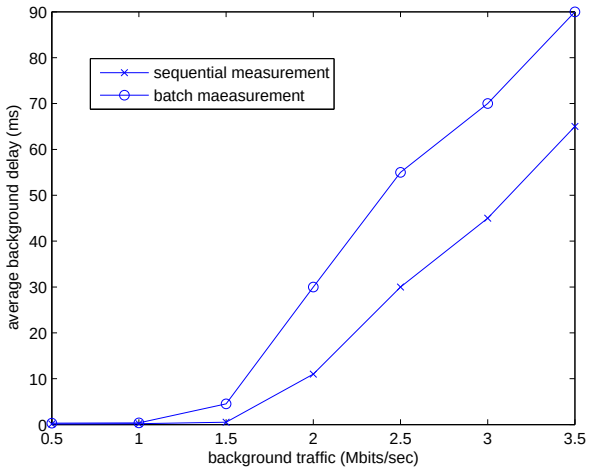


Fig. 13 The comparison of sequential and batch measurement on background delay

Congested Link	l_1	l_2	l_5	l_{10}	l_{16}	l_1, l_2	l_1, l_5	l_1, l_{10}	l_1, l_2, l_3	l_1, l_2, l_9
Detection Result	l_1	l_2	l_5	l_{10}	l_{16}	l_1, l_2, l_3	l_1, l_5	l_1, l_{10}	l_1, l_2, l_3	l_1, l_2, l_3, l_9

Table 3 Detection results for different link congestion locations in the simulation

to congest it in the simulation. As shown in the table, single link congestion can be identified with the proposed scheme and the network is 1-identifiable. However it is not 2-identifiable, because in the scenario where the links l_1 and l_2 are congested, the detection result is not correct. According to Theorem 2, it is not 3-identifiable. This is verified in the simulation, where in the scenario, the links l_1, l_2, and l_9 are congested and the proposed scheme identifies the "good" link l_3 as congested.

5.3 PlanetLab Experiments

We demonstrate our two-stage sequential scheme using 30 nodes of PlanetLab (www.planet-lab.org), located at universities around the world: 2 in South America, 3 in Asia, 1 in Europe, and 24 in the States. The major issue with network tomography is to determine the routing matrix that changes over time as network conditions vary. In our experiments, the link congestion status was observed to be stable in a one-hour window, but varied day-to-day. Therefore, in a one-hour window, the link status stability assumption, as in (1), can be justified.

Internal link status is difficult to observe from the end-hosts, without compromising the routers. In addition, some routers do not respond to "traceroute" and hence the links connecting these routers cannot be identified. As in [10], we use traceroute to identify network topology, routing matrix, and link status. Over 200 links are identified, and the anonymous routers are ignored in the experiment. The part of the topology we measured is shown in Figure 14. Note that this real-life network is highly connected and the tree model used in many network tomography papers is not appropriate [1,7]. Also, the methods based on tree topology are not applicable without multicast techniques in general. On the other hand, due to the limited number of end-hosts, the boundary links are isolated. Since the network contains some degree 2 intermediate-nodes, it is un-identifiable (identifiability 0).

It is observed that links may be asymmetric - links can be congested in one direction, while behaving normally on the other. Further, a link can be congested on one route, but normal on others. Note miss detection may occur because link status is not stable, which is different from the theorem indicated in Sec.4. For example, traceroute from host "princeton" to host "urochester" is completed with 11 hops, where the last hop is from "128.151.251.1" to "urochester". However, traceroute from host "ketsu" to host "urochester" cannot be completed within 30 hops. The last reported router on this route is "128.151.251.1". This was repeated 10 times, before and after the traceroute experiment between "princeton" and "urochester", and the results were observed to be the same. In our experiment, link status is inferred from the results of the traceroute as follows: i. Observe that a traceroute probe cannot reach its destination in 30 hops; ii. Compare the shown route with the route between a third host and the destination, and complete the routing table; iii. Mark the first unreach-

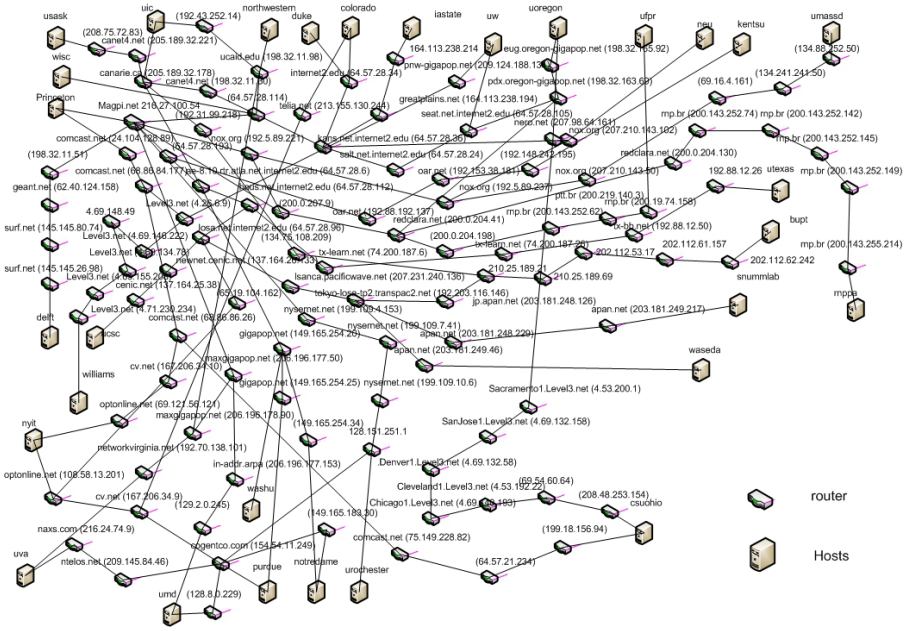


Fig. 14 A Part of the network topology with 30 end-hosts (some routers and links are not shown in the figure)

able link in this routing table as a congested link. For the above discussed "ketsu" to "urochester" route, the link from "128.151.251.1" to "urochester" is identified as congested. 2 congested links are identified.

We then calculate which route to probe next using our proposed algorithm. For each selected route, two probes are sent using a "ping -c 2 address". If at least one of the probes is received, the route is identified as normal. With a single probe (i.e. ping -c 1 address), the results show high variance: probes may be lost on a normal route with a small probability. Because we are using a deterministic model, this variance can result in false positives (i.e. identifying normal links as congested). There exists a trade-off between the false positive rate and the number of probes (the volume of probing traffic). After the algorithm terminates, 3 links are identified as congested, including the 2 identified with the traceroute. The detection rate is 100%. The algorithm outputs a false positive, because the identifiability of this network is 0.

With the proposed scheme, 51 routes are measured instead of 435 routes in batch scheme, resulting in an 88% saving relative to batch approach. Moreover, in the sequential scheme, the result from a measurement is sent to a control node, and the control node decides which route will be measured next. In contrast, in the batch scheme, all results from the end-to-end measurements are sent to the control node almost simultaneously. The sequential scheme not only uses fewer probes, but also avoids the potential congestion caused by the measurement reports.

6 Conclusion

In this paper, a binary deterministic model in network tomography is developed. A two-stage algorithm is proposed, incorporating sequential active probing, based on this model. A greedy algorithm in conjunction with binary splitting is used to minimize the number of probes. The notion of identifiability is defined to evaluate the performance of the algorithm. The necessary and sufficient (n.s) conditions of identifiability are summarized. Our algorithm provides significant reduction in the number of probes needed for identifiability of the link status when the necessary and sufficient conditions are satisfied. On the negative side, the model and algorithms based on it suffer from the high variance of estimation in deterministic models. As discussed in the introduction, in the binary deterministic model, it is assumed that all the packets on a link experience the same good/bad performance. This is not always true, as the delay and loss rate of a link varies over time. To make the measurements more accurate, several probes can be sent on the same route in a single end-to-end measurement, their RTT is averaged and compared to a threshold to obtain the status of the link. By adjusting the number of probes sent in a single measurement, there exists a trade-off between the probing traffic loads on the network and the accuracy of the congestion detection. A complete analysis of this trade-off is left for future work.

Acknowledgements This work was supported in part by the National Science Foundation Graduate Research Fellowship to the 1st author under Grant No. DGE-0718124.

7 Appendix

Proof of Theorem 1:

Denote the kth element in a_j as $a_j(k)$. If $a_{j_1} \bigoplus a_{j_2} \bigoplus ... \bigoplus a_{j_{i+1}} = O$, then $(a_{j_1} + a_{j_2} + ... + a_{j_{i+1}}) mod 2 = O$, where O is a $m \times 1$ zero vector.

So $\forall k \in \{1,2,...,m\}$, we have $a_j(k) = 0, \forall j \in \{j_1, ... j_{i+1}\}$, or $\exists j_\alpha \neq j_\beta, s.t. a_{j_\alpha}(k) = a_{j_\beta}(k) = 1$.

And there is at least one '1' in each column of $\mathbf{A}$ according to (7) in Assumption2. That is, $\forall j \in \{j_1, ..., j_{i+1}\}, a_j \neq O$.

Then$\forall k \in \{1,2,...,m\}$, $(a_{j_1}(k) + a_{j_2}(k) + ... + a_{j_i}(k)) > 0$ and $(a_{j_2}(k) + a_{j_3}(k) + ... + a_{j_{i+1}}(k)) > 0$, or $(a_{j_1}(k) + a_{j_2}(k) + ... + a_{j_i}(k)) = 0$ and $(a_{j_2}(k) + a_{j_3}(k) + ... + a_{j_{i+1}}(k)) = 0$.

So $a_{j_1} + a_{j_2} + ... + a_{j_i} = a_{j_2} + a_{j_3} + ... + a_{j_{i+1}}$, where '+' is the logic OR operation.

Then if $\mathbf{X}_1(k) = 1, \forall k \in \{j_1, ..., j_i\}$; $\mathbf{X}_1(k) = 0, \forall k \notin \{j_1, ..., j_i\}$ and $\mathbf{X}_2(k) = 1, \forall k \in \{j_2, ..., j_{i+1}\}$; $\mathbf{X}_1(k) = 0, \forall k \notin \{j_2, ..., j_{i+1}\}$, $\mathbf{AX}_1 = \mathbf{AX}_2$.

Therefore, there exists $\mathbf{X}_1 \neq \mathbf{X}_2, s.t. \mathbf{AX}_1 = \mathbf{AX}_2$ and $< \mathbf{X}_1 > = < \mathbf{X}_2 > = i$.

□

Proof of Corollary 1:

Because the shortest path routing algorithm is used and cycles are not allowed on any end-to-end route, for the sub routing matrix of the i links connected to the degree

i node, the columns always satisfy $a_1 \bigoplus a_2 \bigoplus ... \bigoplus a_i = 0$. Therefore, the network is not (i-1)-identifiable.

□

Proof of Theorem 2:

If a network is not i-identifiable, then $\exists \mathbf{X}_2 \neq \mathbf{X}_1, < \mathbf{X}_2 >=< \mathbf{X}_1 >= i, \mathbf{AX}_1 = \mathbf{AX}_2$. Denote the index of the '1's in $\mathbf{X}_1$ and $\mathbf{X}_2$ as L_1, L_2, respectively.

If $\exists \alpha \in \{1,2,...,n\}, \alpha \notin L_1, L_2$, then $\mathbf{AX}'_1 = \mathbf{AX}'_2, < \mathbf{X}'_2 >=< \mathbf{X}'_1 >= i+1$, where the index of the '1's in $\mathbf{X}'_1$ is $L'_1 = L_1 \bigcup \alpha$ and the index of the '1's in $\mathbf{X}'_2$ is $L'_2 = L_2 \bigcup \alpha$. Therefore, the network is not (i+1)-identifiable.

If $\nexists \alpha \in \{1,2,...,n\}, \alpha \notin L_1, L_2$, because $i+1 < n, < L_1 \bigcap \overline{L_2} >\geq 2. \forall \alpha \in L_1 \bigcap \overline{L_2}$ and $\beta \in L_2 \bigcap \overline{L_1}$, $\mathbf{AX}'_1 = \mathbf{AX}'_2, < \mathbf{X}'_2 >=< \mathbf{X}'_1 >= i+1$, where the index of the '1's in $\mathbf{X}'_1$ is $L'_1 = L_1 \bigcup \alpha$ and the index of the '1's in $\mathbf{X}'_2$ is $L'_2 = L_2 \bigcup \beta$. Therefore, the network is not (i+1)-identifiable.

By induction, $\forall j \geq i, j \neq n$, the network is not j-identifiable.

□

Proof of Corollary 2:

The degree of the intermediate node v is $deg(v) \leq n$. The network is not $(deg(v) - 1)$-identifiable. Therefore, it is not (n-1)-identifiable.

□

Proof of Theorem 3:

If: If a graph is not 1-identifiable, then $\exists \mathbf{X}_1, \mathbf{X}_2, where \ < \mathbf{X}_1 >=< \mathbf{X}_2 >= 1, and \ \mathbf{AX}_1 = \mathbf{AX}_2$. Because there is only one non-zero element in both $\mathbf{X}_1$ and $\mathbf{X}_2$, we denote the index of the non-zero elements α_1 and α_2. Therefore, under the logic OR and AND operation, $\mathbf{Y}_1 = \mathbf{AX}_1 = a_{\alpha_1}$ and $\mathbf{Y}_2 = \mathbf{AX}_2 = a_{\alpha_2}$, where a_{α_1} and a_{α_2} are the corresponding columns in the routing matrix $\mathbf{A}$. Because $\mathbf{Y}_1 = \mathbf{Y}_2$, $a_{\alpha_1} = a_{\alpha_2}$. So the routing matrix $\mathbf{A}$ has identical columns. Therefore, a graph is 1-identifiable if the routing matrix does not contain any identical columns.

Only If: Suppose there are two identical columns a_i and a_j in the routing matrix $\mathbf{A}$. Then a_i and a_j are linearly dependent under modulo 2 operations. Thus, the network is not 1-identifiable. So a graph is 1-identifiable only if the routing matrix $\mathbf{A}$ does not contain any identical columns.

□

Proof of Theorem 4:

Suppose L_{id} is the set of links whose corresponding columns in $\mathbf{A}$ are identical. According to the definition of $\mathbf{A}$, columns $l_i \in L_{id}$ are also identical in $\mathbf{A_s}$, $\forall s \in S$, where $\mathbf{A_s}$ is the routing matrix for the multicast tree rooted at s. That means, for each $l_i, l_j \in L$, $l_i \in \phi$ if and only if $l_j \in \phi, \forall \phi$.

Case A: Suppose there exists $l_i, \ l_j \in L_{id}$ such that $l_i \cap l_j = v \in V$.
We claim v has a degree of 2 in the multicast tree rooted in s.

By Contradiction: Assume that the above is not true; then $\exists v' \in V$ such that $(v,v') \in L$, and $(v,v') \neq l_i$, $(v,v') \neq l_j$, because $\exists \phi \ s.t. (v,v') \in \phi$. But ϕ cannot contain both l_i and l_j, as a route cannot cover three links connected to the same node in a multicast tree.

Case B: Now suppose there is no l_i, $l_j \in L_{id}$ with a node in common. Because $\exists \phi, s.t. l_i, l_j \in \phi$ and they have no node in common, there is a link $l \in L, l \in \phi$ that has a common node v with l_i and is on the path between l_i and l_j. We claim that $l \in L_{id}$.

By Contradiction: Suppose it is not; then there exists another link $l' \in L$ that is on the path between l_i and l_j, so that the columns corresponding to l and l_i in the routing matrix $\mathbf{A}$ are not identical. In this case there are two paths between l_i and l_j: one through l and the other through l'. This is a contradiction to that there is no cycle in a tree structure. So l belongs to set L. This is a contradiction because we assumed there is no l_i, $l_j \in L_{id}$ with a node in common.
Thus, if the routing matrix $\mathbf{A}$ has identical columns, then there is at least one node $v \in V$ with degree 2 in *all* the multicast trees. □

Proof of Corollary 3:

If: Given there are at least two identical columns in the routing matrix. Denote two links corresponding to identical columns as l_α and l_β. Then if $l_\alpha \in \phi_i$, $a_\alpha(i) = 1$. Because the two columns are identical, $a_\beta(i) = 1$ as well, which indicates $l_\beta \in \phi_i$. Similarly, we can prove $l_\alpha \notin \phi_i \Longrightarrow l_\beta \notin \phi_i$. Thus, if there exists at least two identical columns in the routing matrix, (12) is satisfied.

Only if: Because $l_\alpha \in \phi_i \Longleftrightarrow l_\beta \in \phi_i$, so $a_\alpha(i) = a_\beta(i), \forall i$. Thus, the columns corresponding to l_α and l_β are identical.

□

Proof of Theorem 5:

Suppose the network is not i-identifiable. According to the definition of identifiability, $\exists \mathbf{X}_1 \neq \mathbf{X}_2, s.t. < \mathbf{X}_1 >=< \mathbf{X}_2 >= i, \mathbf{A}\mathbf{X}_1 = \mathbf{A}\mathbf{X}_2$.

Denote the location of the non-zero elements in $\mathbf{X}_1$ as $j_{11}, j_{12}, ..., j_{1i}$, and the location of the non-zero elements in $\mathbf{X}_2$ as $j_{21}, j_{22}, ..., j_{2i}$. Without loss of generality, assume $j_{11} \notin \{j_{21}, j_{22}, ..., j_{2i}\}$.

Then the corresponding links satisfy $\forall \phi \in \Phi, \forall j_1 \in \{j_{11}, j_{12}, ..., j_{1i}\}, l_{j_1} \in \phi \Longrightarrow \exists j_2 \in \{j_{21}, j_{22}, ..., j_{2i}\}, s.t. l_{j_2} \in \phi$.

Thus, $\forall \phi \in \Phi, l_{j_{11}} \in \phi \Longrightarrow \exists j_2 \in \{j_{21}, j_{22}, ..., j_{2i}\}, s.t. l_{j_2} \in \phi$.

This is $\nexists \phi \in \Phi, s.t. l_{j_{11}} \in \phi, l_{j_{21}}, ..., l_{j_{2i}} \notin \phi$.

So a network is i-identifiable if $\forall l_{j_1}, l_{j_2}, ..., l_{j_{i+1}} \in L, \exists \phi \in \Phi, s.t. l_{j_1} \in \phi, l_{j_2}, ..., l_{j_{i+1}} \notin \phi$.

□

Proof of Theorem 6:

Suppose $\exists l_{j_1}, l_{j_2}, ..., l_{j_{i+1}} \in L, s.t. \forall \phi \in \Phi, \forall j \in \{j_2, ..., j_{i+1}\}, l_j \in \phi \Longrightarrow l_{j_1} \in \phi$.

Thus, $\forall \phi \in \Phi, \forall \alpha \in \{j_1, ..., j_i\}, l_\alpha \in \phi \Longrightarrow \exists \beta \in \{j_1, j_3, j_4, ..., j_{i+1}\}, s.t. l_\beta \in \phi$,and $\forall \phi \in \Phi, \forall \beta \in \{j_1, j_3, j_4, ..., j_{i+1}\}, l_\beta \in \phi \Longrightarrow \exists \alpha \in \{j_1, ..., j_i\}, s.t. l_\alpha \in \phi$

Let $\mathbf{X}_1(j) = \begin{cases} 1 & if\ j \in \{j_1, ..., j_i\} \\ 0 & else \end{cases}$, and $\mathbf{X}_2(j) = \begin{cases} 1 & if\ j \in \{j_1, j_3, j_4, ..., j_{i+1}\} \\ 0 & else \end{cases}$.
Then $< \mathbf{X}_1 >=< \mathbf{X}_2 >= i, \mathbf{X}_1 \neq \mathbf{X}_2, \mathbf{A}\mathbf{X}_1 = \mathbf{A}\mathbf{X}_2$. The network is not i-identifiable.

So a network is i-identifiable only if $\forall l_{j_1}, l_{j_2}, ..., l_{j_{i+1}} \in L, \exists \phi \in \Phi, s.t. l_{j_1} \notin \phi, \exists l \in \{l_{j_2}, ..., l_{j_{i+1}}\} \in \phi$.

□

References

1. R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu, "Network tomography: recent developments," *Statist. Sci.*, vol. 19, no. 3, pp. 499–517, 2004.
2. Y. Xia and D. Tse, "Inference of link delay in communication networks," *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, vol. 24, pp. 2235–2248, DEC 2006.
3. T. Bu, N. Duffield, F. L. Presti, and D. Towsley, "Network tomography on general topologies," *SIGMETRICS Perform. Eval. Rev.*, vol. 30, no. 1, pp. 21–30, 2002.
4. Y. Tsang, M. Coates, and R. Nowak, "PASSIVE NETWORK TOMOGRAPHY USING EM ALGORITHMS," *IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS SPEECH AND SIGNAL PROCESSING*, vol. 6, no. 4031, 2001.
5. R. Narasimha, S. Dihidar, C. Ji, and S. W. McLaughlin, "Scalable diagnosis in ip networks using path-based measurement and inference: A learning framework," *Journal of Visual Communication and Image Representation*, vol. 21, no. 2, pp. 175 – 191, 2010.
6. F. Thouin, M. Coates, and M. Rabbat, "Real-time multi-path tracking of probabilistic available bandwidth," *CoRR*, vol. abs/1010.1524, 2010.
7. N. Duffield, "Network tomography of binary network performance characteristics," *IEEE Trans. Inform. Theory*, vol. 52, no. 12, pp. 5373–5388, 2006.
8. Y. Bejerano and R. Rastogi, "Robust monitoring of link delays and faults in ip networks," *Networking, IEEE/ACM Transactions on*, vol. 14, no. 5, pp. 1092 –1103, 2006.
9. H. X. Nguyen and P. Thiran, "Active measurement for multiple link failures diagnosis in ip networks," *LECTURE NOTES IN COMPUTER SCIENCE*, pp. 185–194, 2004.
10. H. X. Nguyen and P. Thiran, "The boolean solution to the congested ip link location problem: Theory and practice," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pp. 2117 –2125, may 2007.
11. A. Dhamdhere, R. Teixeira, C. Dovrolis, and C. Diot, "Netdiagnoser: troubleshooting network unreachabilities using end-to-end probes and routing data," in *Proceedings of the 2007 ACM CoNEXT conference*, (New York, NY, USA), pp. 18:1–18:12, ACM, 2007.
12. T. C. W. Site, "http://www.caida.org/tools/,"
13. V. N. Padmanabhan, L. Qiu, and H. J. Wang, "Passive network tomography using bayesian inference," *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pp. 93–94, 2002.
14. V. N. Padmanabhan, L. Qiu, and H. J. Wang, "Server-based inference of internet link lossiness," *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'03)*, vol. 1, pp. 145–155, 2003.
15. Y. Vardi, "Network Tomography: Estimating Source-Destination Traffic Intensities from Link Data," *Journal of the American Statistical Association*, vol. 91, no. 433, 1996.
16. L. Denby, J. M. Landwehr, C. L. Mallows, J. Meloche, J. Tuck, B. Xi, G. Michailidis, and V. N. Nair, "Statistical Aspects of the Analysis of Data Networks," *TECHNOMETRICS*, vol. 49, no. 3, pp. 318–334, 2007.
17. V. V. Vazirani, *Approximation algorithms*. Berlin: Springer-Verlag, 2001.
18. D. S. Hochbaum, *Approximation Algorithms for NP-hard problems*. Boston: PWS Publishing Company, 1997.
19. D.-Z. Du and F. K. Hwang, *COMBINATORIAL GROUP TESTING AND ITS APPLICATIONS*. Singapore: World Scientific Publishing Company, 2000.
20. H. Gao, F. K. Hwang, M. T. Thai, W. Wu, and T. Znati, "Construction of d(h)-disjunct matrix for group testing in hypergraphs," *Journal of Combinatorial Optimization*, vol. 12, no. 3, pp. 297–301, 2006.

Author Biographies

Sumit Roy received the B. Tech. degree from the Indian Institute of Technology (Kanpur) in 1983, and the M.S. and Ph.D. degrees from the University of California (Santa Barbara), all in Electrical Engineering in 1985 and 1988 respectively, as

well as an M. A. in Statistics and Applied Probability in 1988. Presently he is a professor of Electrical Engineering, Univ. of Washington where his research interests include analysis/design of wireless communication and sensor network systems with a current emphasis on wireless LANs (802.11) and wireless MANs (802.16), multi-standard wireless inter-networking and cognitive radio platforms, and sensor networking involving RFID technology.

Linda Yunlu Bai received the B.S degrees in Electronic Engineering from Tsinghua University, Beijing, China, in 2008. She is at present working towards her Ph.D. degree at the Fundamentals of Networking Laboratory, Department of Electrical Engineering, University of Washington, Seattle. Her current research interests focus on analysis of cognitive radio networks, cross-layer sensing algorithm design and application of compressive sensing in dynamic spectrum access.