An Efficient Solution to Locate Sparsely Congested Links by Network Tomography

Jinbiao Chen, Xiao Qi, Yongcai Wang, *Member, IEEE*
Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China
{cscjb06,qixiao0113}@gmail.com, {wangyc}@tsinghua.edu.cn

***Abstract*—Locating individual congested links in large scale networks is an important but difficult problem, because of the hardness to directly measure the massive links. Current advantages of network tomography propose to infer the link congestion states by end-to-end measurements via solving a set of linear equations in Boolean algebra. But one challenging problem in such approaches is the requirement to construct n linearly independent measurements for uniquely identifying the states of n links. It is especially cost inefficient when the congested links are sparse, but requiring larger than n measurements to form a full-rank observation matrix. In this paper, we focus on efficient methods to take only limited number of path measurements to locate the sparsely congested links. To avoid the ambiguity of solving the boolean equations, at first, we propose a compressive sensing method to estimate the congestion probabilities of the individual links based on the deficient measurements (routing matrix is not full rank). Based on the congestion probability estimation, a greedy iterative estimation algorithm is developed to locate the congested links by online snapshot of the deficient measurements. Extensive simulations shows the effectiveness of proposed methods which reduce the measurement costs while preserving the detection accuracy.**

I. INTRODUCTION

Accurate and timely knowledge of the internal link states of large networks (e.g. delays or congestion state of individual links of a intranet) is essential for many network applications, such as for route optimization, network performance evaluation, and anomalous/malicious behavior detection etc [3, 6]. But measuring the internal link states directly is generally infeasible, because of the high overhead of monitoring the massive links and transmitting the measurement packets. To deal with these difficulties, network tomography, which infers the internal states of individual links by end-to-end path measurements has attracted great research attentions. Since most of the link metrics, such as the link delay are *additive*, mathematically, an end-to-end measurement contributes a summation of metrics for the links along the path. The multiplicative metric (e.g., packet delivery ratio) can be expressed in an additive form by using the $log(\cdot)$ function. So that, taking multiple path measurements will constitute a linear observation model, which can be solved to infer the internal link states. It doesn't need to directly measure internal links and doesn't need the internal nodes to cooperate.

A special case of network tomography is to infer the internal congestion states by end-to-end congestion measurement. In this case, only the binary link states are considered, i.e, "1" represents *congested* and "0" presents *non-congested*. The path measurements are also binary: "1" means there is at least one link along the path are congested and "0" requires all the links along the path are non-congested. These binary features of links and paths request the network tomography model to be calculated and resolved in boolean algebra. In boolean algebra, it is difficult to find efficient method to resolve the ambiguities among the candidate solutions. A measured congested path of length l has $2^l - 1$ alternative internal link states. In order to uniquely determine the congestion states of all n individual links in a network, it needs the observation matrix contain n linear independent path measurements, which generally requires to take much larger than n measurements because some measurements are linearly dependent without providing new information.

But the congested links in a network are generally sparse, whose number is much less than the non-congested links. It is very inefficient to take $\gg n$ measurements for locating only sparsely congested links. So that, seeking for efficient methods to locate sparsely congested links has attracted increasing research attentions. In [12], the authors proposed a method to construct full rank observation matrix by virtually connecting some available observation paths instead of taking additional measurements. They showed that in Boolean Algebra, virtually connected paths could contribute independent observation vector. In [10], the authors showed that when n linearly independent measurement paths exist, there must exist a set of three pairwise independent spanning trees. Based on this, they developed efficient method to construct n linearly independent measurement. However, these methods still need to construct the rank-n observation matrix.

In this paper, we focus on the problem of locating the sparsely congested links by taking even less measurements, i.e., by using deficient observation matrix which is not full rank. Even further, we hope the number of measurement paths is less than the number of links. The contributions of our work include:

1) Instead of directly solving the model in boolean algebra, we propose a compressive sensing (CS) based method to firstly estimate the congestion probabilities of the individual links based on the historical deficient path measurements. This step votes the congestion probabilities of links.
2) Based on the estimated congestion probabilities of the links, we present a greedy iterative algorithm to locate the sparsely congested links when a set of paths are

978-1-4799-2003-7/14/$31.00 ©2014 IEEE

online measured. The algorithm evolves a unique solution that has the maximum likelihood of occurrence to resolve the ambiguity problem.

These proposed methods are extensively evaluated and compared with the methods of constructing full rank observation matrix. The simulation results show good accuracy of locating the congested links by using much less number of path measurements.

The rest of the paper is organized as follows. Problem model in boolean algebra is introduced in Section II. CS-based congestion probability estimation is presented in Section III. Greedy iterative algorithm for locating congestion links is presented in Section IV. Section V shows the evaluation results, and the paper is concluded in Section VI.

II. Problem Model

A. Network model

Let's consider a network, which is modeled by an undirected graph $G=(V,E)$. The vertex set V denotes the nodes (end hosts, switches or routers), and the edge set E represents the links connecting them (edge and link are used exchangablly in this paper). Let $n=|E|$ be the number of edges. Without loss of generality, let's assume G is connected, because different connected components have to be monitored separately. We assume links' states are independent and consider only $k \ll n$ links are congested. A congested link has packet loss rate higher than a threshold α. All the other $n-k$ links are non-congested with packet loss rate near zero. Let vector $\mathbf{x}=\{x_i\}$ represent the states of links, so that $\mathbf{x}$ is a $k-$sparse vector, containing k "1" and $n-k$ "0".

Let $\mathcal{P}$ be a set of the end-to-end measurement paths, where $|\mathcal{P}|=n_p$ is the number of measurements. We assume there is no loops in each path, and the length of a path $l_i < n$. The states of paths are denoted by $\mathbf{y}=\{y_i\}$. A path is congested if it passes through at least one congested link and called non-congested if all links in it are non-congested. Based on $G=(V,E)$ and the measurement paths $\mathcal{P}$, we can construct a binary routing matrix $R_{n_p\times n}$, whose entry $R_{ij}=1$ if the ith path routes through the jth edge and $R_{ij}=0$ otherwise. Each row of R thus indicates an end-to-end measurement path P_i ($P_i\in\mathcal{P}$) and each column represents a link of the network. Without loss of generality, we assume each link is passed by at least one path, otherwise its state cannot be inferred from measurement since link states are independent.

B. Problem Formulation

We can then establish a system of linear equations in Boolean algebra relating y_i and x_j like

$$y_i = \bigvee_{j=1}^{n} x_j \cdot R_{ij}, \tag{1}$$

where "$\vee$" is the Boolean "OR" operator, and "$\cdot$" denotes the usual multiplication operation. The vector form of (1) is

$$\mathbf{y} = \bigvee_{j=1}^{n} x_j \mathbf{r}_j, \tag{2}$$

where $\mathbf{r}_j$ is the jth column of R. We use a bold letter to represent a vector in this paper. The method to determine a path is congested or non-congested in the measurement process will be presented in Section III.

Traditionally, only when R contains n linearly independent measurements in Boolean Algebra, then a unique solution of $\mathbf{x}$ can be determined [11]. However, this generally requires the number of path measurements n_p to be larger than n. If the cost of one measurement is C, the total measurement costs are larger than nC. Our objective is to solve (2) even if $n_p < n$, which we call *deficient measurements*. This is generally a very difficult problem, because there will be ambiguous solutions when the observation matrix R is not full rank. To address this difficulty, we consider following two sub-problems:

1) Can we soften the objective to firstly estimate the link congestion probabilities by the historical deficient measurements if these probabilities remain unchanged during the measurement time?
2) Can we locate the congested links accurately by utilizing the prior congestion probabilities and the real-time deficient measurements?

We answered these two problems in the following sections.

III. Estimate the Link Congestion Probabilities via Compressive Sensing

Given the network topology and suppose $n_p < n$, we formulate the problem that estimating link congestion probabilities by a linear additive model, i.e., a system of linear equations with deficient coefficient matrix. Then we adopt CS based method to recover the these probabilities from this model.

For stating convenience, we first list several useful notations which will be used through this paper:

- $\mathbf{p}=[p_1\ p_2\ \dots\ p_n]^T$: The link congestion probability vector. p_j indicates link e_j's congestion probability and T denotes transposition. This vector is the objective of this section.
- $\mathbf{X}=[X_1\ X_2\ \dots\ X_n]^T$: A random binary variable vector representing the states of links. We assume each X_j is independent from each other.
- $\mathbf{Y}=[Y_1\ Y_2\ \dots\ Y_{n_p}]^T$: A random binary variable vector representing the states of paths.
- $\mathbb{P}_{\mathbf{p}}$: The probability measure on the set of network links when the link congestion probability vector is $\mathbf{p}$. For example, $\mathbb{P}_{\mathbf{p}}(X_j=1)=p_j$, for all $1\le j\le n$.

Additionally, there is one assumption about link congestion probability that is $0\le p_j<1$, for $1\le j\le n$, which implies the link e_j can not be disconnected, otherwise we can not uniquely identify the disconnected link only from end-to-end path measurements. Fig.1 is a simple example explaining why $p_j\neq 1$.

A. Problem Statement

We denote path P_i's congestion probability by ϕ_i, i.e., $\mathbb{P}_{\mathbf{p}}(Y_i=1)=\phi_i$. We can easily establish a relationship between

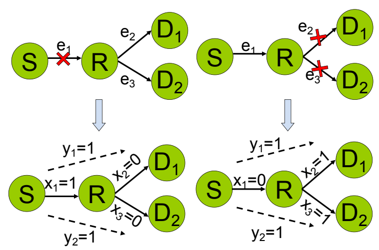


Fig. 1. The case that e_1 is disconnected or the case that e_2 and e_3 both are disconnected will generate the same measurement result: $y_1 = y_2 = 1$, which can not uniquely identify the $p_j, 1 \leq j \leq 3$

each path congestion probability and its links' congestion probabilities:

$$\begin{aligned}\phi_i = &\mathbb{P}_{\mathbf{p}}(Y_i = 1) = \mathbb{P}_{\mathbf{p}}(\vee_{j=1}^{n} X_j R_{ij} = 1)\\ &= 1 - \mathbb{P}_{\mathbf{p}}(\vee_{j=1}^{n} X_j R_{ij} = 0)\\ &= 1 - \prod_{j=1}^{n}(1-p_j)^{R_{ij}}.\end{aligned}$$

Then take logarithms for two sides of above equation, we can get a linear additive equation:

$$-\log(1-\phi_i) = \sum_{j=1}^{n}(-\log(1-p_j))R_{ij}, \tag{3}$$

for $1 \leq i \leq n_p$. Set $\widetilde{\phi}_i = -\log(1-\phi_i)$ and $\widetilde{p}_j = -\log(1-p_j)$, we can get a new form of (4)

$$\widetilde{\phi} = R\widetilde{\mathbf{p}}, \tag{4}$$

where $\widetilde{\phi} = [\widetilde{\phi}_1 \ \widetilde{\phi}_2 \ \ldots \ \widetilde{\phi}_{n_p}]^T$ and $\widetilde{\mathbf{p}} = [\widetilde{p}_1 \ \widetilde{p}_2 \ \ldots \ \widetilde{p}_n]^T$.

Solving (4) is our objective. However, the coefficient matrix or routing matrix R is usually rank deficient. So the conventional methods of solving system of linear equations can not work here. We have mentioned that the congested links only take a small portion of all links, so almost all link congestion probabilities are 0 or close 0 except a few spike values corresponding to the congested links. Mathematically, the vector $\mathbf{p}$ can be approximately treated as a k-sparse vector and so do $\widetilde{\mathbf{p}}$, i.e., $\|\mathbf{p}\|_0 = \|\widetilde{\mathbf{p}}\|_0 = k$. This fact implies that we can reconstruct the vector $\widetilde{\mathbf{p}}$ by solving the under-determined linear model using some CS algorithms, such as l_1 minimization and OMP (Orthogonal Matching Pursuit)[5, 13].

From the perspective of compressive sensing, only when R satisfies some special condition such as NSP (Null Space Property) or RIP (Restricted Isometry Property), the sparse vector $\widetilde{\mathbf{p}}$ can be reconstructed with high probability. So we first need to verify whether R satisfies the special condition. Although so far now R is unknown, we have obtained a property about R that each column of R contains at least one 1 based on the assumption in Section I that each link is passed by at least one path. Based on this property, we find that if the R multiply some constant, then it will satisfy the RIP condition

Definition 1 (Restricted Isometry Property or RIP [5]). *A matrix M satisfies the restricted isometry property of order k if there exists a $\delta_k \in (0,1)$ such that*

$$(1-\delta_k)||\mathbf{z}||_2^2 \leq ||M\mathbf{z}||_2^2 \leq (1+\delta_k)||\mathbf{z}||_2^2,$$

holds for all $||\mathbf{z}||_0 \leq k$.

We can interpret RIP as saying that the matrix M approximately preserves the length of any k-sparse vector $\mathbf{z}$. In the following part, we will construct a matrix satisfied RIP based on R. Denoted by $C = \{c_j, 1 \leq j \leq k\}$ the support set of $\mathbf{z}$, which records the indexes of non-zero elements of $\mathbf{z}$. In this paper, each column of R contains at least one 1-value element. So we can get two inequations as follows:

$$||R\mathbf{z}||_2^2 = \sum_{i=1}^{n_p}(\sum_{j=1}^{k} z_{c_j} R_{i,c_j})^2 \geq \sum_{j=1}^{k} z_{c_j}^2 = ||\mathbf{z}||_2^2;$$

$$||R\mathbf{z}||_2^2 \leq \sum_{i=1}^{n_p}(\sum_{j=1}^{k} z_{c_j})^2 \leq n_p k \sum_{j=1}^{k} z_{c_j}^2 = n_p k ||\mathbf{z}||_2^2.$$

Set $\delta_k = (n_p k - 1)/(n_p k + 1)$ and $M = \sqrt{2/(1+n_p k)}R$, it is easy to verify that the matrix M satisfies the RIP condition.

Although R need to multiply a constant to satisfy RIP, we still don't need to replace R by M, because if we did, the measurement vector also should multiply a same constant to keep the equations hold, which is meaningless. Therefore, it is enough to recover $\widetilde{\mathbf{p}}$ from model (4).

We have constructed the linear model for estimating link probability vector, then the left works are obtaining the measurement vector $\widetilde{\phi}$ and determining the routing matrix R, whose details will be introduced in the following part.

B. Obtain the Measurement Vector

1) How to Determine Congested Path: We first introduce how to determine whether one measurement path is congested or not. Similar with individual link, we set a path packet loss rate threshold β, whose value depends on concrete Internet application, for example we set $\beta = c\alpha$, where c is a constant. If path P_i's packet loss rate exceed β, it is considered as congested, i.e., $y_i = 1$ otherwise $y_i = 0$.

2) How to Obtain Path Congestion Probability: We introduce a new term of measurement snapshot, which contains n_p path measurements in a fixed length time slot. Take T snapshots $\mathcal{Y} = \{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \ldots, \mathbf{y}^{(\mathrm{T})}\}$, and where each snapshot contains the same n_p path measurements. According to above principle of determining congested path, we can get the concrete values of the T measurement vectors.

Consider Y_i is a random binary variable, then its expectation can be calculated like

$$\mathbb{E}_{\mathbf{p}}[Y_i] = \mathbb{P}_{\mathbf{p}}(Y_i = 1) = \phi_i,$$

where $\mathbb{E}_{\mathbf{p}}[Y_i]$ can be approximately computed by averaging all measured values of $y_i^{(t)}$ in $\mathcal{Y}$, i.e.,

$$\mathbb{E}_{\mathbf{p}}[Y_i] = \frac{1}{\mathrm{T}}\sum_{t=1}^{\mathrm{T}} y_i^{(t)},$$

thus each path congestion probability can be obtained.

C. Determine Routing Matrix

As the prior information for estimating congested links in next step, the link congestion probabilities not only can be recovered via CS, but also they must be identifiable. Therefor the routing matrix need to satisfy more conditions to meet this requirement. In this part, we will show a condition that can guarantee the recovered $\mathbf{p}$ is identifiable.

Usually a statistical model is said to be identifiable, if it is theoretically possible to learn the true value of this model's underlying parameters after obtaining an infinite number of observations from it. Mathematically, this is equivalent to saying that different values of the parameter must generate different probability distribution of observable variables. In our context, $\mathbb{P}_{\mathbf{p}}$ is the statistical model, $\mathbf{p}$ is model's underlying parameter and measurement snapshots are the observation data. Hence we say the link congestion probabilities are statistically identifiable if the following inference always holds:

$$\mathbb{P}_{\mathbf{p}}(\mathbf{Y}=\mathbf{y}) = \mathbb{P}_{\widetilde{\mathbf{p}}}(\mathbf{Y}=\mathbf{y}) \Rightarrow \mathbf{p}=\widetilde{\mathbf{p}},$$

for any snapshot $\mathbf{y}$.

For obtaining a identifiable link probability vector, we need a guidance to tell us how to design the measurement such that we can obtain $\mathbf{p}$ correctly from model (4). Fortunately, the previous works [12] gave a such guidance, which is a theorem stating the relationship between the identifiability of $\mathbf{p}$ and routing matrix R:

Theorem 1. *The link congestion probability vector* $\mathbf{p}$ *is identifiable if and only if all the columns of* R *are distinct.*

We omit the proof details which have been shown in [12]. This theorem tell us once we can construct a routing matrix from the n_p path measurements satisfying the condition that all columns are distinct, the probability vector can be identified from a certain number of snapshots. It is not difficult to select such n_p proper measurement paths.

So far now, the prepared works of CS have done, we employ the OMP [13] algorithm to recover $\widetilde{\mathbf{p}}$ from (4). To identify the sparse vector $\widetilde{\mathbf{p}}$, the algorithm determines in a greedy manner which columns of R contribute most to the measurement vector $\widetilde{\phi}$. It picks columns of R in a greedy fashion. At each iteration, it chooses the column of R that is most strongly correlated with the remaining part of $\widetilde{\phi}$. Then it subtracts off the contribution to $\widetilde{\phi}$ and iterates on the residual. The algorithm terminates when a k-sparse vector is determined, which needs only polynomial time computation. More details of OMP can be found in [13].

IV. Locate Congested Links by Using the Prior Probabilities

Based on the prior link congestion probability vector $\mathbf{p}$ and the most recent measurement snapshot $\mathbf{y}=[y_1\ y_2\ \dots\ y_{n_p}]^T$, we propose a greedy iteration algorithm to locate the most possible congested links, i.e., find the most likely solution of (2). We divided into two steps to complete this task.

A. Reduce the Problem Dimension

It is easy to know that when $y_i=0$, i.e., path P_i is a good path, we can infer that any link e_j covered by this path is good link, i.e.,$x_j=0$. So if $\|\mathbf{y}\|_0 < n_p$, then based on above fact, we can simplify this problem as follows.

Let R' be the matrix obtained from R by removing all rows associating with the good paths and deleting all columns corresponding to links in the good paths. Now each row of R' indicates a congested path and each column represents a link which belongs to at least one congested path. Accordingly, the measurement vector $\widetilde{\mathbf{y}}$ and original vector $\widetilde{\mathbf{x}}$ should be deleted the corresponding elements respectively. We denote by $\mathcal{P}_C$ and E_C the left congested path set and suspect congested link set respectively, then the dimension of R' will be $|\mathcal{P}_C|\times|E_C|$.

B. Find the Most Likely Solution

Denote by $E_S\subseteq E_C$ the set of congested links associated with the 1-value elements of the most likely solution, i.e.,if $\widetilde{\mathbf{z}}$ is the most likely solution of (2), then $E_S=\{e_j : x_j = 1, x_j\in\widetilde{\mathbf{z}}\}$. Our goal is to find this set E_S with known the prior probability vector. Firstly, E_S need to be satisfied the condition that all congested paths must be covered by at least one link belonging to it. In other words, each congested path must contain at least one element in E_S. There may be many solutions satisfy this condition, and we need to find a most likely solution that the probability of appearing this solution is maximum based on the vector $\mathbf{p}$. We can formulate it like the following:

$$\arg\max_{E_S\subseteq E_C}\mathbb{P}_{\mathbf{p}}(\mathbf{X}) = \arg\max_{E_S\subseteq E_C}\prod_{j=1}^{|E_C|} p_j^{x_j}(1-p_j)^{(1-x_j)}. \quad (5)$$

subject to

$$\sum_{j=1}^{|E_C|} R'_{ij}x_j \geq 1\ \ for\ all\ 1\leq i\leq|\mathcal{P}_C|. \quad (6)$$

Taking the logarithm of (5), we obtain the optimization problem

$$\begin{aligned}&\arg\max_{E_S\subseteq E_C}\mathbb{P}_{\mathbf{p}}(\mathbf{X})\\ =&\arg\max_{E_S\subseteq E_R}\sum_{j=1}^{|E_C|}\left(x_j\log\frac{p_j}{1-p_j}+\log(1-p_j)\right)\\ =&\arg\max_{E_S\subseteq E_C}\sum_{j=1}^{|E_C|}\gamma_j\end{aligned} \quad (7)$$

subject to the constraint (6), where

$$\gamma_j = x_j\log\frac{1-p_j}{p_j}+\log\frac{1}{1-p_j}.$$

It is easy to know that

$$\gamma_j=\begin{cases}\log\frac{1}{p_j}, & \text{if } x_j=1\\ \log\frac{1}{1-p_j}, & \text{if } x_j=0\end{cases}.$$

We set $\Gamma_1 = \{\gamma_j^{(1)} = \log \frac{1}{p_j} : 1 \leq j \leq |E_C|\}$ and $\Gamma_0 = \{\gamma_j^{(0)} = \log \frac{1}{1-p_j} : 1 \leq j \leq |E_C|\}$. This optimization problem can be described as follows: Select $|E_C|$ elements from Γ_0 and Γ_1 to make their sum value maximum, note that if select $\gamma_j^{(1)}$, can not select $\gamma_j^{(0)}$ and vice verse. Meanwhile the set E_S derived from above selection must satisfy the restriction condition of (5), i.e., $\sum_{j=1}^{|E_C|} R'_{ij} x_j \geq 1$ for all $1 \leq i \leq |\mathcal{P}_C|$.

We propose a greedy iteration algorithm to locate the congested link. In each iteration the algorithm selects the maximum value (suppose $\gamma_k^{(1)}$) from Γ_1, which is equivalent to select one suspected congested link e_k from E_C and put it to E_S, meanwhile delete $\gamma_k^{(1)}$ and the paths associated with e_k (denoted by $D(e_k)$) from Γ_1 and $\mathcal{P}_C$ respectively. The algorithm is done until $\mathcal{P}_C$ becomes empty. The details of this algorithm are shown like follows:

Algorithm 1 The Locating Congested Links Algorithm

Input:
The suspected congested link set E_C;
The set of congested paths $\mathcal{P}_C$.
Output:
The set of the most possible congested links E_S;
The most likely solution of (2) $\mathbf{x}$.

Initialize E_S to an empty set, and $\mathcal{Q}_B := \mathcal{P}_C$, $\mathbf{x} := 0$
while $\mathcal{Q}_B \neq \emptyset$ **do**
  Find a link $e_k \in E_C$ that maximizes $\gamma_k^{(1)} = \log \frac{1}{p_k}$.
  Add e_k to the solution E_S: $E_S := E_S \cup \{e_k\}$, set $x_k := 1$.
  $\Gamma_1 := \Gamma_1 \setminus \gamma_k^{(1)}$, $E_C := E_C \setminus e_k$ and $\mathcal{Q}_B := \mathcal{Q}_B \setminus D(e_k)$
  $D(e_j) := D(e_j) \setminus D(e_k)$ for all $e_j \in E_C$.
end while

V. SIMULATION RESULTS

We use two metrics to evaluate the performance of the locating congested links algorithm: the Detection Rate (DR) and the False Positive Rate (FPR). The former is the percentage of links that are correctly detected as congested, and the latter is the percentage of links that are good but diagnosed as congested. Denote by E_T the set of the actual congested links, and E_S the set of links identified as congested by our approach, the two rates are calculated like:

$$DR = \frac{|E_T \cap E_S|}{|E_T|}; \;\; FPR = \frac{|E_T \setminus E_S|}{|E_S|}$$

A. The Preliminary Knowledge of Simulation

We use NS-3 as the simulation platform, which is a powerful discrete-event network simulator. The NS-3 simulation core supports research on both IP and non-IP based networks, and supports interacting with real systems. In our simulation, we set the network containing 100 nodes which construct some mesh topology. Each link congestion probability p_j is uniformly distributed between 0 and 1. We set the loss rates of congested links uniformly distributed in $[0.05, 1)$ and good links in $[0, 0.01]$. Once each link has been assigned a loss rate, the actual losses on each link follow a Bernoulli process, where the packets are dropped on each link with fixed probability. The pass loss rate is calculated based on the transmissions of 1000 packets. We set the link loss rate threshold t_l is 0.01 whereas the path threshold $t_p = 1 - 0.99^d$ where d is the length of the path.

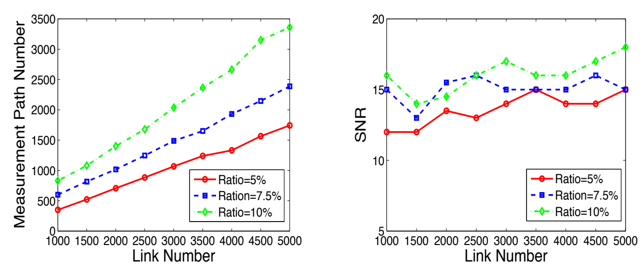
In this simulation, we first take 40 measurement snapshots to learn the prior link congestion probability vector $\mathbf{p}$. Denote by $\widehat{\mathbf{p}}$ the probability vector recovered from our algorithm in Section III. Once obtaining $\widehat{\mathbf{p}}$, we can infer the congested links in subsequent measurements using Algorithm 1.

B. Simulation Results

We first evaluate the accuracy of recovered probability vector $\widehat{\mathbf{p}}$ via calculating the SNR value between $\mathbf{p}$ and $\widehat{\mathbf{p}}$ following the formula

$$SNR_{dB} = 10 \log_{10}^{\frac{||\mathbf{p}||_2^2}{||\mathbf{p}-\widehat{\mathbf{p}}||_2^2}}$$

Fig.2 was drawn for measurement number and SNR values among 9 different link numbers and 3 different congestion ratios ($\frac{k}{n_e}$), where k is the number of congested links. We can find that as the link number and congestion ratio increasing, the measurement number in one snapshot also increases to keep the recovering accuracy of $\widehat{\mathbf{p}}$. Although the SNR values indicate that the accuracy are not perfect, all of them are round 15dBm, which is also a acceptable results in compressive sensing.

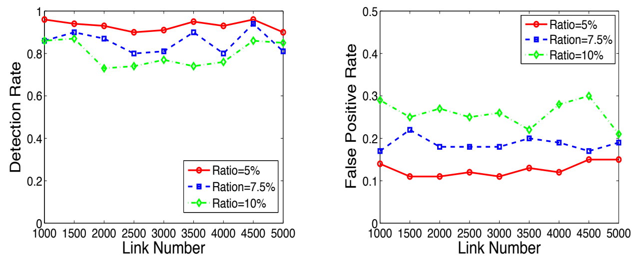


(a) Measurement Number for Different Link Number and Ratio

(b) SNR between $\mathbf{p}$ and $\widehat{\mathbf{p}}$

Fig. 2. Measurement Number and SNR between $\mathbf{p}$ and $\widehat{\mathbf{p}}$

Based on the recovered probability vector $\widehat{\mathbf{p}}$, we infer the congested links via Algorithm 1. We compared the detection and false positive rate regarding different link number and congestion ratio. From Fig 3, we find that although the accuracy of $\widehat{\mathbf{p}}$ is not perfect, we still can detect the congested links with high probability. For example, the detection rate for congested links (Fig.3(a)), we can easily find that almost all the congested links can be checked out and the lower congestion ratio the better performance. Although the false positive rate (Fig.3(b)) for congested links are not very well, we can consider these results as pessimistic estimation, in other words, if some link is ranked as a good link via our algorithm, then this link in general can be considered as a good link.



(a) Detection Rate for Congested links (b) False Positive Rate for Congested Links

Fig. 3. Detection and False Positive Rate

VI. RELATED WORK

Network tomography is an elegant scheme for troubleshooting network, which observes the characteristics of different end-to-end paths and combines the observations to infer the properties of individual links. In 1996, Vardi et al [14] first coined the term network tomography which encompasses these classes of approaches seeking to infer internal link parameters and identify link congestion status. In [3], the authors summarized the early researches in network tomography, and divided the relevant problems into two categories: (1) link-level properties estimation based on end-to-end path-level traffic measurements and (2) sender-receiver path-level traffic intensity estimation based on link-level traffic measurements. Locating congested links just belongs to the first category of problems.

In network tomography, the estimated link-level properties contain link traffic, loss rate, delay, state and so on [7–10, 12]. Based on the model of link properties, most existing tomography approaches can be broadly classified as the algebraic (Linear and Boolean) and the statistical. In the algebraic approaches, the link properties are considered as unknown constants, and computed from the end-to-end path measurements via some techniques of linear or Boolean algebra [4, 10, 11]. Statistical approaches model link properties as random variables with unknown probability distributions, and apply various parametric/nonparametric techniques to infer the distributions from path measurements [8, 9].

Except the above two categories of approach, in recent years, more and more researchers have begun to apply compressive sensing for solving network tomography problems [1, 2, 6], and so do we. Particularly in a practical large-scale network, abnormal links usually take a very small part of all links, adopting compressive sensing can detect these abnormal links with high probability from a relative few end-to-end measurements.

VII. CONCLUSION AND FUTURE WORKS

In this paper, we address the problem of congested links localization, which is one core problem in network tomography. We develop a learning-estimating scheme to infer the congested links in the network. In this scheme, we first construct a linear model regarding path congestion probability and link congestion probabilities, and learn these link congestion probabilities based on a certain number of snapshots via compressive sensing. Then using the prior link congestion probabilities, we develop a greedy iteration algorithm to infer the most possible solution of the linear model. We simulate our method based on NS-3 platform and the simulation results show that almost all congested links can be checked by our approach.

In the future, there are two works we plan to do:(i) design a generic principle for selecting the measurement path and bound the number of the path based on Theorem 1, (ii) apply our approach in a real network to check its validity.

REFERENCES

[1] Mayank Bakshi, Sidharth Jaggi, Sheng Cai, and Minghua Chen. Sho-fa: Robust compressive sensing with order-optimal complexity, measurements, and bits. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pages 786–793. IEEE, 2012.

[2] Sheng Cai, Mayank Bakshi, Sidharth Jaggi, and Minghua Chen. Frantic: A fast reference-based algorithm for network tomography via compressive sensing. Technical report, Technical Report, 2013, available at ht tp://personal. ie. cuhk. edu. hk/˜ cs010/files/ISIT13. pdf.

[3] Rui Castro, Mark Coates, Gang Liang, Robert Nowak, and Bin Yu. Network tomography: recent developments. *Statistical science*, pages 499–517, 2004.

[4] Yan Chen, David Bindel, Hanhee Song, and Randy H Katz. An algebraic approach to practical and scalable overlay network monitoring. In *ACM SIGCOMM Computer Communication Review*, volume 34, pages 55–66. ACM, 2004.

[5] David L Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306, 2006.

[6] Mohammad Hamed Firooz and Sumit Roy. Network tomography via compressed sensing. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–5. IEEE, 2010.

[7] Denisa Ghita, Can Karakus, Katerina Argyraki, and Patrick Thiran. Shifting network tomography toward a practical goal. In *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*, page 24. ACM, 2011.

[8] Denisa Ghita, Hung Nguyen, Maciej Kurant, Katerina Argyraki, and Patrick Thiran. Netscope: practical network loss tomography. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[9] Gang Liang and Bin Yu. Maximum pseudo likelihood estimation in network tomography. *Signal Processing, IEEE Transactions on*, 51(8):2043–2053, 2003.

[10] Liang Ma, Ting He, Kin K Leung, Don Towsley, and Ananthram Swami. Efficient identification of additive link metrics via network tomography. ICDCS, 2013.

[11] Hung X Nguyen and Patrick Thiran. Active measurement for multiple link failures diagnosis in ip networks. In *Passive and Active Network Measurement*, pages 185–194. Springer, 2004.

[12] Hung Xuan Nguyen and Patrick 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*, pages 2117–2125. IEEE, 2007.

[13] Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 53(12):4655–4666, 2007.

[14] Yehuda Vardi. Network tomography: Estimating source-destination traffic intensities from link data. *Journal of the American Statistical Association*, 91(433):365–377, 1996.