Computer Communications 127 (2018) 122–130

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Compressed sensing based loss tomography using weighted ℓ_1 minimization

XiaoBo Fan*, Xingming Li, JianKang Zhang

School of Communication and Information Engineering, University of Electronic Science and Technology of China, China

ARTICLE INFO

Keywords:
Network tomography
Weighted ℓ_1 minimization
Congestion localization
Compressed sensing

ABSTRACT

Network tomography allows the measurements of end-to-end to infer network internal links characteristics such as packet loss rates and delay. In this paper, we focus on the problem of estimating links loss rates, especially locating the congested links in network. Applying concepts of compressed sensing and Maximum A-Posteriori (MAP) estimation, we propose a new loss tomography scheme. Contrary to existing works that use ℓ_1 minimization, the proposed scheme adopts *weighted* ℓ_1 minimization as the implementation of compressed sensing, whose weights can be set wisely in order to improve tomography result. We exploit the temporal correlations of link losses and determine weights using the links prior congestion probabilities. The probabilities can be uniquely identified from multiple measurements by solving boolean algebra equations. We conduct a simulation performance analysis of loss tomography, demonstrating that higher estimation accuracy can be obtained through the proposed scheme.

1. Introduction

Network management tasks such as fault and congestion detection or traffic management often require performance parameter estimation of internal links. *Network tomography* [1–13] allows the measurement of end-to-end to infer network internal links performance characteristics such as link packet loss rates and delay. Roughly speaking, packets are sent from source nodes and processed at receivers to get path performance measurements, then link characteristics are obtained by exploiting the dependence between links and corresponding paths. There are two schemes to collect measurements on measurement paths in network tomography: *passive measurement* and *active measurement*, where the former collects end-to-end measurements by exploiting existing packets in network and the latter collects them by injecting probe packets. Active measurement always can get a more accurate result while needs additional traffic, which increases network burden. However, the additional traffic is small if the measurement paths are well designed. In this paper, we use active measurement scheme in which we also call measurement path *probe path*.

We focus on the problem of estimating links loss rates from paths loss rates, which is also known as *loss tomography*, especially the identification of congested links in network. We consider a link is congested if the number of dropped packets has exceeded a certain percentage of all packets, i.e. the link loss rate is much large. Determining link loss rates is not trivial since the end-to-end path measurements do not provide enough information. In fact, the inference of loss tomography can be represented as a linear model [8], where link loss rates are represented by variables to be solved, that is, we can formulate loss tomography as a linear inverse problem. However, in practice network monitor system, the probe paths often unable to determine loss rates uniquely, which makes the inverse problem *ill-posed*.

Many schemes have been proposed to handle the ill-posed problem in loss tomography [4–6,8–11]. Generally speaking, these methods make different assumptions and bring in additional information. In [4,6], loss rate is estimated based on multicast transmission assumption by exploiting temporal correlation between packets. As multicast is not widely deployed in actual network, unicast is used to imitate the packets behavior of multicast in [5]. Some methods [8–10] do not utilize the correlation of packets but take advantage of correlation between paths, which make it much easier for them to implement. There are also some works combine loss tomography with network coding theory to reveal correlation between probe paths [11]. In this paper, different from the methods above, we adopt the *compressed sensing* theory and apply the *weighted ℓ_1 minimization* method to loss tomography.

The concept of "*compressed sensing*" [14], which is an emerging theory in signal/image processing, has been proposed for network tomography recently. Compressed sensing can solve the ill-posed linear inverse problem with a prior information that the solution is *sparse*. The terminology sparse towards a vector means that only a few non-zero values exist in the vector. One advantage of utilizing compressed sensing to tomography problem is that it only needs a few probe paths. According to the compressed sensing theory, if a vector is k-sparse, then we can precisely recover it with only $O(k\log(n/k))$ measurements

* Corresponding author.
E-mail addresses: xiaobof@outlook.com (X. Fan), xingmingl@uestc.edu.cn (X. Li), zjk0501@163.com (J. Zhang).

https://doi.org/10.1016/j.comcom.2018.06.004
Received 26 July 2016; Received in revised form 18 May 2017; Accepted 11 June 2018
Available online 15 June 2018

[15,16]. That means when we apply compressed sensing to loss tomography, it can significantly reduce traffic burden induced by probe paths.

Many network tomography problems can benefit from compressed sensing due to their naturally sparse characteristics, that is, the links with high loss rates or large delay are always sparse. For loss tomography, it means that the congested links with high link loss rates are only a mall fraction of all links. Several works have been done related to this domain. Xu et al. [12] illuminates compressed sensing under graph constraint. Firooz and Roy [13] prove the condition measurement matrix should be satisfied to identify the k largest value of link delay. T. Matsuda et al. [17] roughly classifies network links according to their packets loss rate using $\ell_1 - \ell_2$ optimization, and recently, a tomography method using sparse Bayesian learning has been proposed in [18]. These works generally try to answer the two questions:

- How many probe paths should be established and how to establish probe paths between measurement nodes under graph constraint.
- How to implement compressed sensing so as to get link-level performance parameters.

Our work here focuses on the latter problem. Most works in the literature adopt ℓ_1 minimization to estimate link characteristics, and a sparse approximation of link loss rates is obtained by solving ℓ_1 minimization. Therefore, it is reasonable to use a *prior* information to improve the accuracy of sparse approximation. The weighted ℓ_1 minimization has been used to enhance sparsity in compressed sensing [19]. The technology of using weights to attach prior information to improve the performance of ℓ_1 minimization have been studied in [20,21]. As far as we know, this is the first time applying it to network tomography.

In this paper, we propose a new loss tomography scheme using weighted ℓ_1 minimization. In the proposed scheme, we employ active measurements, where probe packets are transmitted on measurement paths in order to obtain paths loss rates. Then links loss rates are estimated by weighted ℓ_1 minimization, whose weights are determined by links recent "behavior"—frequent congested links with small weights. Finally, links are classified into *normal* or *congested* classes according to their loss rates. With the proposed scheme, we can efficiently implement loss tomography.

Our contributions in this paper are as follows:

- We propose a loss tomography scheme using weighted ℓ_1 minimization. The scheme enhances the power of traditional compressed sensing applied to network tomography, especially when the number of congested links is so large that beyond the ability of ℓ_1 or $\ell_1 - \ell_2$ minimization.
- We determine the weights using Maximum A-Posteriori (MAP) estimation based on the prior probability of congestion, and thus connect compressed sensing theory with Bayesian theory.

The remainder of this paper is organized as follows. In Section 2, we formulate the loss tomography and congested links location problem. In Section 3, we explain compressed sensing and use it on loss tomography. In Section 4, we describe the proposed scheme, including the determination of the weights and computation of prior probability. In Section 5, we evaluate the performance of the proposed scheme with simulation experiments. Finally, we conclude the paper with reflections on future work in Section 6.

2. Loss tomography model

As is customary, we model the network topology as an undirected graph $G = (V, E)$. The nodes V represents the hosts/routers of the network and E denotes the links. In order to get end-to-end measurements, some nodes in the network are chosen as *measurement nodes*, which usually located at the boundary of a network. Then probe paths are established between the measurement nodes and packets are sent on these paths to get end-to-end performance.

Given the network topology with n links and r probe paths, we can easily establish relationships between the link loss rates and the path loss rates. We employ two loss tomography models in this work: linear model and boolean model. The linear model establishes linear relationships between links and paths while boolean model keeps boolean relationships (the former is known as *analog tomography* while the latter represents *boolean tomography* [22]). In this paper, we use linear model to implement loss tomography and boolean model to calculate the prior probabilities.

2.1. Linear model

Because the overall transmission rate of a path is the product of the transmission rate of all links belonging to the path, take the logarithm and we have the linear model [8]:

$$-\log(\phi_i) = -\sum_{k=1}^{n} r_{ik}\log(\phi_{e_k}) \tag{1}$$

where ϕ_i is the transmission rate (i.e., one minus the loss rate) of path m_i and ϕ_{e_k} is the transmission rate of link e_k, the value of r_{ik} is 1 if measurement path m_i pass link e_k and 0 otherwise. If there are r end-to-end paths, then (1) can be rewrited as the matrix form,

$$\boldsymbol{y} = R\boldsymbol{x} \tag{2}$$

where $\boldsymbol{y} = (y_1, y_2, \cdots, y_r)^{\mathrm{T}}$ with $y_i = -\log(\phi_i)$, and $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^{\mathrm{T}}$ with $x_i = -\log(\phi_{e_i})$. The R is an $r \times n$ *routing matrix* that consists of r_{ik}. In this paper, we use a bold letter to represent a vector and T denote the transposition of a vector or matrix.

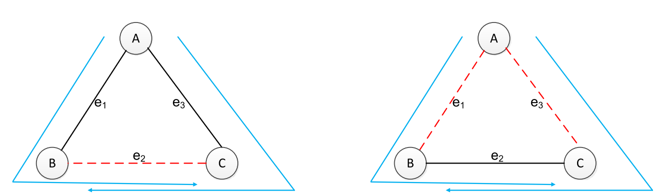
For the simple example in Fig. 1, if we have two measurement paths $\{e_1e_2, e_3e_2\}$, then the equivalent routing matrix is

$$R = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}. \tag{3}$$

Obviously, we can get $\boldsymbol{x} = R_L^{-1}\boldsymbol{y}$ from (2) if the rank $r(R) = n$, where R_L^{-1} is the left-inverse of R. However, in practical loss tomography, $r(R) < n$ is always desirable in order to reduce additional traffic burden, which means the equation has many candidates of the solution. In this paper, we utilize the sparsity of the congested links to deal with the problem. We call link e_k is *congested* if $\phi_{e_k} < \phi_{th}$, and call e_k is *normal* if $\phi_{e_k} \geq \phi_{th}$. We denote E_c the set of congested links and E_n the set of normal links. The threshold ϕ_{th} is specified by user and can be changed by the applications depending on their performance requirements.

2.2. Boolean model

The boolean model is adopted to calculate the prior probabilities. Let p_k denote the probability that link e_k is congested and $\boldsymbol{p} = (p_1, p_2, \cdots p_n)^{\mathrm{T}}$ the vector of link state probabilities. One possible way to obtain $\boldsymbol{p}$ has been studied in [10]. In the paper, they first show theoretically that it is possible to learn $\boldsymbol{p}$ from end-to-end measurements



(a) ℓ_1 minimization (b) weighted ℓ_1 minimization

Fig. 1. ℓ_1 minimization and weighted ℓ_1 minimization for 3-node ring network. The true congested links are e_1, e_3 and the dotted links are the results identified by the algorithms.

based on boolean model, and then develop an algorithm to estimate $\boldsymbol{p}$ from a small number of multiple measurements over a period of time (which they called snapshots). We will briefly describe the algorithm in Section 4.3.2 and readers can delve into details in the reference.

Unlike the linear model, the boolean model treats the state of links or paths binary—1 if congested and 0 if normal. We define a path to be congested when it contains at least one congested link and be normal when it go through only good links. Let boolean variable s_i and t_i represent the state of the path m_i and link e_i respectively: $s_i = 0$ if m_i is normal and $t_i = 0$ if e_i is normal, otherwise they equal to 1. Then an equation between s_i and t_i can be established as:

$$s_i = \bigvee_{k=1}^{n} r_{ik} \cdot t_k \qquad i = 1, 2, \cdots, r \tag{4}$$

where "$\bigvee$" denotes the binary max operation and "$\cdot$" denotes the usual multiplication operation. r_{ik} remains the same as linear model. Note that the Eq. (4) can be considered as the boolean version of linear model (2).

There is no paradox between boolean model and the linear model. We solve different problems based on different models. The boolean model is used to calculate prior because the random variable which represents link congested probability is boolean, while the linear model exists throughout this paper. Taking Fig. 1 for example, if we set the transmission rates of corresponding links $\phi_{e_1} = 0.92$, $\phi_{e_3} = 0.92$, $\phi_{e_2} = 0.995$, moreover, we assume $\phi_{th} = 0.97$. Then for linear model, we have $\boldsymbol{x} = (-\log 0.92, -\log 0.92, -\log 0.995)^{\mathrm{T}}$ and $\boldsymbol{y} = (-\log 0.92 - \log 0.92, -\log 0.92 - \log 0.995)^{\mathrm{T}}$. Yet for boolean model, $\boldsymbol{s} = (1, 1, 0)^{\mathrm{T}}$ and $\boldsymbol{t} = (1, 1)^{\mathrm{T}}$.

3. Compressed sensing and ℓ_1 minimization

Compressed sensing is a theory for obtaining unknown sparse solution in ill-posed linear inverse problem, with a prior information that the solution is sparse. Taking (2) for example, as only congested links have large entries in $\boldsymbol{x}$, and the number of congested links is sparse compare to all links (suppose no more than k, $k \ll n$), we are interested in solutions $\boldsymbol{x}$ with only a few large entries. If all the other entries are exactly zero, we call such vector exactly k-sparse. For loss tomography, the other entries are often small but non-zero ($x_k = 0$ only when $\phi_{e_k} = 1$), we refer to such vector as *nearly* k-sparse.

Perhaps the most famous and most commonly used method to recover a sparse vector is the ℓ_1 minimization, which is formulated as

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \qquad \text{subject to} \quad R\boldsymbol{x} = \boldsymbol{y} \tag{5}$$

where $\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |x_i|$ is the ℓ_1 norm of $\boldsymbol{x}$. The conditions imposed on R to guarantee that $\hat{\boldsymbol{x}}$ approximately equals true solution $\boldsymbol{x}$ have been studied in [13,23]. The ℓ_1 minimization tends to find the most sparse solution under linear constraints. For loss tomography, it usually works since there are only a few congested links in most cases. However, sometimes more than k congested links exist in network, especially when some links have higher congested *probabilities* than others.

In order to state the problem of using ℓ_1 minimization when there are more congested links than it can deal with, let us consider congestion detection problem in the simple topology of Fig. 1. The probe paths are $\{e_1e_2, e_3e_2\}$, and the constructed routing matrix (3) can detect any single congested link according to [13]. We set the loss rates of corresponding links $\bar{\phi}_{e_1} = 0.08$, $\bar{\phi}_{e_3} = 0.08$, $\bar{\phi}_{e_2} = 0.005$ (we denote $1 - a$ as $\bar{a}$ in general). That is, the link e_2 is normal while e_1 and e_3 are congested. We conduct the ℓ_1 minimization on this topology and get $\bar{\phi}_{e_1} = 0$, $\bar{\phi}_{e_3} = 0$, $\bar{\phi}_{e_2} = 0.0846$. It is surprising that the link e_2 is determined as congested and e_1, e_3 are determined as normal, thus completely conflict with the fact.

4. Loss tomography with weighted ℓ_1 minimization

4.1. Weighted ℓ_1 minimization

The above situation that ℓ_1 minimization failed inspires us to use weighted ℓ_1 to get a more robust tomography result. The weighted ℓ_1 minimization can treat links unequally by setting different weights, which is formulated as

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \sum_{k=1}^{n} \omega_i |x_i| \qquad \text{subject to} \quad R\boldsymbol{x} = \boldsymbol{y}, \tag{6}$$

where $\omega_1, \omega_2, \ldots, \omega_n$ are weights and we denote vector $\boldsymbol{\omega} = (\omega_1, \omega_2, \cdots, \omega_n)$. Note that the ℓ_1 minimization can be treated as a special case of (6) with $\omega_1 = \omega_2 = \cdots = 1$. Just like its "unweighted" counterpart ℓ_1 minimization, the convex problem (6) can be recast as a linear program and can be solved by any LP optimizer.

Because weighted ℓ_1 minimization is generalization of unweighted ℓ_1, we may get result of (6) at least as good as (5). Let us illustrate the weighted ℓ_1 minimization by coming back to our simple example in Fig. 1 . Consider now a hypothetical weighting vector $\boldsymbol{\omega} = (1, 1, 3)$. Using the same probe paths, we will get the result $\bar{\phi}_{e_1} = 0.0846$, $\bar{\phi}_{e_3} = 0.0846$, $\bar{\phi}_{e_2} = 0$ by solving the weighted ℓ_1 minimization. It indeed correctly locate the congested links (e_1, e_3) compare to the unweighted ℓ_1. Hence, we think of the weights $\boldsymbol{\omega}$ as free parameters in the (6), whose values—if set wisely—could improve the diagnostic result.

This raises the immediate question: what values for the weights will improve loss tomography? Generally speaking, the large entries in ω_i force the solution $\hat{\boldsymbol{x}}$ to concentrate on the indices where ω_i is small. If we know which links are actual congested, we can just set zeros for the corresponding ω_i and large values for the others, i.e.,

$$\omega_i = \begin{cases} 0 & e_i \text{ is congested,} \\ c & e_i \text{ is normal.} \end{cases} \tag{7}$$

where c is positive. In fact, by construction of (7), the weighted ℓ_1 minimization can precisely locate the congested links as long as probe paths cover all the links. It is of course impossible to construct the precise weights (7) without knowing the congested links themselves. However, it is possible to estimate the prior probabilities of links been congested according to [10], this inspires us that small weights could be used to the links with high congested probability (discourage big entries in the recovered $\hat{\boldsymbol{x}}$), while large weights could be used to the links with low congested probability (encourage small entries in the recovered $\hat{\boldsymbol{x}}$).

4.2. Recovery error via weighted ℓ_1 minimization

We defer presenting the method of setting weights to next subsection and instead provide a theoretical analysis on the quality of the solution to the weighted ℓ_1 minimization. Specifically, we study the recovery error if routing matrix R is viewed as the adjacency matrix of an expander graph—to be defined below.

Definition 1. A $(t, \epsilon) - $ *expander* is a bipartite simple graph $G = (A, B, E)$[1], with left degree d such that for any $\Theta \subset A$ with $|\Theta| \le t$, the following condition holds:

$$|N(\Theta)| \ge (1 - \epsilon) d |\Theta|$$

where $N(\Theta)$ is the set of neighbors of Θ.

The adjacency matrix of an expander graph can be denoted by $M = [m_{ij}]$, where $m_{ij} = 1$ if node $i \in A$ is connected to node $j \in B$, otherwise $m_{ij} = 0$. Therefore, a routing matrix R can be represented as the adjacency matrix of an expander graph. The estimation error via expander graph using unweighted ℓ_1 minimization has been well

[1] A bipartite graph $G = (A, B, E)$ is a graph whose nodes can be divided into two disjoint sets A and B such that every edge in set E connects a node in A to one in B [24].

studied in [23], and then [13] presents a relaxing condition on expander graph when the left degree is not equal. For simplicity, the statement and proof of our result (Theorem 1) are closely related to the results of [23], although the Theorem can be extend to the relaxing condition of [13]. Our contribution here is the modest adaptation from unweighted ℓ_1 to weighted ℓ_1 minimization.

In Theorem 1, we use S^c to denote the complement of S for any set $S \subset \{1 \cdots n\}$. Also, for any vector $\boldsymbol{v} \in \mathbb{R}^n$, we define $\boldsymbol{v}_S \in \mathbb{R}^n$ such that $(v_S)_i = v_i$ if $i \in S$, and $(v_S)_i = 0$ otherwise. We use notation for an ℓ_1 norm restricted to coordinates in a set S as $\|x\|_{S,\,1}$.

Theorem 1. *Let R be a adjacency matrix of a $(2k, \epsilon)$-expander graph. Assume that $\boldsymbol{x}$ is the true (unknown) vector and $\hat{\boldsymbol{x}}$ is the solution of (6) with $\omega_{\min} > 2\omega_{S,\,\max}\alpha(\epsilon)$. Let S be the set of k largest (in magnitude) coefficients of $\boldsymbol{x}$, then*

$$\left\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_1 \le \frac{2\|W\boldsymbol{x}\|_{S^c,1}}{\omega_{\min} - 2\omega_{S,\max}\alpha(\epsilon)}\right.$$

where

- *W is the diagonal matrix with diagonal entries ω_i.*
- *$\omega_{\min}$ is the minimum value of $\boldsymbol{\omega}$ and $\omega_{S,\,\max}$ is the maximum value of $\boldsymbol{\omega}_S$.*
- *$\alpha(\epsilon) = (2\epsilon)/(1 - 2\epsilon)$.*

Proof. see the Appendix. □

Theorem 1 gives the upper bound of weighted ℓ_1 minimization if routing matrix satisfies the corresponding condition. The bound depends on the choice of $\boldsymbol{\omega}$ and the value of $n - k$ *smallest* items of $\boldsymbol{x}$. Note that it degenerate to Theorem 1 in [23] if $\omega_1 = \cdots = \omega_n$.

4.3. Determine weights ω_i

While Theorem 1 presents the recovery error via weighted ℓ_1 minimization, it is still unclear how to set the weights. As described before, small weights could be used to the links with high congested probability, while large weights could be used to the links with low congested probability. So the weight should vary inversely with corresponding link prior probability. Here we proposed an optional method to determine the weights using Maximum A-Posteriori (MAP) estimation. We inspired by the knowledge from machine learning field, that maximizing the posterior distribution is equivalent to minimizing a regularized function [25]. For example, if $\boldsymbol{x}$ is imposed the Laplacian prior, the ℓ_1 minimization can be regarded as MAP estimator [26]. Our problem is somewhat different since we can only get the congested probability, not the probability distribution of loss rates. However, we shall see that the result is similar under the following assumptions:

- **A.1:** For normal links, the loss rates are close to zero, therefore $x_i \approx 0$ if $e_i \in E_n$.
- **A.2:** For congested links, the loss rates are large and are approximately same, therefore if $e_i, e_j \in E_c$.

The first is reasonable since normal links drop few packets. The second assumption can be violated if some congested links are *much more* congested. However, we notice that the success of our weighted ℓ_1 minimization bases on its ability to reveal *relative importance* of x_i rather than their precise values.

4.3.1. Setting weights using MAP

In the MAP estimation, we find the most likely $\boldsymbol{x}$ in (2) given the prior conditions. In other words, we look for the vector $\hat{\boldsymbol{x}}$ that maximizes the conditional probability of vector $\boldsymbol{x}$ given the measurement $\boldsymbol{y}$, i.e.,

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}} P(\boldsymbol{X} = \boldsymbol{x} | \boldsymbol{Y} = \boldsymbol{y}) \tag{8}$$

where we denote $\boldsymbol{X}$ the random vector of dimension n representing the states of links, and $\boldsymbol{Y}$ the random vector of dimension k representing the resulting states of paths. Applying Bayes' rule,

$$P(\boldsymbol{X} = \boldsymbol{x} | \boldsymbol{Y} = \boldsymbol{y}) = \frac{P(\boldsymbol{X} = \boldsymbol{x}) \cdot P(\boldsymbol{Y} = \boldsymbol{y} | \boldsymbol{X} = \boldsymbol{x})}{P(\boldsymbol{Y} = \boldsymbol{y})}.$$

As the term $P(\boldsymbol{Y} = \boldsymbol{y})$ only depends on the measurement, not on the $\boldsymbol{x}$, we have the following equivalent optimization problem

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}} P(\boldsymbol{Y} = \boldsymbol{y} | \boldsymbol{X} = \boldsymbol{x}) \cdot P(\boldsymbol{X} = \boldsymbol{x}). \tag{9}$$

We assume X_i $(i = 1, 2, \ldots, n)$ are independent random variables,

$$P(\boldsymbol{X} = \boldsymbol{x}) = \prod_{i=1}^{n} P(X_i = x_i). \tag{10}$$

It is difficult (even impossible) to measure the prior probability $P(X_i = x_i)$. However, we shall see latter that it is possible to get the prior congested probability $\boldsymbol{p}$. Thanks to assumptions **A.1** and **A.2**, we can write (we replace $\approx$ with $=$ in the assumptions for convenience, just remember the approximate characteristic in final result)

$$P(X_i = x_i) = \begin{cases} p_i & x_i \approx x_c, \\ 1 - p_i & x_i \approx 0. \end{cases}$$

If we define a function $I(t)$ such that $I(t) = 1$ when $t = x_c$ and 0 when $t = 0$, (10) can be written as

$$P(\boldsymbol{X} = \boldsymbol{x}) = \prod_{i=1}^{n} p_i^{I(x_i)} (1 - p_i)^{(1 - I(x_i))}.$$

Since $P(\boldsymbol{Y} = \boldsymbol{y} | \boldsymbol{X} = \boldsymbol{x})$ equals zero everywhere except $\boldsymbol{y} = R\boldsymbol{x}$, the MAP problem (8) is converted to

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}} P(\boldsymbol{X} = \boldsymbol{x}) \quad \text{subject to} \quad \boldsymbol{y} = R\boldsymbol{x}. \tag{11}$$

Taking the logarithm of $P(\boldsymbol{X} = \boldsymbol{x})$ and ignoring the terms that unrelated to $\boldsymbol{x}$, we obtain the optimization problem

$$\begin{aligned} \hat{\boldsymbol{x}} &= \arg\max_{\boldsymbol{x}} \prod_{i=1}^{n} p_i^{I(x_i)} (1 - p_i)^{(1 - I(x_i))} \\ &= \arg\max_{\boldsymbol{x}} \sum_{i=1}^{n} I(x_i) \log \frac{p_i}{1 - p_i} \\ &= \arg\min_{\boldsymbol{x}} \sum_{i=1}^{n} I(x_i) \log \frac{1 - p_i}{p_i} \quad \text{subject to} \quad \boldsymbol{y} = R\boldsymbol{x}. \end{aligned} \tag{12}$$

Although the problem of (12) is somewhat similar to that in (6), it is far more difficult to solve because of the discrete and the non-convex natures of $I(x_i)$. We simply linearize $I(x_i)$ to $k|x_i|$ where $k = 1/x_c$, then

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \sum_{i=1}^{n} \log \frac{1 - p_i}{p_i} |x_i| \quad \text{subject to} \quad \boldsymbol{y} = R\boldsymbol{x}. \tag{13}$$

which is weighted ℓ_1 minimization with weights $w_i = \log(1 - p_i)/p_i$.

Note that compare with the unweighted ℓ_1 minimization, we can consider (13) exploits the temporal correlations of link congestion. That is, if a link is congested frequently before, it is likely has large loss rates currently, which is realized by setting a small weight in (13).

4.3.2. Estimation of the prior

One way to estimate $\boldsymbol{p}$ has been studied in [10]. They use the path congested probability to estimate the link congested probability based on the boolean algebra Eq. (4). The congested probability of paths can be estimated via multiple snapshots. Let $\boldsymbol{S}$ denote the random binary vector of dimension r representing the states of paths, then $P(S_i = 1) = N_i/N$ and $P(S_i = 0) = (N - N_i)/N$ if we observe path N_i snapshots congested in overall N snapshots. Similarly, let $\boldsymbol{T}$ denote the random binary vector of dimension n representing the states of links (note that $\boldsymbol{S}$ and $\boldsymbol{T}$ are binary randoms contrast to the real characters of

$\boldsymbol{X}$ and $\boldsymbol{Y}$ in last subsection). From (4), the following equations hold:

$$P(S_i = 0) = P\left(\bigvee_{k=1}^{n} r_{ik} \cdot T_k = 0,\ 1 \le k \le n\right) = \prod_{k=1}^{n} (1 - p_k)^{r_{ik}}.$$

Taking the logarithm on both sides,

$$\log P(S_i = 0) = \sum_{k=1}^{n} r_{ik} \log(1 - p_k) \qquad i = 1, 2, \ldots, r. \tag{14}$$

We still can not solve p_k from (14) since $r < n$. However, unlike the linear model, new independent equations can be created when combine multiple paths in (14). For example, let us consider the combination of two paths i and j. Then the following joint probability holds:

$$\log P(S_i = 0, S_j = 0) = \sum_{k=1}^{n} \{r_{ik} \vee r_{jk}\} \log(1 - p_k). \tag{15}$$

The meaning of (15) is straightforward: the probability of two good paths equals that of all good links which are passed by them.

Similarly, the joint probability in (15) can be estimated by $P(S_i = 0, S_j = 0) = N_{ij}/N$, where N_{ij} denotes number of snapshots that both paths are good. The equations (15) can offer another $r(r-1)/2$ equations since i, j may range from $1 \le i < j \le r$ (we let $i < j$ because $P(S_i = 0, S_j = 0) = P(S_j = 0, S_i = 0)$). So combining (14) and (15), we have a total of $r(r+1)/2$ linear constraints. It was shown in [27] that the number of links between r nodes scales as $O(r\log r)$ if the underlying network has a power-law topology, so $r(r+1)/2 \gg n$. Consequently, although some equations are linearly dependent, we can determine prior probabilities $\boldsymbol{p}$ uniquely[10].

4.4. Overview of the proposed scheme and discussion

The overview of the proposed scheme is shown in Fig. 2. After estimating the prior probability p_i of each link, we use weighted ℓ_1 minimization to implement loss tomography.

We adopt the algorithm described in Section 4.3.2 to estimate the prior probability. The algorithm uses a number of snapshots to learn about the congestion probabilities. After getting the prior probability $\boldsymbol{p}$, the weight w_i of link e_i is determined by its congested probability p_i. In Section 4.3.1, we adopt the MAP estimation to find $\boldsymbol{w}$ with $w_i = \log(1 - p_i)/p_i$. Once weights are determined, we can formulate the weighted ℓ_1 minimization of (6) based on the routing matrix R and path measurements $\boldsymbol{y}$. Then link loss rates are estimated by solving weighted ℓ_1 minimization and congested links (denoted by s_d in Fig. 2) are identified according to threshold ϕ_{th}.

In practical application, we can update $\boldsymbol{p}$ by taking advantage of former diagnostic results (the dotted line in Fig. 2), thus easing the expensive time consumption of computing prior probabilities. The method of use multiple S_d to estimate $\boldsymbol{p}$ is straightforward. Since the boolean random variable which represents link state follows Bernoulli distribution, we can just use the following equation to estimate $\boldsymbol{p}$:

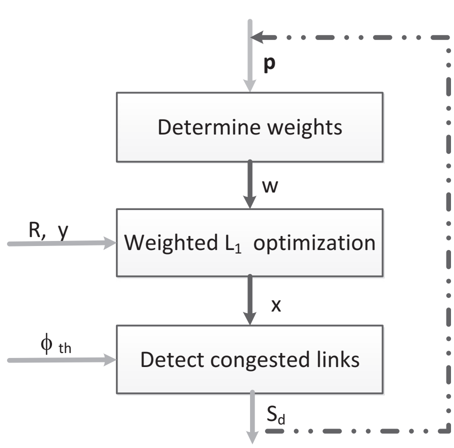


Fig. 2. The detection part of the proposed scheme.

$$\widehat{p_i} = \frac{congestion\ times\ of\ e_i}{all\ diagnosis\ times},$$

which keeps the maximum likelihood estimator for Bernoulli distribution [25].

4.4.1. Discussion

Note that the weights also may not be functions of $\boldsymbol{p}$. In fact, we can treat them as the free parameters in the model, then calculate their optimum values based on additional information. For example, In [20], the authors determine weights using partial support information in $\boldsymbol{x}$, and in [21], weights are determined by the prior information about the probability of some entries being nonzero. However, we can not obtain such information but the prior probabilities in loss tomography, hence make (13) optimization feasible and reasonable.

The idea of utilizing prior probabilities to infer network state has been raised in [10], where the authors proposed an algorithm *CLINK* to locate the congested links. *CLINK* tries to find the smallest set of weighted links whose congestions can explain the observed measurements. There are significant differences between weighted ℓ_1 and *CLINK* that make our method a totally different tomography scheme.

- In [10], the task of identifying the congested links is formulated as the weighted set cover problem (WSCP), a known NP-complete problem. Therefore, a greedy heuristic algorithm is adopted to get the approximate solution. While in our scheme, the weighted ℓ_1 optimization (13) is just a linear program and can be solved by any LP optimizer.
- Since the tomography method in [10] is based on the boolean model (each link is estimated as either "good" or "bad"), the only function of it is congestion localization. In this work, the link loss rates are analog, which permits us not only to locate the congested links, and also to identify the congestion level of each link.

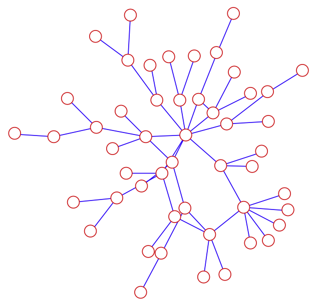
Both the differences are important, that is, the proposed scheme allows us to infer link loss rates, which is more than congestion localization. Even in the case of congestion localization, different solutions can be expected. Weighted ℓ_1 minimization is essentially a *combination* of ℓ_1 minimization and *CLINK*: we use the prior probabilities but adopt them as the weights of ℓ_1 norm.

5. Performance evaluation

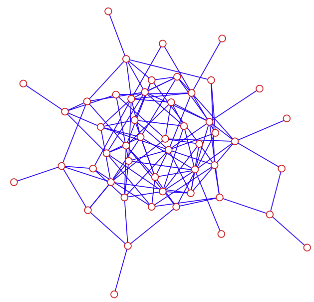
5.1. Network topology and routing matrix

We evaluate the proposed scheme by simulations in both synthetic topologies and real topologies. The synthetic topologies are used to test our algorithm in extreme cases while real topologies represent practical situation. Especially, we use recent topology generator COLD (Combined Optimized Layered Design) [28] to generate random graphs with different average node degree. The generator includes simple parameters to be tuned to produce much different types of topologies that is very appropriate to test new algorithms. Fig. 3 shows the synthetic topologies (50 nodes) used in this section, including low node degree topology (LDT), middle node degree topology (MDT) and high node degree topology (HDT)[2] We use LDT and HDT to represent the

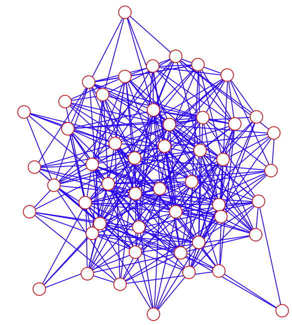
[2] We use the parameters k_2, k_3 in COLD to generate these topologies, and their values are set to (0.0001, 100), (0.0018, 50) and (0.01, 1) for LDT MDT and HDT respectively. Other parameters use the recommended settings [29].



(a) Low node degree topology (hub and leaf style)



(b) Middle node degree topology



(c) High node degree topology (clique style)

Fig. 3. Synthetic topologies with different node degree (created by COLD and modified by MATLAB).

extreme situations, that is, there are only a few links (for LDT) or a great many links (for HDT) to be solved in the topology.

In each topology, we choose 20 nodes with the least out-degree to perform measurements (measurement nodes). The routing matrix of each network topology is first created by containing the shortest paths between any two measurement nodes. Then we delete all links and nodes that do not contribute to any probe path in routing matrix, since link state is unidentifiable if it is not covered by any probe path. Table 1 summarizes the parameters and routing matrices of synthetic topologies. As expected, the rank of routing matrix R is less than number of links.

5.2. Simulator

We built a simulator based on simulated network topology and routing matrix. We first assume that each link e_k in the network is congested with a prior probability p_k (p_k between 0 and 1). Then the following assumptions are made in each simulation runs:

- For each link e_k, we determine whether it will be congested or not, such that we respect the congested probability p_k determined in the beginning.
- To each link, we assign a packet loss rate according to the loss rates model LM1 in [7] (also similar to model in [8]), where congested links have loss rates uniformly distributed in [0.05, 0.1] and normal links have loss rates [0, 0.01].
- For each link e_k, the actual loss follows Bernoulli process, where packets are dropped with a fixed probability such that we respect the packet loss rate determined in the previous step.
- We measure the packet loss rate of each path according to the routing matrix. A path is identified as congested if its loss rate is above $1-(1-0.01)^d$ as proposed in [10] where d is the length of the path.

We take 30 snapshots to learn the prior probability $\boldsymbol{p}$ as suggested in [10]. Based on $\boldsymbol{p}$, we can implement loss tomography in subsequent measurements using the weighted ℓ_1 minimization.

Table 1
Topologies used in simulation.

Topology	# Links	# Probe paths	Rank of R
LDT	41	190	35
MDT	77	190	56
HDT	153	190	104

5.3. Metrics and alternative solutions

The performance of proposed scheme is evaluated in two aspects: inferring loss rates and identifying congested links. We define the estimation error of loss rates is the normalized ℓ_1 norm as follows:

$$ERR = \frac{\|\boldsymbol{l}-\hat{\boldsymbol{l}}\|_1}{\|\boldsymbol{l}\|_1}$$

where $\boldsymbol{l}$ and $\hat{\boldsymbol{l}}$ are the true and estimated loss rates vectors respectively. For congestion localization, two metrics are adopted to evaluate the proposed scheme: the detection rate (*DR*) and false positive rate (*FR*). The *DR* is the probability of correctly locating the congested link(s), and *FR* is the percentage of links that are normal but are identified as congested, we express the two metrics as follows:

$$DR = \frac{|S_d \cap S_a|}{|S_a|} \qquad FR = \frac{|S_d \setminus S_a|}{|S_d|}$$

where S_d is the set of links that are *diagnosed* as congested and S_a is the *actual* congested links. Both *DR* and *FR* vary from zero to one, and the closer *DR*'s value to one, the better performance the algorithm is, yet *FR* is on the contrary.

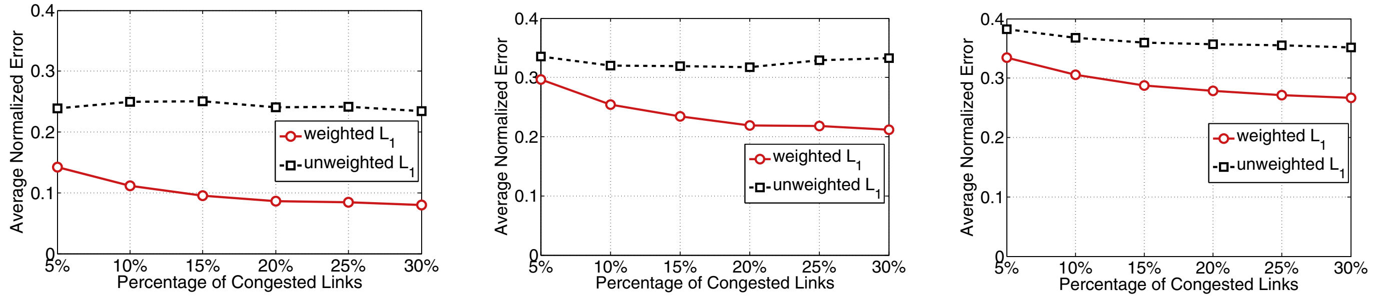
The proposed method is compared with two other closely related tomography algorithms: (unweighted) ℓ_1 minimization [13,17] and *CLINK* [10]. Note that some loss tomography algorithms use ℓ_1 minimization but add some other terms. For example, *Norm* method in [30] adds a ℓ_2 norm to handle the measurement noise. We do not compare with these methods since they always introduce some parameters that should be chosen carefully. Furthermore, our method can easily absorb these added terms, just like unweighted ℓ_1 minimization.

Since *CLINK* is not able to infer the loss rate of links, for loss rates estimation, we only compare the results of weighted ℓ_1 and unweighted ℓ_1. All the results of each simulation are repeated 1000 times, then the average values for the results are calculated.

5.4. Results: loss rates estimation

We first compare the performance of our scheme with the ℓ_1 minimization in loss rates estimation. In order to examine how different tomography algorithms perform as the lossy links increases, we choose different prior probability vector $\boldsymbol{p}$ such that the percentage of lossy links in networks increases.

Fig. 4 presents the average loss rates estimation error (*ERR*) versus percentage of congested links using weighted and unweighted ℓ_1 minimization. The percentage of congested links varies from 5% to 30%. For all the topologies, weighted ℓ_1 has lower average estimation error than unweighted ℓ_1 minimization since it utilizes the *prior knowledge* about network state. We also find that for a specific



(a) Low node degree topology (b) Middle node degree topology (c) High node degree topology

Fig. 4. Average loss rates estimation error *ERR* vs. percentage of congested links for different topologies.

algorithm, the topology LDT has the lowest estimation error and HDT has the highest. This is because the LDT is *highly identifiable* (from Table 1, the rank of routing matrix is approximately equal to the number of links for LDT) and we can get x from (2) accurately. While for HDT, there are too many links to be identified, thus we can only get a rather rough result for any tomography algorithm. The differences of tomography results among the three topologies will also be elaborated in Section 5.5. A notable point is that the value of normalized error is the ratio of $\|\boldsymbol{l}-\hat{\boldsymbol{l}}\|_1$ to $|\boldsymbol{l}\|_1$, both of which increase as the number of congested links increases, we can see from Fig. 4 that the value of $\|\boldsymbol{l}-\hat{\boldsymbol{l}}\|_1$ may not increases as fast as $|\boldsymbol{l}\|_1$.

The superiority of our method also can be seen in Fig. 5, where the estimation accuracy is characterized as the fraction of all links that are correctly estimated by a algorithm. Links are correctly estimated if the estimated value is within 1 percentage point of the real loss rate. Obviously, high fraction can be expected for a good tomography algorithm. Fig. 5 presents the result for the middle node degree topology. It shows that weighted ℓ_1 has better performance in loss rate estimation, which is consistent with the conclusion in Fig. 4.

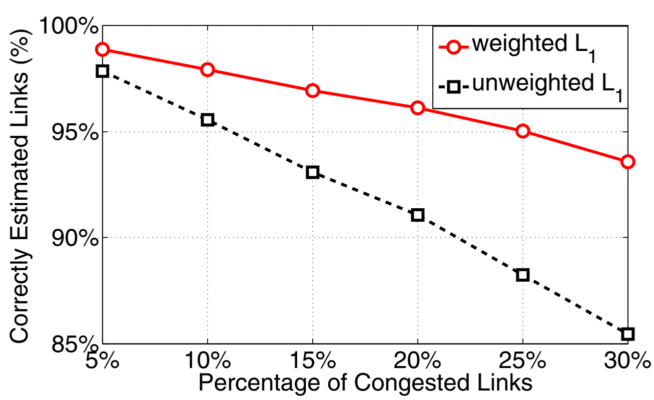


Fig. 5. The percentage of links that are correctly estimated vs. percentage of congested links for the middle node degree topology.

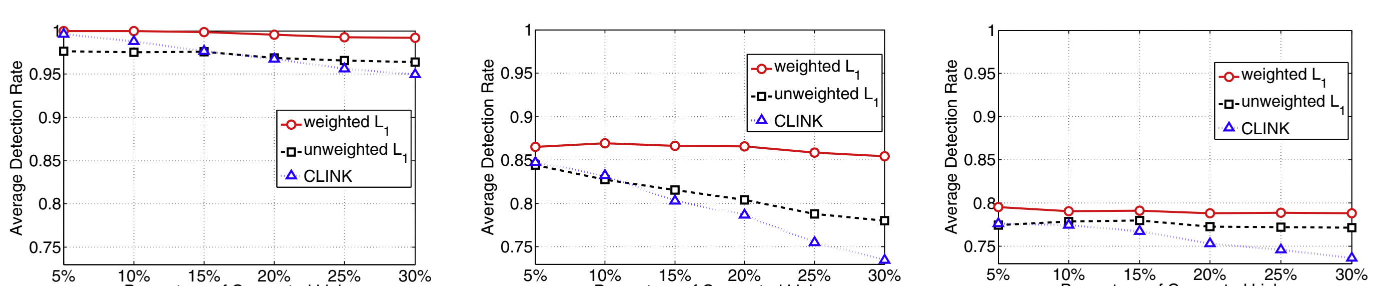
5.5. Results: locating congested links

We also evaluate the weighted ℓ_1 minimization as applied to locating congested links in network. The threshold is set to $\phi_{th} = 97\%$ for (weighted and unweighted) ℓ_1 minimization. Figs. 6 and7 show the average detection rate and false positive rate respectively, with three tomography algorithms in different topologies.

We first analyze the influence of different topologies. The subfigures of Figs. 6 and7 have the same scales on the y-axis. For the low node degree topology, the detection rate is quite high, i.e., almost all the congested links can be identified, and for the high node degree topology, all algorithms have a relatively poor performance. The reason is that—as mentioned in Section 5.4—LDT is highly identifiable, which makes us much sure about the link characteristics. On the contrary, HDT has so many links that beyond the ability of any algorithm, the only solution to get a more accurate result is probably adding more probe paths. Since LDT and HDT are created to represent two kinds of extreme situations, we expect real topologies are more like the MDT topology, in which weighted ℓ_1 has a higher detection rate, especially when the number of congested links increases.

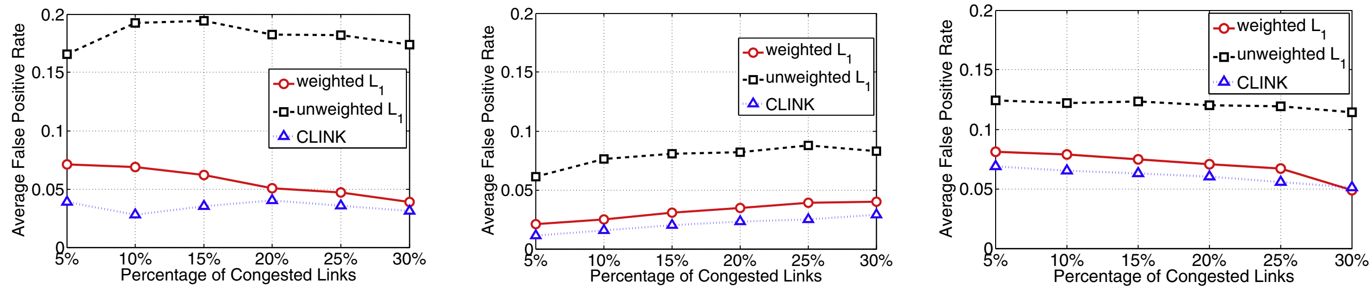
Comparing with unweighted ℓ_1, weighted ℓ_1 has a significant advantage by detection rate and false positive rate. For instance, when 30% of the links are lossy, weighted ℓ_1 detects 85% of the lossy links with a false positive rate of 4% for the MDT topology, while the unweighted counterpart has a detection rate of 78% and false positive rate of 9%. This is because weighted ℓ_1 uses previous measurement snapshots to learn about the prior probabilities and embody them in weights to determine the links which will be congested most likely.

As illustrated in Figs. 6 and7, *CLINK* always has the lowest false positive rate, this phenomenon agrees with the fact in [22] that boolean tomography aims to find the *minimum* set of links shared between bad paths and often identifies fewer lossy links. However, it has relatively low detection rate in doing so. Comparing with *CLINK*, weighted ℓ_1 has higher detection rate with similar false positive rate. Since *CLINK* also utilizes the prior probabilities to locate congested links, the results seem a little interesting. Our interpretation is the following:



(a) Low node degree topology (b) Middle node degree topology (c) High node degree topology

Fig. 6. Average detection rate vs. percentage of congested links for different topologies.



(a) Low node degree topology (b) Middle node degree topology (c) High node degree topology

Fig. 7. Average false positive rate vs. percentage of congested links for different topologies.

Table 2
Real topologies.

Topology	# Links	# Probe paths	Rank of R
AS3257	183	1161	157
AS7018	252	1485	183

Table 3
Accuracy of the weighted ℓ_1, unweighted ℓ_1 and *CLINK*.

Topology	Weighted ℓ_1			Unweighted ℓ_1			*CLINK*	
	DR	*FR*	*ERR*	*DR*	*FR*	*ERR*	*DR*	*FR*
AS3257	96.5%	3.0%	0.11	92.8%	6.2%	0.17	90.3%	2.2%
AS7018	92.2%	3.2%	0.21	85.1%	8.5%	0.34	75.5%	2.2%

- *CLINK* does not consider the loss rate of congested paths and it simply aims to find the minimum set of (weighted) links shared between such paths.
- The task of identifying the congested links in *CLINK* is formulated as a NP-complete problem, and it adopts a $\log(n + 1)$-approximation greedy algorithm, which might not be as accurate as weighted ℓ_1.

In fact, we found that the performance of boolean tomography depends on the number of probe paths, even them do *not* increase the rank of routing matrix R.

5.6. Real topologies

Besides the synthetic topologies, the algorithm is tested in real topologies. Specifically, two router-level ISP topologies discovered by Rocketfuel [31] are adopted: AS3257 and AS7018. We conduct the experiments under the same conditions as synthetic topologies except that we use the nodes with degree one to perform measurements. Table 2 summaries the real topologies. The results are shown in Table 3, when 20% links are congested. Table 3 leads us to the similar conclusion with synthetic topologies: weighted ℓ_1 performs better than the two alternatives, hence our method is applicable for practical loss tomography. In addition, we can also find that the performance of the real topologies lie between the synthetic topologies LDT and MDT.

6. Conclusion

In this work, we proposed a new loss tomography scheme. The scheme adopts the concept of compressed sensing and uses weighted ℓ_1 minimization to estimate the loss rate in the network. The weights in weighted ℓ_1 minimization are determined by links prior probabilities of being congested, which can be estimated by multiple path measurements. By reformulating ℓ_1 minimization with the weights, the scheme combines the best features of *CLINK* and ℓ_1 minimization tomography.

We evaluate the proposed scheme by simulations in both synthetic topologies and real topologies. The synthetic topologies are used to test tomography algorithms in different scenarios, while real topologies represent practical situation. Simulation results show that proposed tomography method performs better than the unweighted counterpart and boolean tomography *CLINK* in terms of estimation accuracy, detection rate and false positive rate. Some technical issues remain in the proposed scheme, for example, since our work here focuses on the estimation of link loss rate, a nature extension is the detection of anomalies in the network.

Acknowledgment

This research was supported by the National Natural Science Foundation of China (no. 61671112).

Appendix A. Proof of Theorem 1

Let $\boldsymbol{e} = \hat{\boldsymbol{x}} - \boldsymbol{x}$, since $R\boldsymbol{x} = R\hat{\boldsymbol{x}}$, obviously $R\boldsymbol{e} = 0$. We have:

$$\begin{aligned}
||W\boldsymbol{x}||_1 &\geq ||W\hat{\boldsymbol{x}}||_1 = ||W(\boldsymbol{e} + \boldsymbol{x}))||_1 \\
&= ||W\boldsymbol{e} + W\boldsymbol{x}||_{S,1} + ||W\boldsymbol{e} + W\boldsymbol{x}||_{S^c,1} \\
&\geq ||W\boldsymbol{x}||_{S,1} - ||W\boldsymbol{e}||_{S,1} + ||W\boldsymbol{e}||_{S^c,1} - ||W\boldsymbol{x}||_{S^c,1} \\
&= ||W\boldsymbol{x}||_1 - 2||W\boldsymbol{x}||_{S^c,1} + ||W\boldsymbol{e}||_1 - 2\,||W\boldsymbol{e}||_{S,1}.
\end{aligned}$$

where we use notation for an ℓ_1 norm restricted to coordinates in a set S as $\|x\|_{S,\,1}$.

Therefore, we have $2||W\boldsymbol{x}||_{S^c,1} \geq ||W\boldsymbol{e}||_1 - 2\,||W\boldsymbol{e}||_{S,1}$. Since $\omega_{\min}$ denote the minimum value of $\boldsymbol{\omega}$ and $\omega_{S,\,\max}$ denote the maximum value of $\boldsymbol{\omega}_S$, then

$$\begin{aligned}
2||W\boldsymbol{x}||_{S^c,1} &\geq ||W\boldsymbol{e}||_1 - 2\,||W\boldsymbol{e}||_{S,1} \\
&\geq \omega_{\min}\,||\boldsymbol{e}||_1 - 2\omega_{S,\max}\,||\boldsymbol{e}||_{S,1}.
\end{aligned} \tag{A.1}$$

By using the result directly from Lemma 1 in [23], the following inequality holds:

$$||\boldsymbol{e}||_{S,1} \leq \alpha(\epsilon)\ ||\boldsymbol{e}||_1.$$

where $\alpha(\epsilon) = (2\epsilon)/(1 - 2\epsilon)$. Substituting the above equation into (A.1) yields:

$$2||W\boldsymbol{x}||_{S^c,1} \geq \omega_{\min}\ ||\boldsymbol{e}||_1 - 2\omega_{S,\max}\alpha(\epsilon)\ ||\boldsymbol{e}||_1.$$

Therefore we have:

$$||\boldsymbol{e}||_1 = ||\hat{\boldsymbol{x}} - \boldsymbol{x}||_1 \leq \frac{2||W\boldsymbol{x}||_{S^c,1}}{\omega_{\min} - 2\omega_{S,\max}\alpha(\epsilon)}.$$

References

[1] M. Coates, A.O. Hero III, R. Nowak, B. Yu, Internet tomography, IEEE Signal Process. Mag. 19 (3) (2002) 47–65.

[2] L. Ma, T. He, K.K. Leung, A. Swami, D. Towsley, Identifiability of link metrics based on end-to-end path measurements, Proc. ACM IMC, (2013).

[3] K. Deng, Y. Li, W. Zhu, J.S. Liu, et al., Fast parameter estimation in loss tomography for networks of general topology, Ann. Appl. Stat. 10 (1) (2016) 144–164.

[4] R. Cáceres, N.G. Duffield, J. Horowitz, D.F. Towsley, Multicast-based inference of network-internal loss characteristics, IEEE Trans. Inf. Theory 45 (7) (1999) 2462–2480.

[5] N.G. Duffield, F.L. Presti, V. Paxson, D. Towsley, Inferring link loss using striped unicast probes, INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, vol. 2, IEEE, 2001, pp. 915–923.

[6] T. Bu, N. Duffield, F.L. Presti, D. Towsley, Network tomography on general topologies, ACM SIGMETRICS Performance Evaluation Review, vol. 30, ACM, 2002, pp. 21–30.

[7] V. Padmanbhan, L. Qiu, H. Wang, Server-based inference of internet performance, In Proc. of IEEE INFOCOM, (2003).

[8] H.X. Nguyen, P. Thiran, Network loss inference with second order statistics of end-to-end flows, Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, ACM, 2007, pp. 227–240.

[9] D. Ghita, K. Argyraki, P. Thiran, Network tomography on correlated links, Proc. IEEE SIGCOMM, (2010).

[10] H.X. Nguyen, P. Thiran, The boolean solution to the congested IP link location problem: theory and practice, Proc. IEEE INFOCOM, (2007).

[11] P. Sattari, A. Markopoulou, C. Fragouli, M. Gjoka, A network coding approach to loss tomography, IEEE Trans. Inf. Theory 59 (3) (2013) 1532–1562.

[12] W. Xu, E. Mallada, A. Tang, Compressive sensing over graphs, Proc. IEEE INFOCOM, (2011).

[13] M.H. Firooz, S. Roy, Link delay estimation via expander graphs, IEEE Trans. Commun. 62 (1) (2014) 170–181.

[14] D.L. Donoho, Compressed sensing, IEEE Trans. Inf. Theory 52 (4) (2006) 1289–1306.

[15] R. Berinde, A.C. Gilbert, P. Indyk, H. Karloff, M.J. Strauss, Combining geometry and combinatorics: A unified approach to sparse signal recovery, Communication, Control, and Computing, 2008 46th Annual Allerton Conference on, IEEE, 2008, pp. 798–805.

[16] W. Xu, B. Hassibi, Efficient compressive sensing with deterministic guarantees using expander graphs, Information Theory Workshop, 2007. ITW'07. IEEE, IEEE, 2007, pp. 414–419.

[17] T. Matsuda, M. Nagahara, K. Hayashi, Link quality classifier with compressed sensing based on $\ell_1 - \ell_2$ optimization, IEEE Commun. Lett. 15 (10) (2011) 1117–1119.

[18] X. Fan, X. Li, Network tomography via sparse bayesian learning, IEEE Commun. Lett. 21 (4) (2017) 781–784.

[19] E.J. Candes, M.B. Wakin, S.P. Boyd, Enhancing sparsity by reweighted 1 minimization, J. Fourier Anal. Appl. 14 (5–6) (2008) 877–905.

[20] M.P. Friedlander, H. Mansour, R. Saab, O. Yilmaz, Recovering compressively sampled signals using partial support information, IEEE Trans. Inf. Theory 58 (2) (2012) 1122–1134.

[21] M.A. Khajehnejad, W. Xu, A.S. Avestimehr, B. Hassibi, Weighted l1 minimization for sparse recovery with prior information, 2009 IEEE International Symposium on Information Theory, IEEE, 2009, pp. 483–487.

[22] S. Zarifzadeh, M. Gowdagere, C. Dovrolis, Range tomography: combining the practicality of boolean tomography with the resolution of analog tomography, Proceedings of the 2012 ACM Conference on Internet Measurement Conference, ACM, 2012, pp. 385–398.

[23] R. Berinde, P. Indyk, Sparse recovery using sparse random matrices, (2008). MIT-CSAIL Technical Report http://dspace.mit.edu/handle/1721.1/40089.

[24] J. Clark, D.A. Holton, A First Look at Graph Theory, World Scientific Publishing Co Inc, 1991.

[25] Y. Anzai, Pattern Recognition & Machine Learning, Elsevier, 2012.

[26] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B (1996) 267–288.

[27] Y. Chen, D. Bindel, H. Song, R.H. Katz, An algebraic approach to practical and scalable overlay network monitoring, ACM SIGCOMM Computer Communication Review, vol. 34, ACM, 2004, pp. 55–66.

[28] R. Bowden, M. Roughan, N. Bean, Cold: Pop-level network topology synthesis, Proceedings of the 10th ACM SIGCOMM CoNEXT, ACM, 2014, pp. 173–184.

[29] R. Bowden, Cold: pop-level topology generator, (https://github.com/rhysbowden/COLD).

[30] H.H. Song, L. Qiu, Y. Zhang, Netquest: a flexible framework for large-scale network measurement, ACM SIGMETRICS Performance Evaluation Review, vol. 34, ACM, 2006, pp. 121–132.

[31] N. Spring, R. Mahajan, D. Wetherall, Measuring isp topologies with rocketfuel, Proc. ACM SIGCOMM 32 (4) (2002) 133–145.