Computer Networks 55 (2011) 3533–3547

Contents lists available at ScienceDirect

Computer Networks

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

Joint time–frequency sparse estimation of large-scale network traffic

Dingde Jiang [a,b,*], Zhengzheng Xu [a], Zhenhua Chen [a], Yang Han [a], Hongwei Xu [a]

[a] College of Information Science and Engineering, Northeastern University, Shenyang, China
[b] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China

ARTICLE INFO

Article history:
Received 3 January 2011
Received in revised form 13 June 2011
Accepted 28 June 2011
Available online 23 July 2011

Keywords:
Network traffic
Time–frequency analysis
Sparse decomposition
Traffic estimation
Traffic engineering

ABSTRACT

When 3G, WiFi, and WiMax technologies are successfully applied to access networks, current communication networks become more and more complex, more and more heterogeneous, and more difficult to manage. Moreover, network traffic exhibits the increasing diversities and concurrently shows many new characteristics. The real-time end-to-end demand urges network operators to learn and grasp traffic matrix covering their networks. However, unfortunately traffic matrix is significantly difficult directly to attain. Despite many studies made previously about traffic matrix estimation problem, it is a significant challenging to obtain its reliable and accurate solution. Here we propose a novel approach to solve this problem, based on joint time–frequency analysis in transform domain. Different from previous methods, we analyze the time–frequency characteristics about traffic matrix and build the time–frequency model describing it. Generally, traffic matrix can be divided into tendency terms and fluctuation terms. We find that traffic matrix in time–frequency domain owns the more obvious sparsity than in time domain. Obviously, its tendency terms and fluctuation terms also have the lower dimensions in time–frequency domain. This brings us into the field of compressive sensing that is a generic technique for data reconstruction. Additionally, we take into account updating time–frequency model presented with link loads to make our model adaptive. Finally, comparative analysis in two real backbone networks confirms that the accuracy, stability, and effectiveness of our approach.

1. Introduction

With 3G, WiFi, and WiMax technologies be successfully applied to access networks, current communication networks are growing more complex, more heterogeneous, and more difficult to manage and maintain than before. At the same time, new kinds of traffic arise quickly with new sorts of network applications. However, this increases new difficulty for network operation and maintenance in turn. Strictly speaking, the Internet's strictly layered architecture products different topologies while traffic matrix describes the traffic exchanges over these given network topologies [1]. Traffic matrix in a network consists of all Origin–Destination (OD) pairs over it, which reflects traffic exchange from ingress nodes to egress nodes in the whole network. Thereby, traffic matrix provides the valuable information about the current network status to network operators and researchers [2]. Many network engineering and management tasks, for instance network design, routing optimization, anomaly detection and so on, depend heavily on the available and accurate traffic matrix [3,4]. However, it is a significant challenge to obtain its reliable and accurate estimation.

Generally, the fact that traffic matrix in backbone networks is extremely difficult to estimate is not only because traffic matrix itself hold all kinds of characteristics such as temporal and spatial relations, heavy distribution,

* Corresponding author at: College of Information Science and Engineering, Northeastern University, Shenyang, China.
E-mail address: jiangdd@mail.neu.edu.cn (D. Jiang).

1389-1286/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2011.06.027

self-similar nature and so forth, but also because the high-rate traffic exists in current networks [5]. More importantly, although traffic matrix, link loads, and route matrix are satisfied with linear constraints, traffic matrix estimation is a highly under-constrained inverse problem and consequently own the extremely ill-posed nature because the number of OD flows in a given backbone network is often much larger than that of links [4,6].

Although many studies were made about traffic matrix estimation problem, it is still extremely difficult to perform the reliable estimation in the real backbone network [7]. To capture precisely inherent properties of traffic matrix, statistic model [4], gravity model [6], independent model [8] and so forth are proposed. Unfortunately, statistic model is sensitive to prior information of traffic matrix, while some shortcoming exists in other models [3,9]. As a result, a dual approach to turn the constrained primal optimization problem under the gravity model into an unconstrained dual optimization problem can make a distributed estimation and achieve performance improvement [10]. Additionally, due to the temporal and spatial correlations of traffic matrix, a spatial model and Kalman filtering approach are suited [11]. The probability model combining traditional single-link (-flow) traffic model with the routing can capture global behavior of network traffic [12]. Nevertheless, these methods proposed above are still difficult to obtain the reliable estimation of traffic matrix. Hence, some synthetic methods are presented to generate traffic matrix so as to conduct the normal network activity [13,14]. Generally, the real traffic matrix exhibits intensively the low-dimension and sparse property [15]. Compressive sensing is a generic approach suited for reconstructing the sparse data [16]. Spatio-Temporal Compressive Sensing sufficiently considers the sparsity in traffic matrix [17]. As mentioned in [6], TomoGravity based on gravity model is a fast traffic matrix estimation method from link counts because it do not need the traffic of OD flows but only need to obtain link counts by simple network management protocol. In some networks, measuring OD traffic is very difficult, and thus traffic estimation such as TomoGravity is a useful method. Similarly, Sparsity Regularized Sigular Value Decomposition (SRSVD) in [17] either do not require measurement data on OD traffic. In contrast to TomoGravity and SRSVD, Principal Component Analysis (PCA) estimation in [15] need to directly measure the traffic of OD flows to model its corresponding model for performing traffic matrix estimation. Hence, PCA need the additional measurement overheads, but in general using the measured OD traffic can improve traffic matrix estimation results. In this paper, we also use the measured OD flow traffic to model our model.

However, most previous methods analyze and estimate traffic matrix only in the time domain. Traffic matrix as a time series does not only own the time domain properties, but should also hold the frequency domain characteristics. In this paper, we investigate traffic matrix estimation in the time–frequency domain. Firstly, by S-transform method, we analyze joint time–frequency properties of link loads and traffic matrix in the time–frequency domain. We find that they exhibit the more obvious sparsity in the time–frequency domain. Furthermore, their time–frequency distributions are mostly aggregated in the lower frequency, whereas they are relatively sparse in other frequency parts. Secondly, in terms of routing matrix, derive the constrained matrix of link loads and traffic matrix in the time–frequency domain. Thirdly, according to the above findings, we build joint time–frequency model about traffic matrix in certain time window, using the measured data. At the same time, use the time–frequency parameters of link loads as the updating variable to adjust this model to make it adaptive. Fourthly, we divide traffic matrix into tendency terms and fluctuation terms. Tendency terms are described with time–frequency model proposed in this paper. Then fluctuation terms are split further into sparse terms and non-sparse terms. For sparse terms, we exploit compressive sensing theory [16] to reconstruct, whereas non-sparse terms are inferred by sparse terms and link load deviation. Finally, we present our time–frequency algorithm to traffic matrix estimation and analyze the accuracy, stability, and effectiveness of our approach using real data from two backbone networks [18,19]. In our previous work, the simulated annealing and generalized inference method have been used for estimating traffic matrix [20]; By taking Mahalanobis distance as optimal metric, we have proposed a regressive inference method for traffic matrix estimation problem [21]. However, the two methods make traffic matrix estimations only in the time domain, using the statistical analysis theory. Due to space limitation, we here have no detailed comparison of them with the method proposed in this paper, but the analysis and comparison are to make in detail in near future work. The work most related to this paper is one in [22]. There we exploit time–frequency approach to analysis and extract the anomaly network traffic characteristics, while in this paper it is used to model and estimate traffic matrix in large-scale network.

The rest of this paper is organized as follows. Section 2 introduces S-transform used for time–frequency analysis and discusses the time–frequency characteristics of link loads and traffic matrix. Section 3 derives the time–frequency model and estimation algorithm. Section 4 presents the simulation results and analysis, and evaluates the accuracy, stability, and effectiveness of our approach. Finally, we conclude our work in Section 5.

2. Problem fomulation

Without loss of generalization, assume that there are n nodes and L links in a large-scale IP backbone network. Consequently, there exist $N = n^2$ OD flows. Generally, TM and link loads evolve with time. They are time sequences owning dynamic nature, and in a given network they follow certain constraints as well. Therefore, TM and link loads can be represented as $x(t)$ and $y(t)$, respectively, at a given time t. If so, constraint relations between them can be denoted into

$$y(t) = Ax(t), \tag{1}$$

where A is $L \times N$ routing matrix whose element A_{ij} is equal 1 if OD flow j traverses link i or zero otherwise. Usually, $L \ll N$ exists in a designated large-scale IP backbone network, with the result that this makes the problem denoted by Eq. (1)

highly under-constrained and thus ill-posed enough. How to overcome such under-constrained and ill-posed properties is main challenge faced for the moment.

2.1. Time–frequency analysis

In general, the time domain analysis of a signal only describes its property evolving over time but does not formulate its frequency characteristics. Different from this, the time–frequency analysis simultaneously capture the time and frequency nature of a signal by jointing time domain and frequency domain analysis. Thereby, the time–frequency analysis can more accurately extract the inherent properties of a signal and it is a important method of signal analysis.

We use S-transform [23] to perform time–frequency analysis. For a K-point discrete time series $h[kT]$, its S-transform can be denoted as follows [23].

$$S\left[jT,\frac{n}{KT}\right]=\sum_{m=-K/2}^{K/2-1}H\left[\frac{m+n}{NT}\right]e^{-\frac{2\pi^2m^2}{n^2}+\frac{j2\pi mj}{K}}, \tag{2}$$

where $n\neq 0$, $H\left[\frac{n}{KT}\right]$ is the Fourier transform of the above time series $h[kT]$. Likewise, discrete inverse S-transform be formulated into [23]:

$$h[kt]=\frac{1}{K}\sum_{n=0}^{K-1}\left\{\sum_{j=0}^{K-1}S\left[\frac{n}{KT},jT\right]\right\}e^{\frac{j2\pi nk}{K}}. \tag{3}$$

2.2. Time–frequency characteristics of link loads

According to (2) and (3), for each of L link loads in (1), one can calculate its S-transform in certain time window. Analyzing link loads from two real networks Abilene [18] and GÉANT [19], respectively, we find that each link loads in these two networks belongs to one of two types of time–frequency distributions. Fig. 1 plots these two main types of time–frequency distribution of link loads. From Fig. 1(a) and (b), one may see that time–frequency distributions of link loads in the Abilene principally focus on the lower frequency parts, while the higher frequency components only have a relative small value. Similarly, from Fig. 1(c) and (d), one can find that link loads in the GÉANT also own nearly the same time–frequency distributions, i.e., their time–frequency distributions mainly assemble in the lower frequency parts.

2.3. Time–frequency characteristics of OD flows

Likewise, according to (2) and (3), for each of N OD flows consisting of traffic matrix in (1), one can calculate its S-transform in certain time window. Analyzing OD flows in the Abilene and GÉANT, we find that each OD flow in each network belongs to one of two types of time–frequency distributions. Fig. 2 plots these two main types of time–frequency distribution of OD flows from the Abilene and GÉANT, respectively. Fig. 2(a) and (b) denote that OD flows in the Abilene own the obvious time–frequency characteristics in the lower frequency, whereas their time–frequency characteristics in the higher frequency have a relative small value. Fig. 2(c) and (d) formulate that OD flows in the GÉANT also hold the same properties.

Additionally, analyzing the eigenvalue of covariance matrix about time–frequency parameters of link loads and OD flows, we find that only a few eigenvalues are not equal to zero and others are all zero. This implies that link loads and

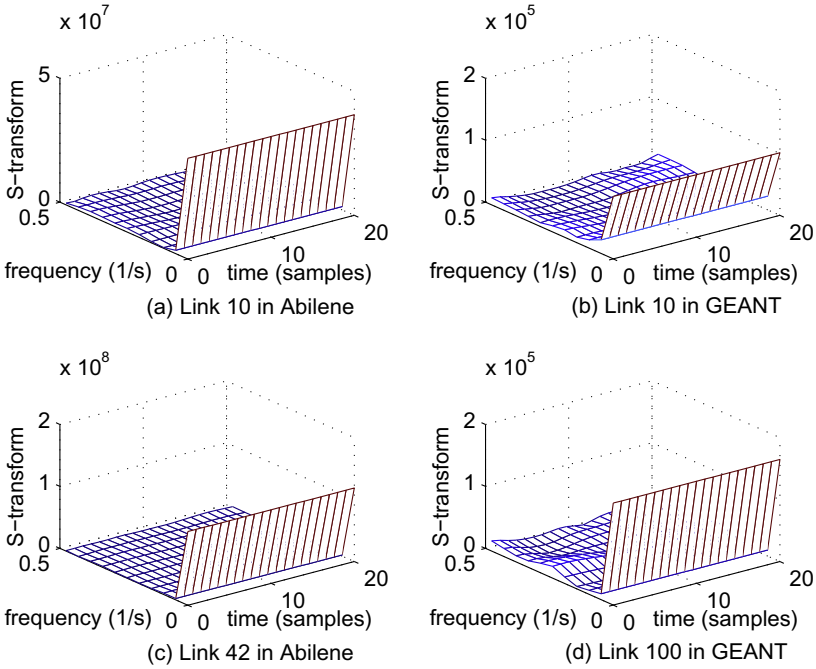


Fig. 1. Time–frequency distribution of link loads.

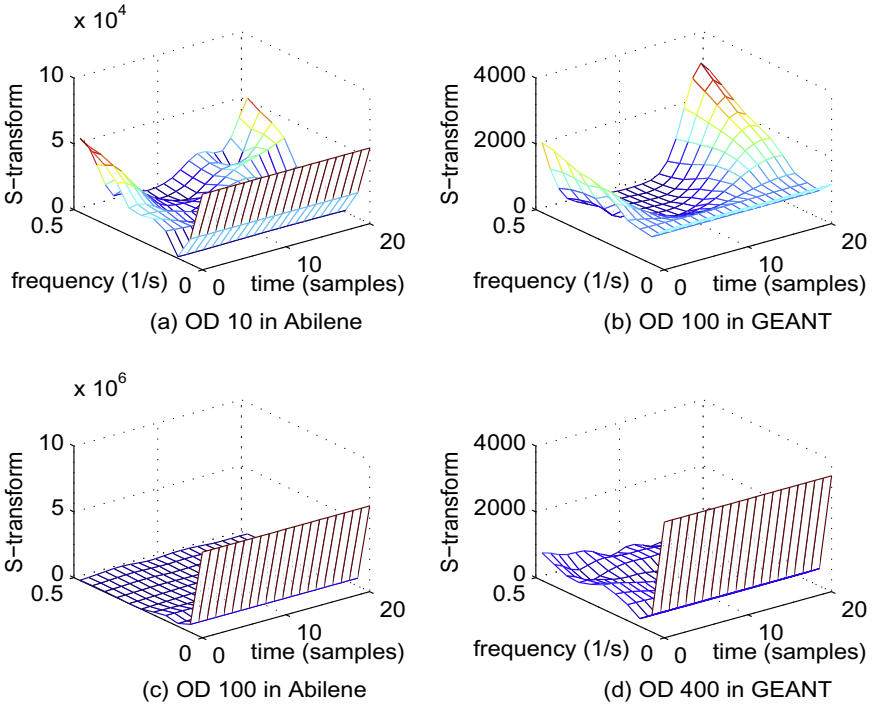


Fig. 2. Time–frequency characteristics of OD flows.

OD flows have the more intensive sparsity in the time–frequency domain than in the time domain. One can find that the time–frequency distributions of OD flows are similar to those of link loads. It is not too difficult to find this cause. In terms of (1) and the linear properties of S-transform [23], time–frequency distributions between link loads and OD flows necessarily exhibit certain constraint relations, which we will discuss in detail in the following section.

3. Model and algorithm

In this section, we outline our model for the analysis of traffic matrix in joint time–frequency domain. Most previous studies about traffic matrix estimations are performed only in the time domain. Due to the ill-posed nature of traffic matrix estimation problem, in particular time-varying properties that traffic matrix holds, it is extremely intractable to accomplish the accurate estimation of traffic matrix in a network. We aim to avoid the difficulties encountered in the time domain and analyze simultaneously its characteristics in both time and frequency domains. As discussed in signal processing, time–frequency analysis considers those modalities that signals own not only in the time domain but in the frequency domain. Here we exploit S-transform [23] proposed by Stockwellc to formulate time and frequency characteristics of traffic matrix in large-scale IP backbone network.

3.1. Time–frequency constraints

As discussed above, because of linear constraints between link loads and OD flows in time domain, their time–frequency distributions are necessary to be satisfied with linear constraints in the time–frequency domain. Without loss of generalization, suppose that S-transform of link loads and OD flows is made in the same time window with its size to be w-point (where w is even number). Then in the w-point time window, link loads are denotes as follows:

$$Y = \begin{bmatrix} y_{11}, y_{12}, \ldots, y_{1w} \\ y_{21}, y_{22}, \ldots, y_{2w} \\ \ldots \\ y_{L1}, y_{L2}, \ldots, y_{Lw} \end{bmatrix}. \tag{4}$$

Likewise, OD flows are written in the same time window into

$$X = \begin{bmatrix} x_{11}, x_{12}, \ldots, x_{1w} \\ x_{21}, x_{22}, \ldots, x_{2w} \\ \ldots \\ x_{N1}, x_{N2}, \ldots, x_{Nw} \end{bmatrix}, \tag{5}$$

where X and Y are still satisfied with constraints $Y = AX$.

Then we perform S-transform for each row in (4) and (5), respectively. For time series $z = (z_1, z_2, \ldots, z_w)$, its S-transform is a $h \times w$ matrix (where h is an integer approximate to $1 + w/2$) as follows:

$$Z_S = \begin{bmatrix} z^S_{11}, z^S_{12}, \ldots, z^S_{1w} \\ z^S_{21}, z^S_{22}, \ldots, z^S_{2w} \\ \ldots \\ z^S_{h1}, z^S_{h2}, \ldots, z^S_{hw} \end{bmatrix}. \tag{6}$$

Z_S is converted into a vector in terms of column, namely

$$Z_S = \left(z^S_{11}, z^S_{12}, \ldots, z^S_{1w}, z^S_{21}, z^S_{22}, \ldots, z^S_{2w}, \ldots, z^S_{h1}, z^S_{h2}, \ldots, z^S_{hw}\right)^T.$$

This equation can be written for short into:

$$Z_S = \left(z^S_1, z^S_2, \ldots, z^S_r\right)^T, \tag{7}$$

where $r = h \times w$.

According to (6) and (7), Y in (4) can be turned by S-transform into:

$$Y_S = \begin{bmatrix} \left(y^S_{11}, y^S_{12}, \ldots, y^S_{1r}\right)^T \\ \left(y^S_{21}, y^S_{22}, \ldots, y^S_{2r}\right)^T \\ \ldots \\ \left(y^S_{L1}, y^S_{L2}, \ldots, y^S_{Lr}\right)^T \end{bmatrix}, \tag{8}$$

where each row in (8) denotes S-transform of each row in (4).

Likewise, X in (5) can be converted by S-transform into:

$$X_S = \begin{bmatrix} \left(x^S_{11}, x^S_{12}, \ldots, x^S_{1r}\right)^T \\ \left(x^S_{21}, x^S_{22}, \ldots, x^S_{2r}\right)^T \\ \ldots \\ \left(x^S_{N1}, x^S_{N2}, \ldots, x^S_{Nr}\right)^T \end{bmatrix}. \tag{9}$$

Proposition 3.1. *According to (1), (8) and (9), Ys and Xs are satisfied with*

$$Y_S = HX_S, \tag{10}$$

where H is derived from A in (1).

The proof can be found in Appendix A. This indicates that link loads and OD flows still meet with linear constraints in the time–frequency domain.

3.2. Time–frequency domain model

Now we formulate time–frequency model used for traffic matrix estimation. In terms of the above analysis, assume the measured sample data about traffic matrix with the size by P. We can slide time window with its size by w from 1 to P to compute S-transform in each time window, which is denoted in Fig. 3.

According to (9), we can obtain $P - w + 1$ values of $X_S$and calculate mean time–frequency parameters in these $P - w + 1$ time windows. As a result, one can attain basic time–frequency model of traffic matrix as follows:

$$X^S_a = \frac{1}{P - w + 1} \sum_{k=1}^{P-w+1} X^k_S. \tag{11}$$

According to (10), for X^S_a, the deviations exist in Y_S and HX^S_a, namely

$$dY = Y_S - HX^S_a. \tag{12}$$

The deviations dY derive from the difference between X^S_a and X_S. Consequently, Eq. (12) can be further converted into:

$$dY = HdY = H\left(X_S - X^S_a\right). \tag{13}$$

To achieve the more accurate time–frequency parameters X_S in a given time window, need to adjust X^S_a to be close to X_S by the updating variable dY, i.e. $Y_S - HX^S_a$. Then we obtain the following time–frequency model:

$$\begin{cases} X^S_a = \frac{1}{P-w+1} \sum_{k=1}^{P-w+1} X^k_S, \\ \widehat{X}^S_a = X^S_a + H^{-1}\left(Y_S - HX^S_a\right). \end{cases} \tag{14}$$

Eq. (14) shows that X^S_a achieved by the sample data need to update by the deviation i$Y_S - HX^S_a$ n order to approximate the accurate time–frequency parameters in certain time window. $\widehat{X}^S_a$ in (14) denotes the time–frequency characteristics in current time window corresponding to Y_S. By inverse S-transform for $\widehat{X}^S_a$, one can attain their corresponding time domain value $\hat{x}_a$:

$$\hat{x}_a = S^{-1}\left(X^S_a + H^{-1}\left(Y_S - HX^S_a\right)\right), \tag{15}$$

where $S^{-1}(\cdot)$ denotes inverse S-transform and H^{-1} indicates the pseudo-inverse matrix computation.

3.3. Dynamic change of OD flows

Generally, traffic matrix often exhibits the obvious tendency and fluctuation. Thus traffic matrices X in (5) can be divided into the below tendency terms x_a and fluctuation terms Δx:

$$X = x_a + \Delta x, \tag{16}$$

where the tendency terms x_a denotes the stationary property of a signal while the fluctuation terms Δx exhibit its non-stationary nature and dynamic change.

Though Eq. (14) can describe main modalities of traffic matrix in the time–frequency domain, namely corresponding to tendency terms x_a, it only denotes the approximate stationary properties about traffic matrix in the time domain. Due to the extremely time-varying properties, it is necessary to further consider its fluctuation in a given time window. Here we use the difference of OD flows between the current moment and the last moment to describe its fluctuations in time. According to each OD flows given in (1), the following new matrix $\phi(t)$ can be obtained:

$$\phi(t) = x(t) - x(t-1). \tag{17}$$

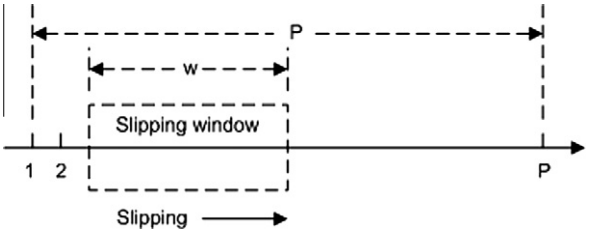


Fig. 3. Computer time–frequency parameters by sliding window.

Likewise, in terms of each link loads given in (1), the second new series $\varphi(t)$ can be attained:

$$\varphi(t) = y(t) - y(t-1). \tag{18}$$

According to (1), (17) and (18), $\phi(t)$ and $\varphi(t)$ meet with the below constrained relations:

$$\varphi(t) = A\phi(t). \tag{19}$$

After performing S-transform for $\phi(t)$ and $\varphi(t)$ in certain w-point time window, respectively, obtain S-transform v and u, respectively. Thus, In terms of Proposition 3.1, Eq. (18) can be turned into:

$$u = Hv. \tag{20}$$

Because the size of columns about matrix H is much larger than the size of its rows, it is very difficult to infer the variable v from u and H. As discussed above, major of u and v own the lower value, so they also own obvious sparsity. To estimate accurately the variable v, assume

$$v = v_s + v_r, \tag{21}$$

where v_s and v_r, respectively denote sparse parts and non-sparse parts in the variable v. Moreover, v_s represents main parts in the variable v. As a result, the Eq. (20) can be approximated as follows:

$$u \approx Hv_s. \tag{22}$$

Obviously, H is only related with A, but it is not related with v_s. According to compressive sensing theory [24], v_s can be reconstructed accurately by the following optimal process:

$$\begin{cases} \min \|v_s\|_1, \\ s.t.\ u \approx Hv_s. \end{cases} \tag{23}$$

Consequently, under theory framework of compressive sensing, we can find the accurate estimation result $\hat{v}_s$ in accordance with (23). Then in terms of (20) and (21), one can easily seek the estimation value $\hat{v}_r$ of the unknown variable v_r, namely

$$\hat{v}_r = H^{-1}(u - H\hat{v}_s), \tag{24}$$

where H^{-1} indicates the pseudo-inverse matrix compu tation.

Hence, in the light of (21), one can find the estimation $\hat{v}$ about the variable v, namely

$$\hat{v} = \hat{v}_s + \hat{v}_r. \tag{25}$$

By inverse S-transform for $\hat{v}$, one can found the corresponding time value $\Delta\hat{x}$

$$\Delta\hat{x} = S^{-1}(\hat{v}) = S^{-1}(\hat{v}_s + \hat{v}_r). \tag{26}$$

According to (15), (16) and (26), inverse S-transform about traffic matrix in the current time window can be found as follows:

$$\widehat{X} = \hat{x}_a + \Delta\hat{x}. \tag{27}$$

Because traffic matrix is non-negative and is satisfied with $Y = AX$, combining with (27), we constructer the following estimation equation about it:

$$\begin{cases} \widehat{X} = \hat{x}_a + \Delta\hat{x}, \\ s.t.\ \hat{x}_{ij} \geqslant 0, \quad i = 1,2,\ldots,N;\ j = 1,2,\ldots,w, \\ Y = A\widehat{X}. \end{cases} \tag{28}$$

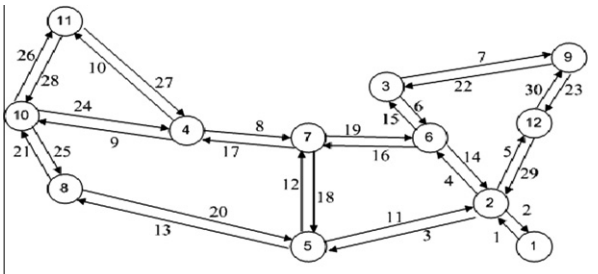
3.4. Estimation algorithm

According to Eqs. (15), (26) and (28), it is not too difficult to obtain traffic matrix's estimations. In the light of the above discussion, our approach combines the time–frequency analysis and sparse properties about traffic matrix to estimate them. Hence, we refer to it as time–frequency sparse estimation (TFSE) algorithm. TFSE algorithm is proposed as follows:

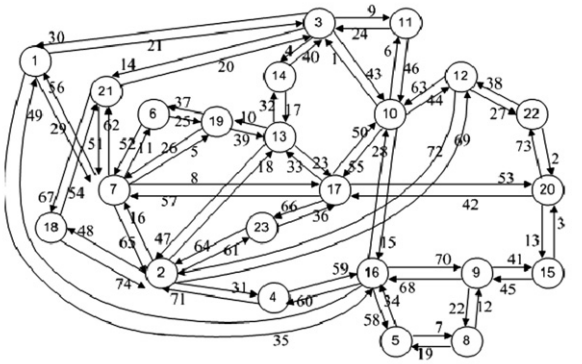
Step 1. Give the size w of time window.
Step 2. According to (3) and (11), calculate time–frequency parameters X_a^S about traffic matrix in the case that the P-point measurement results of traffic matrix is obtained by measuring directly OD flows.
Step 3. In the given time window, in terms of (8), calculate Y_S; at the same time, adjust X_a^S by (14) to achieve $\widehat{X}_a^S$.
Step 4. By (15), make the inverse S-transform of $\widehat{X}_a^S$ to find the corresponding estimation $\hat{x}_a$ of tendency terms of traffic matrix.
Step 5. In the light of (18), computer the difference $\varphi(t)$ of link loads $y(t)$ in the given time window.
Step 6. According to (2), turn $\varphi(t)$ into u by S-transform.
Step 7. Based on compressive sensing theory, according to (23), obtain the sparse solution $\hat{v}_s$ satisfied with (20).
Step 8. By (24), attain the non-sparse solution $\hat{v}_r$ satisfied with (20).
Step 9. In terms of (25), computer the resulting solution $\hat{v}$ meeting with (20).
Step 10. Achieve the deviation $\Delta\hat{x}$ of traffic matrix corresponds to $\hat{v}$.
Step 11. According to (28), computer the resulting estimation $\widehat{X}$ of traffic matrix satisfied with those constraints in (28).
Step 12. If all traffic matrices are estimated, then save the estimation results and exit, or go back to Step 3.

4. Simulation results and analysis

We test two real-world networks Abilene [18] and GÉANT [19] shown in Fig. 4. The number in the circle represents the serial number of routers; the number beside the directed line denotes the serial number of the inner links. The Abilene, an Internet2 backbone network, which is backbone network used for the American education and research, includes 12 routers, 30 internal links, and 24 external links [18]. But the GÉANT is a more scale-large backbone network, which is international network for education and research in European, including 23 routers, 74 internal links, and 46 external links [19]. Traffic data from the Abilene are collected in the 5-min sampling interval by Netflow, while traffic data from the GÉANT consist of the



(a) Abilene network.



(b) GÉANT network.

Fig. 4. Network topologies for simulation.

15-min sampled traffic matrix. In these tow backbone networks, we conduct a series of simulations to validate TFSE, evaluating traffic matrix estimation, analyzing estimation errors, and discussing stability of our method. Since TomoGravity (for short TomoG) [6], PCA estimation [15], and SRSVD [17] are reported as the accurate and acceptable methods for traffic matrix estimation, TFSE will be compared with them. We use the 2000-point real data from the Abilene and GÉANT to simulate performance of four algorithms, respectively. To analyze the accuracy and stability of these algorithms, the first 500-point real data from the Abilene and GÉANT are, respectively, used to construct the model used for PCA and the time–frequency model used for TFSE, while the rest of data are, respectively, exploited to test four algorithms. To fairly compare all four methods, we test them on the identical data in the same case. Here we choose the first 500-point measured data to model based on the following cause: (1) In Abilene and GÉANT, sampling intervals are 5 min and 15 min separately, so there are 288 samples and 96 samples during one day period for them, respectively. Since traffic matrix holds the significantly temporal correlation as mentioned in [4,6,15], at least 288 time slots and 96 time slots are required to accurately capture the characteristics of traffic matrix. Testing shows that 500-point data is suited for both networks. (2) Of course, the more data points should be able to improve the model performance, while the more data need the more measurement and computation overheads. Moreover, simulation also indicates that over 500-point data do not only have no obvious improvement but also result in the significant computation overheads.

4.1. Validation of time–frequency model

To validate reasonableness of the time–frequency model depicted in (14) and (15), we accomplish this by analyzing whether our model can capture the time–frequency characteristics of traffic data in the Abilene as time window is sliding forward. Fig. 5 plots the curves of OD flows in Abilene depicted by time–frequency model.

Fig. 5 shows that our time–frequency model cannot only describe the change tendency of OD flows such as period properties in OD 64 and OD 126, but can also capture the burst or dramatic changes, for instance burst traffic of OD flows in Fig. 5. More importantly, we only use a small parts of traffic data, for instance 500-point, to learn for constructing time–frequency model, whereas this built model can capture the longer change tendency such as 1500-point. That is to say, from Fig. 5, we can see that the estimations from 501 to 2000 time slots are still very accurate. This shows that time–frequency analysis is suited for solving traffic matrix estimation problem.

To validate further our time–frequency model, we analyze joint time–frequency distributions of OD flows denoted in Fig. 5, obtained by our time–frequency model. According to (2), we make S-transform of OD flows above. Fig. 6 exhibits their time–frequency distributions with the size of time window by 20-point, where the top is 2D time–frequency map and the bottom is 3D. Though OD flows in Fig. 5 are attained through time–frequency model, Fig. 6 indicates that their time–frequency distributions are mainly aggregated in the lower frequency parts and the intensity in the lower frequency parts are by nearly 10 orders of magnitude larger than other frequency components. This is consistent to our discussion in Section 2. In turn, this further confirms that our model is indeed able to describe fairly accurately the time–frequency characteristics of OD flows in backbone networks. Similarly, Figs. 7 and 8 plot the curves of OD flows in GÉANT depicted by time–frequency model and their time–frequency distributions, respectively. In contrast to OD flows in Abilene, those in GÉANT hold similar time–frequency distribution, but the value of their time–frequency transforms is different from each other.

4.2. Comparative analysis of traffic matrix estimation

In this subsection, we will analyze comparatively traffic matrix estimation in the Abilene and GÉANT with different algorithms. Here we will regard these algorithms as baseline to be compared. Figs. 9 and 10, respectively, denote traffic matrix estimations in the Abilene and GÉANT with these four algorithms. The first 500-point traffic data are used for PCA and TFSE to learn in order to build the accurate traffic model, while the other 1500-point traffic data are employed to conduct comparative analysis of different algorithms. Figs. 9 and 10 denote several kinds of typical properties of OD flows in the Abilene and GÉANT, for instance extremely time-varying or dynamic changes, burst or dramatic evolution in time, stationary nature such as period variety, and so forth. Fig. 9 shows in the Abilene that though the rate of traffic is up to 10e7 orders of magnitude, all four algorithms can capture traffic change tendency in

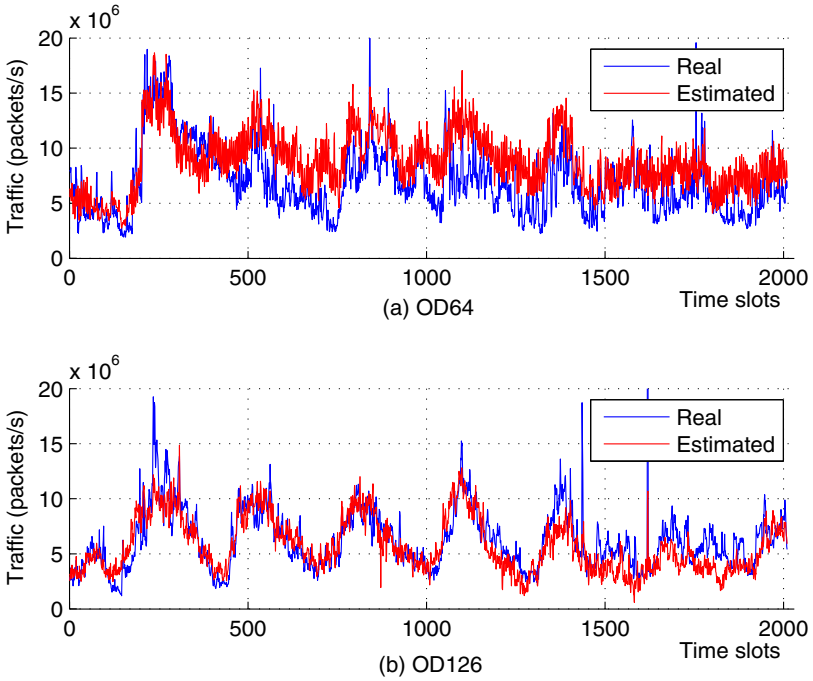


Fig. 5. OD flows in Abilene depicted by time–frequency model.

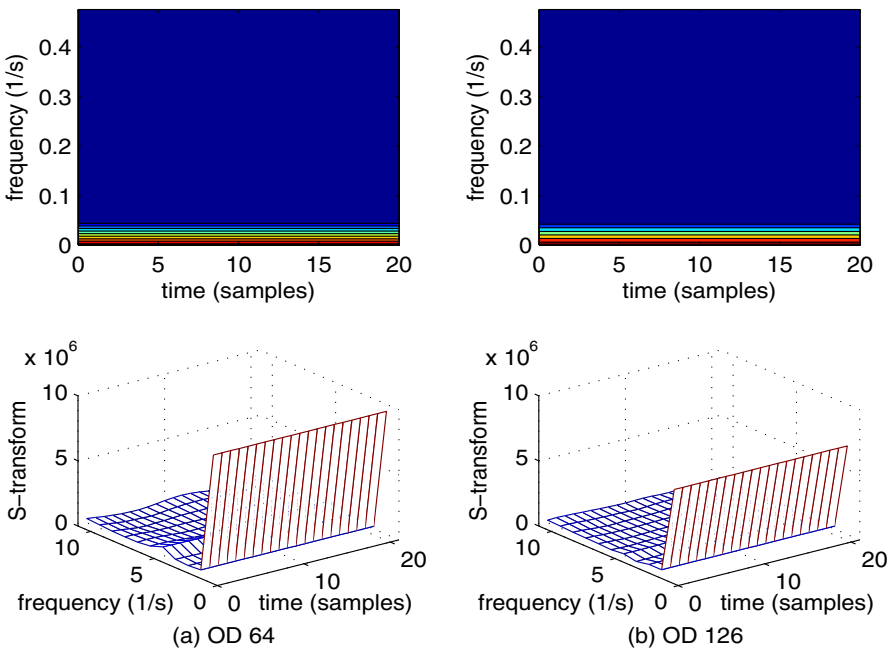


Fig. 6. Time–frequency distributions of estimated OD flows in Fig. 5.

time. Comparatively speaking, TFSE can strictly evolve with real traffic. PCA can also track these dynamic changes, but fluctuate largely near real traffic. TomG and SRSVD cannot keep up with change intensity of traffic and consequently take place the under-estimation. Similarly, Fig. 10 indicates that, in the GÉANT, TFSE can still strictly captures the time-varying nature of OD flows and come up with their change tendency. The other three algorithms take place either over-estimation or under-estimation. Thereby, in contrast to the other three algorithms, TFSE can perform the more accurate estimation for traffic matrix. This shows that new idea that traffic matrix is ana-

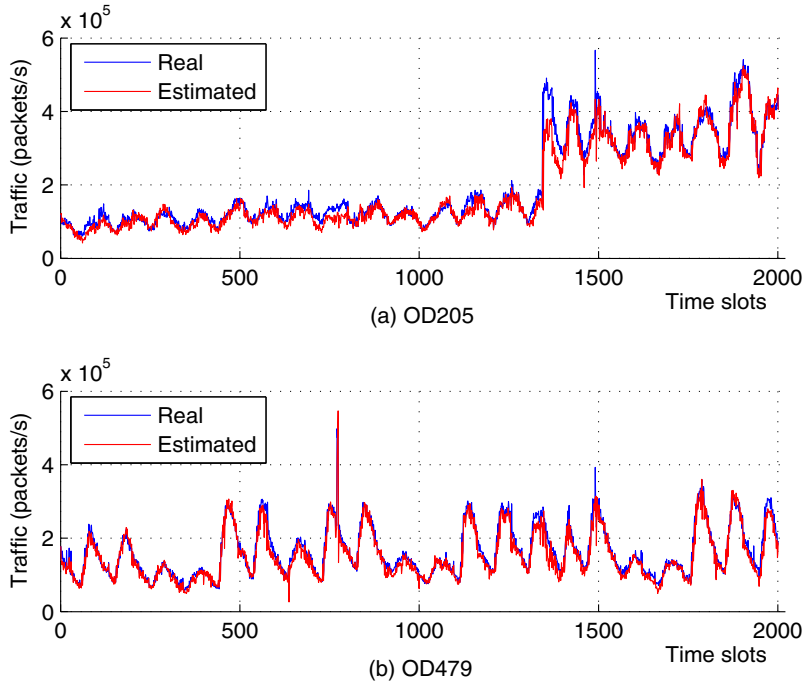


Fig. 7. OD flows in GÉANT depicted by time–frequency model.

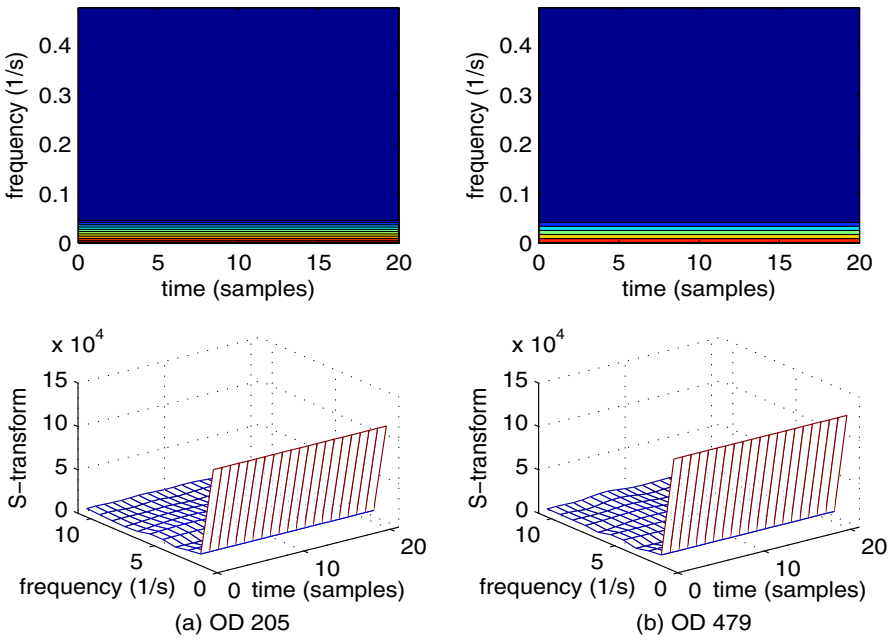


Fig. 8. Time–frequency distributions of estimated OD flows in Fig. 7.

lyzed and estimated in the time–frequency domain is reasonable and practical.

Alternatively, from Figs. 9 and 10, we can see that both TFSE and TomG can more stably estimate OD flows compared with other two algorithms. SRSVD is more stable than PCA. This implies that our method and TomG can have the stronger ability to capture inherent nature of traffic matrix, whereas the larger fluctuation happens to PCA and SRSFM. This further suggests that our method can obtain the more accurate estimation of traffic matrix.

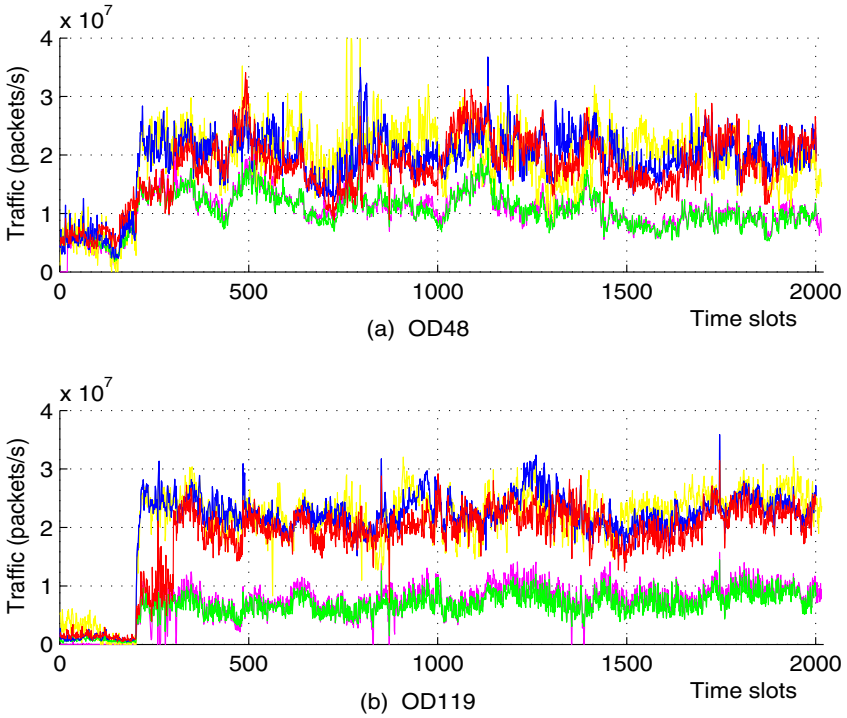


Fig. 9. Traffic matrix estimations in Abilene, with real in blue, TFSE in red, TomoG in green, PCA in yellow, SRSVD in pink. (For interpretation of the references in colour in this figure legend, the reader is referred to the web version of this article.)

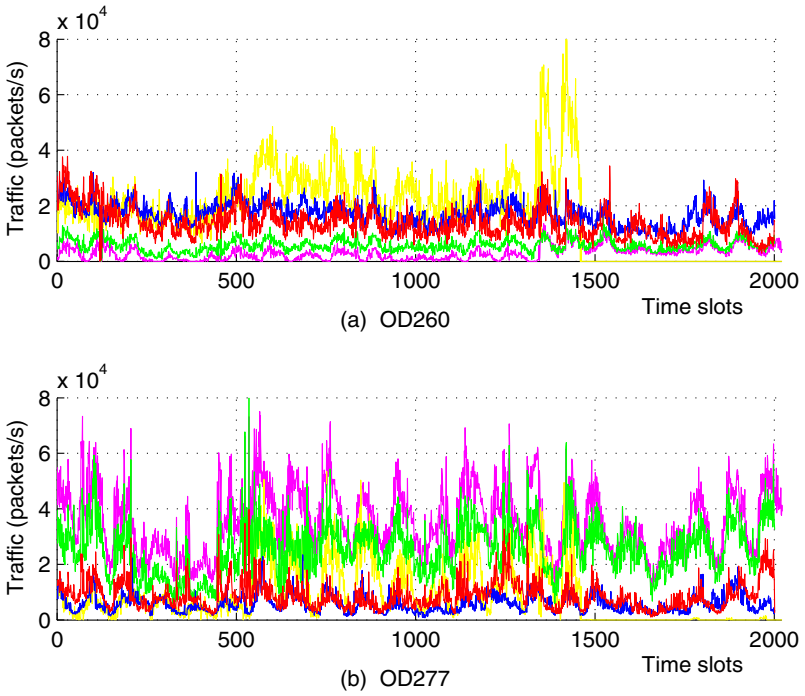


Fig. 10. Traffic matrix estimations in GÉANT, with real in blue, TFSE in red, TomoG in green, PCA in yellow, SRSVD in pink. (For interpretation of the references in colour in this figure legend, the reader is referred to the web version of this article.)

4.3. Stability and estimation accuracy

For a estimation method of traffic matrix, its stability denotes the case that its estimation errors have no too large variation over time. And thereby stability is an important metric describing the estimation method performance. To evaluate the stability and the estimation accuracy of different algorithms, we exploit the spatial relative errors (SREs) and the temporal relative errors (TREs) to make the comparative analysis.

As mentioned in [4], SREs are denoted as follows:

$$\begin{cases} er_{sp}(n) = \frac{\|\hat{x}_T(n) - x_T(n)\|_2}{\|x_T(n)\|_2}, \\ n = 1, 2, \ldots, N. \end{cases} \tag{29}$$

Likewise, TREs are denoted into [4]:

$$\begin{cases} er_{tm}(t) = \frac{\|\hat{x}_N(t) - x_N(t)\|_2}{\|x_N(t)\|_2}, \\ t = 1, 2, \ldots, T, \end{cases} \tag{30}$$

where N and T are the total number of OD flows and measurement moments, respectively; $|\cdot\|_2$ is L_2 norm; $er_{sp}(n)$ and $er_{tm}(t)$ denote the SREs and TREs, respectively. SREs implies the spatial correlations of estimation errors among different OD flows, hence SREs embodies the distributions of estimation errors in the spatial domain. However, TREs reflects the temporal correlations of estimation errors in different moment and describes the distributions of estimation errors in the temporal domain. Thereby, TREs denotes the stability of different algorithm in time. To precisely evaluate the estimation performance of different algorithms, we examine the cumulative distribution functions (CDFs) of their SREs and TREs.

Figs. 11–14 show SREs and TREs, as well as their CDFs, with different algorithm in the Abilene and GÉANT, respectively. For the Abilene, we can see from Fig. 11(a) that the mean SREs of TFSE, TomoG, PCA, and SRSVD in the Abilene are 0.50, 0.68, 0.59, and 0.78, respectively. Hence, TFSE's SREs are the lowest in four algorithms, the second is PCA, and then next is TomoG and SRSVD in turn. Fig. 11(b) exhibits that the mean TREs of TFSE, TomoG, PCA, and SRSVD are 0.16, 0.26, 0.27, and 0.29, respectively. Thus, TFSE's TREs are the lowest in four algorithms, the second is TomoG, and then next is PCA and SRSVD in turn. In Fig. 12(a), we find that, for TFSE, TomoG, PCA, and SRSVD, about 81%, 73%, 63%, and 60% of OD flows arrive at SREs = 0.67. similarly, Fig. 12(b) shows that for about 80% of measurement moment, their TREs are 0.18, 0.30, 0.32, and 0.31 in turn. Thereby, TFSE owns the lowest estimation errors in the Abilene in contrast to the other three algorithms.

Likewise, for the GÉANT, Fig. 13(a) indicates that the mean SREs of TFSE, TomoG, PCA, and SRSVD are 8.92, 25.0, 2.77, and 15.7, respectively. Hence, PCA's SREs are the lowest in four algorithms, the second is TFSE, and then next is SRSVD and TomoG in turn. Fig. 13(b) denotes that the mean TREs of TFSE, TomoG, PCA, and SRSVD are 0.12, 0.50, 0.64, and 0.48, respectively. Thus, TFSE's TREs are the lowest in four algorithms, the second is SRSVD, and then next is TomoG and PCA in turn. From Fig. 14(a), we find that, for about 80% of OD flows, SREs of four algorithms arrive at 0.44, 0.26, 0.16, and 0.30. Fig. 14(c) suggests that for about 70% of measurement moment, their TREs are 0.12, 0.52, 0.54, and 0.51 in turn. Thereby, TFSE also has the lowest estimation errors in the GÉANT compared with the other three algorithms.

To evaluate the stability of four algorithms, we revisit their TREs in the Abilene and GÉANT. Fig. 11(b) indicates that, in the Abilene, four algorithms are relative stable in time. However, TomoG is the most stable, next is TFSE, the third is SRSVD, and the last is PCA. In the GÉANT, Fig. 13(b) represents that TomoG has the same stability as SRSVD, but TFSE has the worse stability and PCA's stability is better.

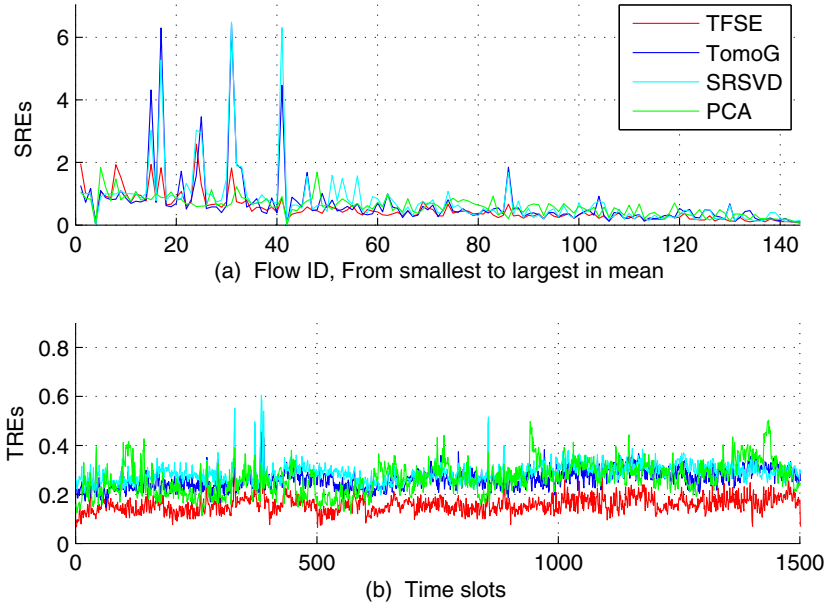


Fig. 11. Relative errors in Abilene.

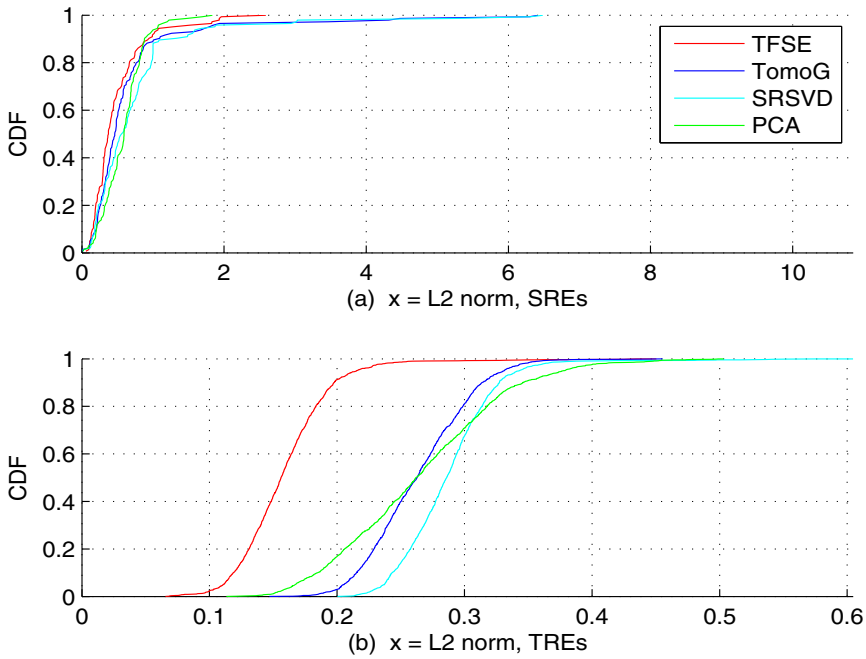


Fig. 12. CDFs of relative errors in Abilene.

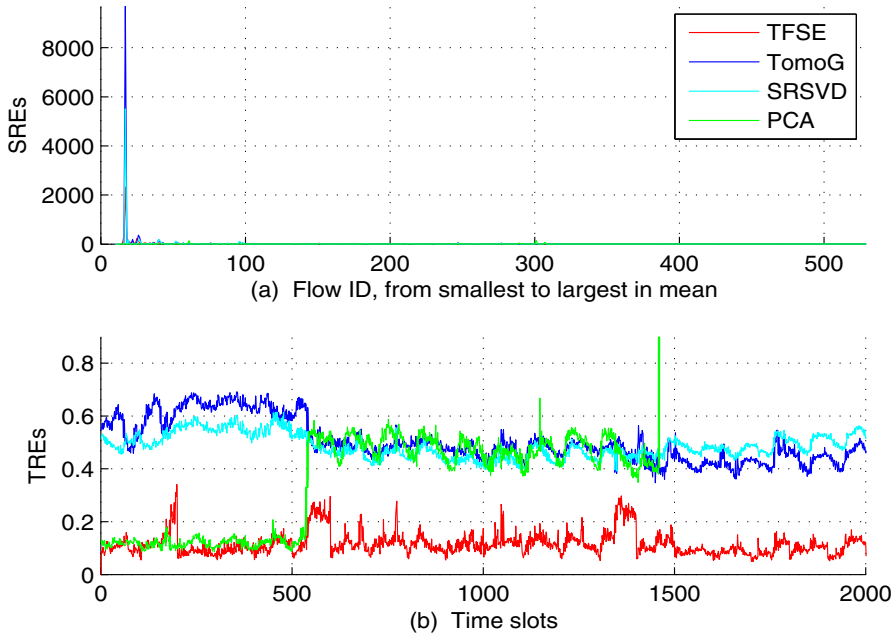


Fig. 13. Relative errors in GÉANT.

4.4. Evaluation of performance improvement

To analyze further the performance of different algorithms, we will discuss their performance improvement. Performance improvement can be defined into:

$$r = \frac{\sum |\hat{x}_A(t) - x(t)| - \sum |\hat{x}_B(t) - x(t)|}{\sum |\hat{x}_A(t) - x(t)|}, \tag{31}$$

where $\sum |\hat{x}_A(t) - x(t)|$ denotes the absolute estimation errors of algorithm A. Fig. 15 shows that, in the Abilene, the average performance improvement for TFSE over TomoG is nearly the same as over PCA, while that over SRSVD is largest up to 44.1%. However, in the GÉANT, TFSE over TomoG has the same average performance improvement as over SRSVD, which falls into the interval from 0.68 to

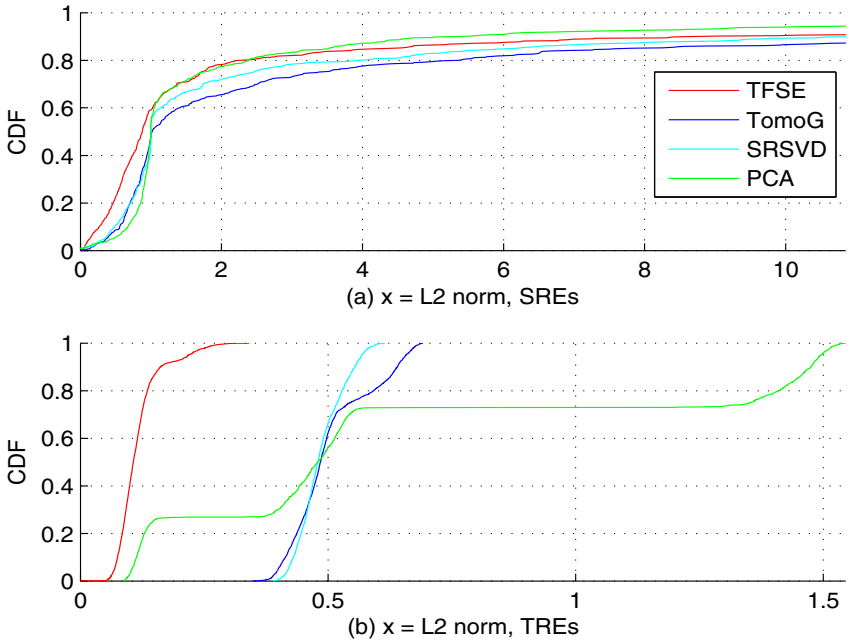


Fig. 14. CDFs of relative errors in GÉANT.

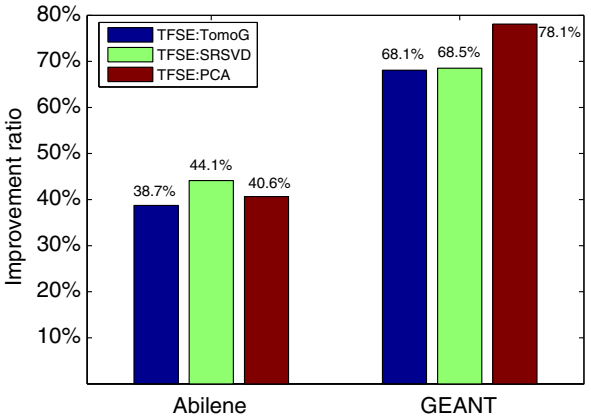


Fig. 15. Average performance improvement.

0.69. At the same time, it over PCA owns the average performance improvement up to 78.1%. To sum up, the improvement of TFSE is extremely obvious.

Of course, although TFSE can obtain the better performance improvement in contrast to other methods, it needs to carry out the direct measurement of traffic matrix and additional computation to build the right model in order to make the more accurate estimation. That is to say, Compared to TomoG, PCA, and SRSVD, TFSE obtains the lower estimation errors at the expense of the measurement and computation overheads. However, this expense relative to the significant performance improvement is worth to make.

5. Conclusion

In this paper, we investigated traffic matrix estimation in large-scale backbone networks from the time–frequency domain. We found in new transform domain that traffic matrix has the extremely obvious time–frequency characteristics mostly in the lower frequency parts. Moreover, analyzing the time–frequency distributions of link loads and OD flows in two real backbone networks, we also saw that their time–frequency distributions are largely similar to each other. Strictly speaking, they exhibit the intensive sparsity in the time–frequency domain. Hence, we naturally used the compressive sensing theory to reconstruct network traffic. Besides, based on the time–frequency properties of network traffic, we proposed a time–frequency model describing the time–frequency nature of network traffic. Then we presented a time–frequency sparse estimation algorithm about traffic matrix. Finally, simulation results in two real backbone networks show that the accuracy, stability, and effectiveness of our approach.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No. 61071124), the Specialized Research Fund for the Doctoral Program of Higher Education (No. 20100042120035), the Open Project of State Key Laboratory of Networking and Switching Technology (No. SKLNST-2009-1-04), and the Fundamental Research Funds for the Central Universities (No. N090404014). The authors thank the reviewers for their helpful comments.

Appendix A

A.1. Proof of Proposition 3.1

Proof. According to (1), (4), and (5), there exists

$$Y = AX. \tag{32}$$

Namely

$$\begin{bmatrix} y_{11}, y_{12}, \ldots, y_{1w} \\ y_{21}, y_{22}, \ldots, y_{2w} \\ \ldots \\ y_{L1}, y_{L2}, \ldots, y_{Lw} \end{bmatrix} = A \begin{bmatrix} x_{11}, x_{12}, \ldots, x_{1w} \\ x_{21}, x_{22}, \ldots, x_{2w} \\ \ldots \\ x_{N1}, x_{N2}, \ldots, x_{Nw} \end{bmatrix}. \tag{33}$$

Consequently, Eq. (33) can be converted into:

$$\begin{bmatrix} y_1 \\ y_2 \\ \ldots \\ y_L \end{bmatrix} = A \begin{bmatrix} x_1 \\ x_2 \\ \ldots \\ x_N \end{bmatrix}, \tag{34}$$

where

$$\begin{cases} y_1 = \{y_{11}, y_{12}, \ldots, y_{1w}\}, \\ y_2 = \{y_{21}, y_{22}, \ldots, y_{2w}\}, \\ \ldots \\ y_L = \{y_{L1}, y_{L2}, \ldots, y_{Lw}\} \end{cases} \tag{35}$$

and

$$\begin{cases} x_1 = x_{11}, x_{12}, \ldots, x_{1w}, \\ x_2 = x_{21}, x_{22}, \ldots, x_{2w}, \\ \ldots \\ x_N = x_{N1}, x_{N2}, \ldots, x_{Nw}, \end{cases} \tag{36}$$

$$A = \begin{bmatrix} a_{11}, a_{12}, \ldots, a_{1N} \\ a_{21}, a_{22}, \ldots, a_{2N} \\ \ldots \\ a_{L1}, a_{L2}, \ldots, a_{LN} \end{bmatrix}. \tag{37}$$

According to (34), then

$$\begin{cases} y_1 = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1N}x_N, \\ y_2 = a_{21}x_1 + a_{22}x_2 + \cdots + a_{2N}x_N, \\ \ldots \\ y_L = a_{L1}x_1 + a_{L2}x_2 + \cdots + a_{LN}x_N. \end{cases} \tag{38}$$

From (2), one can know that S-transform follows linear properties. Hence, we can obtain

$$\begin{cases} Y_1^S = a_{11}X_1^S + a_{12}X_2^S + \cdots + a_{1N}X_N^S, \\ Y_2^S = a_{21}X_1^S + a_{22}X_2^S + \cdots + a_{2N}X_N^S, \\ \ldots \\ Y_L^S = a_{L1}X_1^S + a_{L2}X_2^S + \cdots + a_{LN}X_N^S, \end{cases} \tag{39}$$

where $Y_1^S, Y_2^S, \ldots, Y_L^S$ are the S-transform of time series $y_1, y_2, \ldots, y_L$, respectively; $X_1^S, X_2^S, \ldots, X_N^S$ are the S-transform of time series $x_1, x_2, \ldots, x_N$, respectively.

According to (39), change $Y_1^S, Y_2^S, \ldots, Y_L^S, X_1^S, X_2^S, \ldots, X_N^S$ from matrices to column vecters $y_1^S, y_2^S, \ldots, y_L^S, x_1^S, x_2^S, \ldots, x_N^S$, respectively. Then Eq. (39) can be turned into:

$$\begin{cases} y_1^S = H_{11}x_1^S + H_{12}x_2^S + \cdots + H_{1N}x_N^S, \\ y_2^S = H_{21}x_1^S + H_{22}x_2^S + \cdots + H_{2N}x_N^S, \\ \ldots \\ y_L^S = H_{L1}x_1^S + H_{L2}x_2^S + \cdots + H_{LN}x_N^S, \end{cases} \tag{40}$$

where $H_{ij} = I_r A_{ij}$ and $i = 1, 2, \ldots, L$; $j = 1, 2, \ldots, N$, and I_r is the $r \times r$ unit matrix and $r = h \times w$.

According to (40), then

$$H = \begin{bmatrix} H_{11}, H_{12}, \ldots, H_{1N} \\ H_{21}, H_{22}, \ldots, H_{2N} \\ \ldots \\ H_{L1}, H_{L2}, \ldots, H_{LN} \end{bmatrix}. \tag{41}$$

The proof is concluded. □

References

[1] D. Alderson, H. Chang, M. Roughan, et al., The many facets of Internet topology and traffic, Networks and Heterogeneous Media 1 (4) (2006) 569–600.

[2] Xuetao Wei, Lei Guo, Xingwei Wang, et al., Availability guarantee in survivable WDM mesh networks: a time perspective, Information Science 178 (11) (2008) 2406–2415.

[3] A. Soule, F. Silveira, H. Ringberg, et al., Challenging the supremacy of traffic matrices in anomaly detection, in: Proc. IMC 2007, 2007, pp. 1–6.

[4] A. Soule, A. Lakhina, N. Taft, et al., Traffic matrices: Balancing measurements, inference and modeling, in: Proc. SIGMETRICS 2005, Banff, Canada, June 2005.

[5] D. Jiang, G. Hu, GARCH model-based large-scale IP traffic matrix estimation, IEEE Communications Letters 13 (1) (2009) 52–54.

[6] Y. Zhang, M. Roughan, N. Duffield, et al., Fast accurate computation of large-scale IP traffic matrices from link loads, ACM SIGMETRICS Performance Evaluation Review 31 (3) (2003) 206–217.

[7] A. Gunnar, M. Johansson, T. Telkamp, Traffic matrix estimation on a large IP backbone: A comparison on real data, in: Proc. IMC 2004, Taormina, Italy, 2004.

[8] V. Erramilli, M. Crovella, N. Taft, An independent-connection model for traffic matrices, in: Proc. IMC 2006.

[9] I. Juva, Sensitivity of traffic matrix estimation techniques to their underlying assumption, in: Proc. ICC 2007, Glasgow, Scotland, 2007.

[10] J. Ni, S. Tatikonda, E.M. Yeh, A Large-Scale Distributed Traffic Matrix Estimation Algorithm, in: Proc. Globecom 2006, 2006, pp. 1–5.

[11] P. Casas, S. Vaton, L. Fillatre, et al., Efficient methods for traffic matrix modeling and on-line estimation in large-scale IP networks, in: Proc. ITC21 2009, 2009, pp. 1–8.

[12] S. Stoev, G. Michailidis, J. Vaughan, Global modeling of backbone network traffic, in: Proc. Globalcom 2010, 2010, pp. 1–5.

[13] M. Roughan, Simplifying the synthesis of Internet traffic matrices, SIGCOMM Computer Communications Review 35 (5) (2005) 93–96.

[14] K.V. Vishwanath, A. Vahdat, Swing: Realistic and responsive network traffic generation, IEEE/ACM Transactions on Networking 17 (3) (2009) 712–725.

[15] A. Lakhina, K. Papagiannaki, M. Crovella, et al., Structural analysis of network traffic flows, in: Proc. SIGMETRICS 2004, 2004, pp. 61–72.

[16] D.L. Donoho, Compressed sensing, IEEE Transactions on Information Theory 52 (4) (2006) 1289–1306.

[17] Y. Zhang, M. Roughan, W. Willinger, et al., Spatio-temporal compressive sensing and Internet traffic matrices, in: Proc. SIGCOMM 2009, Barcelona, Spain, 2009, pp. 267–278.

[18] <http://www.cs.utexas.edu/yzhang/research/Abilene-TM/>.

[19] S. Uhlig, B. Quoitin, S. Balon, et al., Providing public intradomain traffic matrices to the research community, ACM SIGCOMM Computer Communication Review 36 (1) (2006) 83–86.

[20] D. Jiang, X. Wang, L. Guo, An optimization method of large-scale IP traffic matrix estimation, AEU-International Journal of Electronics and Communications 64 (7) (2010) 685–689.

[21] D. Jiang, X. Wang, L. Guo, Mahalanobis distance-based traffic matrix estimation, European Transactions on Telecommunications 21 (3) (2010) 195–201.

[22] D. Jiang, Y. Han, Z. Xu, et al., A time-frequency detecting method for network traffic anomalies. In: Proceedings of the International Conference on Computational Problem-Solving (ICCP'10), Lijiang, Yunnan, Dec. 3–5, 2010, pp. 94–97.
[23] R.G. Stockwell, Why use the S-transform? Fields Institute Communications Series, vol. 52, American Mathematical Society, 2007. pp. 279-309.
[24] E. Cands, J. Romberg, T. Tao, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information, IEEE Transactions on Information Theory 52 (2) (2006) 489–509.

Dingde Jiang received the Ph.D. degree in communication and information systems from School of Communication and Information Engineering, University of Electronic Science and Technology of China, Chengdu, China, in 2009. He is currently an Associate Professor in College of Information Science and Engineering, Northeastern University, Shenyang, China. His research interests include network measurement, network security, Internet traffic engineering, and communication networks. Jiang is a member of IEEE and IEICE.

Zhengzheng Xu is currently working toward Ph.D. in management science and engineering in College of Information Science and Engineering, Northeastern University, Shenyang, China. She is currently a Research Member at Key Lab of Comprehensive Automation of Process Industry of Ministry of Education, College of Information Science and Engineering, Northeastern University, Shenyang, China. She is also presently a Research Member at Systems Engineering Research Institute at the same university. Her research interests include supply chain and logistics management, decision analysis, modeling, and optimization.

Zhenhua Chen received BSc in College of Information Science and Engineering, Northeastern University, Shenyang, China, in 2010. He is currently a Master at the same university. His research interests include network measurement and cognitive networks.

Yang Han received BSc in College of Information Science and Engineering, Hainan University, Hainan, China, in 2010. He is currently a Master in Communication and Information System, Northeastern University, China. His research interests include cognitive radio network and cooperation communication.

Hongwei Xu received BSc in College of Information Science and Engineering, Shenyang University, Shenyang, China, in 2010. He is currently a Master in Communication and Information System, Northeastern University, China. His research interests include network measurement and performance analysis.