An Accurate Method to Estimate Traffic Matrices from Link Loads for QoS Provision

Xingwei Wang, Dingde Jiang, Zhengzheng Xu, and Zhenhua Chen

Abstract: **Effective traffic matrix estimation is the basis of efficient traffic engineering, and therefore, quality of service provision support in IP networks. In this study, traffic matrix estimation is investigated in IP networks and an Elman neural network-based traffic matrix inference (ENNTMI) method is proposed. In ENNTMI, the conventional Elman neural network is modified to capture the spatio-temporal correlations and the time-varying property, and certain side information is introduced to help estimate traffic matrix in a network accurately. The regular parameter is further introduced into the optimal equation. Thus, the highly ill-posed nature of traffic matrix estimation is overcome effectively and efficiently.**

Index Terms: **Ill-posed nature, origin-destination flow, quality of service (QoS) provision, traffic engineering, traffic matrix estimation.**

I. INTRODUCTION

Quality of service (QoS) provision is essential in next generation IP network and thus needs a variety of QoS facilities to be deployed in IP network. Traffic engineering as an effective means to provide QoS support in IP network has received much attention in academic and industrial worlds; however, it requires measurement, modeling, characterization, and control of IP network traffic, and thus requires network operators to know how network traffic flows in their networks; this means effective traffic matrix estimation is necessary in a IP network. A traffic matrix describes the amount of traffic that flows between every pair of origin-destination (OD) nodes in a network. It provides valuable information about the current network state to network operators. However, because of the causes mentioned in [1], especially with the size of the IP network growing exponentially, it is significantly difficult and even impossible to measure the traffic matrix directly and accurately. This forces network operators to estimate it. The estimation results of the traffic matrix is so far used by network operators to conduct traffic engineering [2], QoS routing [3], failure management [4]–[7], network dimensioning [8], load balancing [9], [10], network capacity planning [11]–[13], survivability enhancement [14]–[17], optimized network node deployment [18]–[22], and so on. Therefore, this research topic has recently received much attention [23]–[27].

Since the problem of traffic matrix estimation has the highly ill-posed properties [28], [29], this topic is a challenging research problem. Vardi [28] and Tebaldi *et al.* [29] first used the statistical inference techniques to investigate this problem over a local area network (LAN). They modeled OD flows as an independent and identically-distributed (IID) Poisson model. Then the expectation-maximization (EM) algorithm was exploited to obtain the value of traffic matrix estimation. Cao *et al.* [30] studied a more complex case and used the modified EM algorithm to predict the time-varying traffic over a LAN. Zhang *et al.* [24], [25] studied traffic matrix estimation in the IP network. They introduced the gravity model into this problem. Tan *et al.* [26] made the traffic matrix estimation by calculating the {1}-inverse of a routing matrix. Nevertheless, as reported in [27] and [29], the statistical inference techniques are sensitive to prior information, while the gravity model methods still have larger estimation errors, though it partially reduces the sensitivity to prior information. Moreover, new traffic characteristics and new user behaviors will result in new problems for traffic matrix estimation. How to perform effective and accurate estimation of the traffic matrix is a challenge. Hence, there is a need to develop a new method to estimate the traffic matrix.

We study traffic matrix estimation in IP networks and present a novel method called the Elman neural network-based traffic matrix inference (ENNTMI). As mentioned in [1] and [27], the traffic matrix itself not only holds the temporal, spatial, and spatio-temporal correlations, but also exhibits a time-varying property. The accuracy of traffic matrix estimation largely depends on whether we can capture these characteristics. Elman neural network (ENN) [31] is a powerful modeling tool. It is extensively applied to signal processing, pattern recognition, modeling, and so on. The conventional ENN is modified to model the problem of traffic matrix estimation in IP network. After the modified ENN's outputs are introduced back into its inputs, the temporal correlations of the traffic matrix can be captured correctly. At the same time, the link loads of the several measurement moments before current moment are also led into the ENN's inputs. This further ensures that the modified ENN can reflect accurately the temporal properties of traffic matrix. To seize the spatial nature of traffic matrix, we deal with all the OD flows of the measured networks in a parallel way. Moreover, because of the parallel structure of ENN, it can also denote accurately the spatial nature of traffic matrix. Thus, the modified ENN can capture the above characteristics of traffic matrix by

Manuscript received September 07, 2008; approved for publication by Dongman Lee, Division III Editor, November 01, 2010.

This study was supported by the National Natural Science Foundation of China (Nos. 61071124, 61070162, 71071028, 60802023, 70931001), the Specialized Research Fund for the Doctoral Program of Higher Education (Nos. 20100042110025, 20100042120035, 20070145017), 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 (Nos. N090404014, N090504003, N090504006).

X. Wang, Z. Xu, and Z. Chen are with the College of Information Science and Engineering, Northeastern University, Shenyang 110819, China, email: wangxw@mail.neu.edu.cn, chjdd@tom.com, merry_99@sina.com.

D. Jiang is with the College of Information Science and Engineering, Northeastern University, Shenyang 110819, China, and State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China, email: jiangdd99@sina.com.

1229-2370/10/$10.00 © 2010 KICS

training it. Built on the modified ENN model, the regular parameter is introduced into the optimal equation and describes the problem of traffic matrix estimation as an optimal process satisfied with some constraints. By seeking the optimal solution of traffic matrix, ENNTMI can easily get rid of the ill-posed nature of this problem and obtain its fairly accurate estimation. The real data from the Abilene [32], GÉANT [33], and campus networks are used to validate ENNTMI. Simulation results show that in contrast to previous methods, ENNTMI exhibits lower estimation errors and stronger robustness to noise.

This paper is organized as follows. Problem statement is described in the next section. The proposed estimation model for traffic matrix and ENNTMI method are discussed in detail. Section III presents the experimental results and analysis in the Abilene, GÉANT, and campus networks. Concluding remarks are given in the last section.

II. Problem Statement

An IP network, assuming there are n nodes and L links, will have $N = n^2$ OD flows. The traffic matrix and link loads at time t can be denoted as $x(t) = (x_1(t)\ x_2(t) \cdots x_N(t))^T$ and $y(t) = (y_1(t)\ y_2(t)\ \cdots\ y_L(t))^T$, respectively. The traffic matrix $x(t)$ and link loads $y(t)$ are associated by the L by N routing matrix $\mathbf{A} = (A_{ij})_{L\times N}$, where A_{ij} equals 1 if OD flow j traverses link i and zero otherwise. They are satisfied with the following constraints:

$$y(t) = \mathbf{A}x(t). \tag{1}$$

By collecting SNMP measurement data, we can directly compute link loads. The routing matrix can be obtained from the status and configuration information of the network. Therefore, the problem of traffic matrix estimation is that given link loads $y(t)$ and routing matrix $\mathbf{A}$ how can one seek to obtain a required solution $x(t)$ satisfying (1). For an IP network, the number of OD flows is generally much larger than that of links, i.e., $L \ll N$. The linear problem denoted by (1) is highly under-constrained. This shows that the solution satisfied with (1) is not unique. Hence, IP traffic matrix estimation is a highly ill-posed inverse problem. How to overcome the ill-posed nature of this problem is the main challenge faced at present.

A. Estimation Model for Traffic Matrix

Fig. 1(a) shows that the traffic in the Abilene network not only has the period characteristic, but also evolutes along with time. This shows that the traffic in the Abilene network holds temporal correlations and time-varying property. Similarly, Fig. 1(b) shows that the traffic in the GÉANT network also has these characteristics. Compared with Figs. 1(a) and 1(b) shows that the traffic in the GÉANT network changes more quickly over time, i.e., its time-varying property is more obvious. Fig. 1(c) shows that the traffic in the campus network also holds the temporal correlations, spatial correlations, time-varying nature, and so on. Hence, the traffic matrix holds temporal correlations and time-varying properties too. How to capture accurately these characteristics is significantly difficult. Moreover, as mentioned in [1] and [27], traffic matrix also holds the spatial and spatio-temporal correlations. The accuracy of traffic matrix estimation

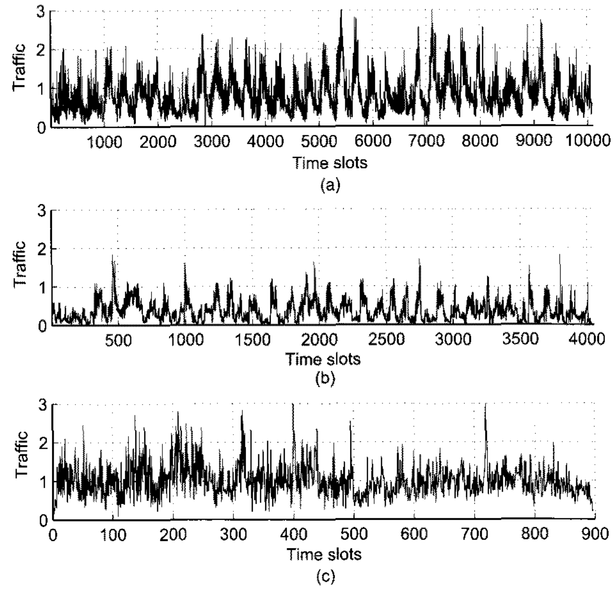


Fig. 1. Traffic in (a) Abilene, (b) GÉANT, and (c) campus networks.

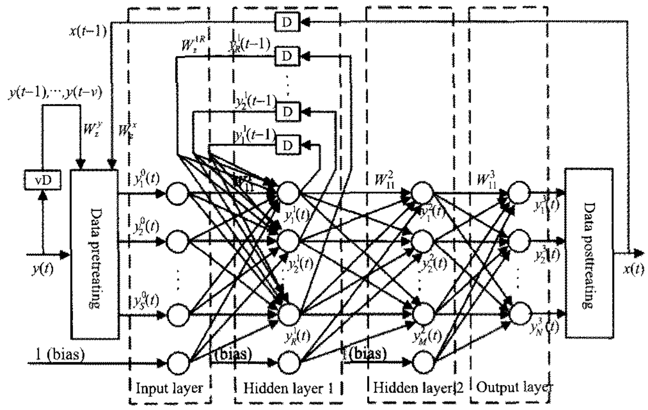


Fig. 2. Schematic diagram of the modified ENN model, with the first hidden layers fully recurrent, the second hidden layers forward, and with an output layer one-step prediction.

largely depends on whether the built model can capture correctly all these characteristics. As a powerful modeling tool, the ENN has the ability of learning and generalizing. Thus, we modify the conventional ENN to capture all these properties so that the highly ill-posed nature of the problem of traffic matrix estimation can be overcome successfully.

Fig. 2 plots the modified ENN architecture used for the problem of traffic matrix estimation, where it holds an input layer, two hidden layers with one fully recurrent layer and one forward layer, and an output layer with one-step linear prediction; link loads $y(t)$ and traffic matrix $x(t)$ are used as the input and output of the modified ENN model, respectively; "D" and "vD" denote the unit delay and v-unit delays, respectively; $y^0(t) = (y_1^0(t)\ y_2^0(t)\ \cdots\ y_S^0(t))^T$, $y^1(t) = (y_1^1(t)\ y_2^1(t)\ \cdots\ y_R^1(t))^T$, $y^2(t) = (y_1^2(t)\ y_2^2(t)\ \cdots\ y_M^2(t))^T$, and $y^3(t) = (y_1^3(t)\ y_2^3(t) \cdots y_N^3(t))^T$ represent the outputs of the first hidden, second hidden, and output layers, respectively; $W^1 = (W_{11}^1\ W_{12}^1 \cdots W_{(S+1)R}^1)^T$, $W^2 = (W_{11}^2\ W_{12}^2 \cdots W_{(R+1)M}^2)^T$, and $W^3 = (W_{11}^3\ W_{12}^3\ \cdots\ W_{(M+1)N}^3)^T$ denote the forward connection weight matrix of the first hidden, second hidden, and output layers, respectively; $W_z = (W_z^{11}\ W_z^{12} \cdots W_z^{RR})^T$, $W_z^x = (W_z^{x1}\ W_z^{x2} \cdots W_z^{xN})^T$, and $W_z^y = (W_z^{y,11}\ W_z^{y,12} \cdots W_z^{y,vL})^T$ denote the feedback connection weight matrix of the first hidden layer, model input, and model output, respectively. On the basis of the conventional ENN, we have proposed a three-layer ENN structure. The recurrent connection of the first hidden layer can sufficiently capture the traffic matrix's temporal correlations, while the ENN's space structure can accurately seize its spatial correlations. Furthermore, by introducing the delayed inputs and outputs into the input of the model, we can further ensure that the modified ENN model can accurately capture the traffic matrix's spatio-temporal correlations and time-varying nature. In addition, from Fig. 2, we can see that the parallel structure of the modified ENN model can make quickly the estimation of the traffic matrix as soon as this model is successfully established.

According to Fig. 2, the following equation is attained.

$$\begin{aligned} y_h^3(t) &= \varphi_h^3(n_h^3(t)), \\ n_h^3(t) &= \sum_{i=1}^{M} W_{hi}^3 y_i^2(t) + b_h^3, \\ y_i^2(t) &= \varphi_i^2(n_i^2(t)), \\ y_i^2(t) &= \sum_{j=1}^{R} W_{ij}^2 y_j^1(t) + b_i^2, \\ y_j^1(t) &= \varphi_j^1(n_j^1(t)), \\ n_j^1(t) &= \sum_{k=1}^{S} W_{jk}^1 y_k^0(t) + \sum_{l=1}^{R} W_z^{jl} y_k^1(t-1) + b_j^1, \\ y_k^0(t) &= f_k(y(t-1), \cdots, y(t-v), y(t), x(t-1)) \end{aligned} \tag{2}$$

where $h = 1, 2, \cdots, N$, φ_s^r and n_s^r $(r = 1, 2, 3)$ is the activation function and output of the sth neural cell in the rth layer, respectively; b_u^r represents the uth bias value in the rth layer. Equation (2) formulates the traditional ENN model. Additionally, the data pre-treating and post-treating processes in Fig. 2, respectively, are denoted as follows:

$$x(t) = \gamma(y^3(t)) \tag{3}$$

and

$$y_k^0(t) = f_k(y(t-1), \cdots, y(t-v), y(t), x(t-1)) \tag{4}$$

where $k = 1, 2, \cdots, S$; $f = (f_1, f_2, \cdots, f_S)$ and γ denote the data pre-treating and post-treating processes, respectively.

According to (2)–(4), we can obtain the below equation:

$$\begin{aligned} x(t) &= \gamma(y^3(t)), \\ y^3(t) &= \varphi^3(W^3 y^2(t) + b^3), \\ y^2(t) &= \varphi^2(W^2 y^1(t) + b^2), \\ y^1(t) &= \varphi^1(W^1 y^0(t) + W_z y^1(t-1) + b^1), \\ y^0(t) &= f(y(t-1), \cdots, y(t-v), y(t), x(t-1)) \end{aligned} \tag{5}$$

where φ^i $(i = 1, 2, 3)$ is the activation function of the ith layer. Equation (5) represents the following mapping relation:

$$x(t) = \phi(y(t)) \tag{6}$$

where ϕ denotes the complex mapping from link loads $y(t)$ to traffic matrix estimation $x(t)$. Moreover, in the post-treating process, we make the ENN's output satisfy two constraints, namely, (1) and $x_i(t) \geq 0$ $(i = 1, 2, \cdots, N)$. Hence, the output of the modified ENN's model is given as follows:

$$\begin{aligned} & x(t) = \phi(y(t)), \\ \text{s.t.} \quad & y(t) = \mathbf{A}x(t), \\ & x_i(t) \geq 0, i = 1, 2, \cdots, N \end{aligned} \tag{7}$$

From (2)–(7), we can see that we are able to obtain the traffic matrix's accurate estimation following the constraints of (1) and $x_i(t) \geq 0$ $(i = 1, 2, \cdots, N)$) as long as the weights in (5) and (6) are determined. By training the modified ENN model denoted by Fig.1 with the input-output data pairs, we can avoid the complex mathematical computation and can easily build the traffic matrix's estimation model. And then according to (7), the traffic matrix's estimation $\hat{x}_e(t) = \phi(y(t))$ that satisfies the above constraints can be attained.

Until now, we have proposed the process of predicting a traffic matrix according to the estimation model denoted in Fig. 1. The following Algorithm 1 presents the complete steps in this process.

Algorithm 1

Step 1. Initialize the network model denoted in Fig. 2. Set the error δ and total iterative steps Z, θ, and $k = 0$.

Step 2. According to (2)–(7), train the above model with the input-output data pairs and get the output of the model.

Step 3. Calculate the gradient of the model with backpropagation algorithm, and update its weights.

Step 4. Compute the total error $\varepsilon = ||\hat{x}(t) - x(t)||_2$ of the output of the model.

Step 5. Compute the estimation error $\alpha = ||y(t) - \mathbf{A}\hat{x}(t)||_2$.

Step 6. If $\varepsilon < \delta$ or $k > Z$ or $\alpha < \theta$, then exit the training process and go back to step 7, or set $k = k + 1$ and go back to step 2.

Step 7. Give the input data to the network model denoted in Fig. 2, and make the data pretreatment process.

Step 8. According to $\hat{x}_e(t) = \phi(y(t))$, get the traffic matrix estimation $\hat{x}_e(t)$.

Step 9. If the estimating process is over, output the estimation result and exit, or go back to step 7.

B. Traffic Matrix Inference

On the basis of the above estimation model, we describe traffic matrix estimation into an optimal process to further overcome the ill-posed nature of this problem. The objective function is given as follows:

$$\begin{aligned} & \min((y(t) - \mathbf{A}x(t))^T(y(t) - \mathbf{A}x(t)) + \\ & \quad \lambda(x(t) - x_0(t))^T \mathbf{C}(x(t) - x_0(t)), \\ \text{s.t.} \quad & y(t) = \mathbf{A}x(t), \\ & x_i(t) \geq 0, \quad i = 1, 2, \cdots, N \end{aligned} \tag{8}$$

where $x(t)$ and $x_0(t)$ denote the traffic matrix and its initial value at time t, respectively, $\mathbf{C}$ represent a smoothing matrix;

and λ represents a regularization parameter, with its value being 0.01 or so.

$$\text{Set} \quad x(t) = x_0(t) + \Delta x(t). \quad (9)$$

According to (8) and (9), obtain the following equation:

$$\begin{aligned} &\min((y(t) - \mathbf{A}x_0(t)) - \mathbf{A}\Delta x(t))^T((y(t) - \\ &\quad \mathbf{A}x_0(t)) - \mathbf{A}\Delta x(t)) + \lambda(\Delta x(t))^T \mathbf{C}\Delta x(t), \\ &\text{s.t.} \quad y(t) = \mathbf{A}x(t), \\ &\qquad x_i(t) \geq 0, \quad i = 1, 2, \cdots, N. \end{aligned} \quad (10)$$

Then the optimal solution of (10) is obtained as follows:

$$\Delta x(t) = (\mathbf{A}^T\mathbf{A} + \lambda\mathbf{C})^{-1}\mathbf{A}^T(y(t) - \mathbf{A}x_0(t)). \quad (11)$$

Hence, according to (11), we construct the following iterative equation:

$$\begin{aligned} x^{v+1}(t) &= x^v(t) + \Delta x^{v+1}(t), \\ \Delta x^{v+1}(t) &= (\mathbf{A}^T\mathbf{A} + \lambda\mathbf{C})^{-1}\mathbf{A}^T(y(t) - \mathbf{A}x^v(t)), \\ x^0(t) &= \hat{x}_e(t) \end{aligned} \quad (12)$$

where v represents the iterative step and $\hat{x}_e(t)$ is the estimation attained by Algorithm 1. Equation (12) is the iterative equation of traffic matrix inference. According to (12), we can attain the traffic matrix's optimal estimation $\hat{x}_g(t)$.

From (8)–(12), we can see that by introducing the regular parameter into the process of traffic matrix inference, we can further get rid of the ill-posed nature of this problem. Moreover, this iterative inference can further decrease the estimation errors and improve the estimation accuracy. We have presented the process of traffic matrix inference. Algorithm 2 gives a description of this process.

Algorithm 2

Step 1. Set the error α and iterative step V. Let $v = 0$ and $x^0(t) = \hat{x}_e(t)$.

Step 2. According to (12), perform traffic matrix inference.

Step 3. If $\|y(t) - \mathbf{A}x^{v+1}(t)\|_2 \leq \alpha$ or $v > V$, exit and output the result, or go back step 2.

C. The Complete Algorithm

The complete ENNTMI method is described as follows.

Step 1. Obtain the initial traffic matrix $\hat{x}_e(t)$ according to Algorithm 1.

Step 2. According to Algorithm 2, perform the traffic matrix inference to attain the optimization solution $\hat{x}_g(t)$.

Step 3. Adjust $\hat{x}_g(t)$ with IPFP and obtain a more accurate estimation $\hat{x}(t)$ satisfied with the constraints in (8).

The creativity of ENNTMI method includes several aspects. First, for the proposed estimation model for traffic matrix, the link loads of the several measurement moments before the current moment are introduced into the inputs of this model. This helps to further capture the temporal correlations of the traffic matrix. Second, because the traffic matrix holds temporal, spatial, and spatio-temporal correlations and time-varying property, and its real value is several orders of magnitude, it is impossible that ENN be directly exploited to handle traffic matrix estimation. The pre-treating and post-treating processes are added before the ENN's inputs and after its outputs, respectively. The pre-treating process handles the link loads and the feedback inputs of traffic matrix in order to be suited for the requirements of ENN's inputs. However, in the post-treating process, the ENN's outputs are handled and transferred to meet the real size of the traffic matrix. On the other hand, the ENN's outputs are also treated exactly so that the outputs of this estimation model are satisfied with some constraints in (7). Third, the feedback outputs are those of this model instead of the ENN's outputs. This help to capture the actual properties that the traffic matrix itself owns. Additionally, to capture the spatial nature of the traffic matrix, all the OD flows of the measured networks are dealt with in a parallel way. Finally, statistical inference is introduced to further overcome the ill-posed nature of the problem of traffic matrix estimation. The smoothing matrix $\mathbf{C}$ in (8) is replaced with the covariance matrix of the sample traffic matrix here. This covariance matrix can denote the spatial and temporal correlations. Thus, (8) can describe the spatio-temporal correlations. Moreover, a regularization parameter λ can turn the under-constrained problem into a well-posed one. According to (12), the ill-posed nature can be overcome further and accurate estimation of traffic matrix can be obtained.

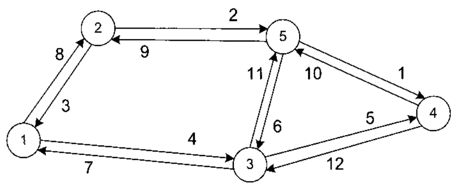


Fig. 3. Campus network topologies for simulation.

III. EXPERIMENTAL RESULTS AND ANALYSIS

We test three real-world networks: Campus network plotted in Fig. 3, and Abilene [32] as well as GÉANT [33] networks shown in Fig. 4, where campus network is one of the university networks in China, while Abilene and GÉANT are two backbone networks used for education and research in America and Europe, respectively. The number in the circle denotes the serial number of routers; the number beside the directed line denotes the serial number of the inner links. A series of simulations are conducted to validate ENNTMI, analyzing traffic matrix tracking, estimation errors (spatial relative errors (SREs) and temporal relative errors (TREs)), and robustness to the noise. Tomo-Gravity [24], [25] and $\{1\}$-inverse [26] are reported as the accurate methods of traffic matrix estimation, and thus ENNTMI will be compared with them. We use, respectively, the thirty-five-day, forty-two-day, 900-time-slot (about three days) real data from the Abilene, GÉANT, and campus networks to simulate the performance of three methods. The first fourteen-day, twenty-one-day, 500-time-slot data from the Abilene, GÉANT, and campus networks is, respectively, used to train the modified ENN and construct the traffic matrix estimation model, while the rest of data are, respectively, exploited to test three methods.

Figs. 5, 6, and 7 shows traffic matrix tracking of three methods in the Abilene, GÉANT, and campus networks. From these figures, we can evidently see that three methods can track the dy-

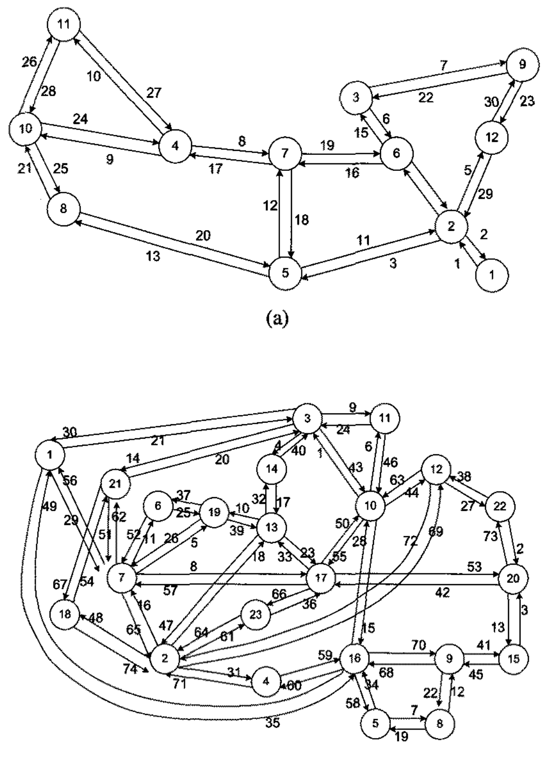


Fig. 4. Backbone network topologies for simulation: (a) Abilene network and (b) GÉANT network.

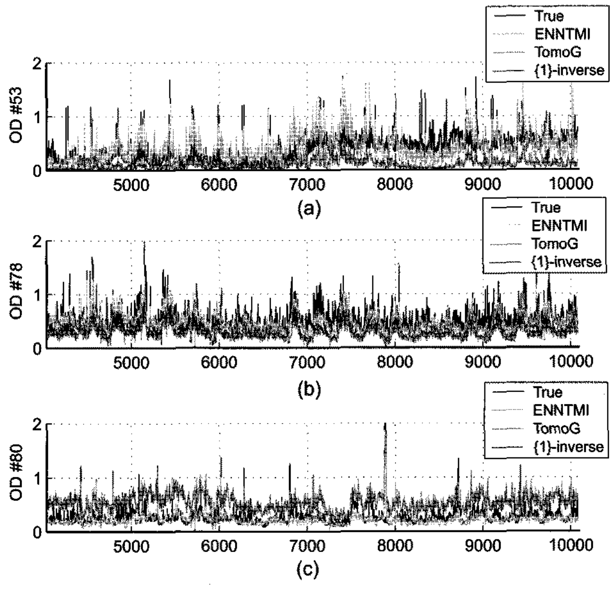


Fig. 5. Estimations of OD flows in Abilene.

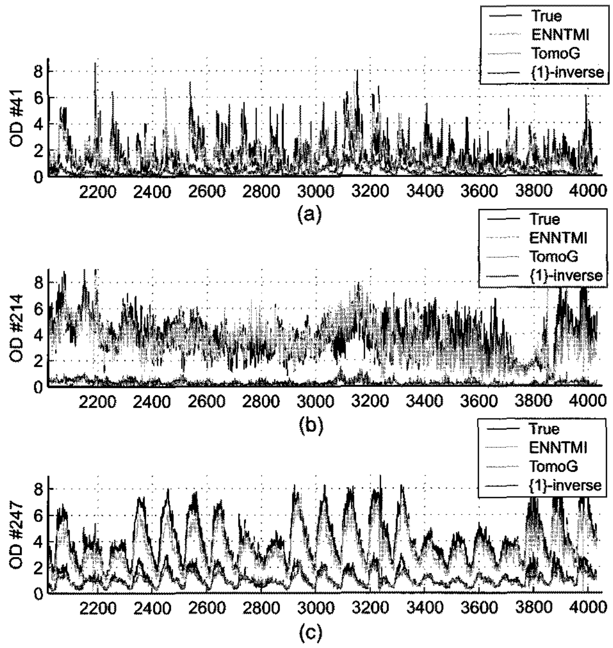


Fig. 6. Estimations of OD flows in GÉANT.

namics of OD flows while ENNTMI can more accurately predict OD flows than the other two methods. TomoGravity and {1}-inverse hold the nearly same estimation accuracy. ENNTMI can track the change trend of OD flows, but TomoGravity and {1}-inverse yield under-estimations or over-estimations. The following discussions will show that ENNTMI holds lower estimation errors. More importantly, we only use the two-week, three-week, or 500-time-slot data to train the estimation model in Fig. 2, while ENNTMI can accurately predict the other three-week or 400-time-slot traffic matrices. Hence, ENNTMI can not only more accurately track the TM's dynamics than TomoGravity and {1}-inverse, but also make a long-term prediction for the traffic matrix.

The SREs and TREs are denoted as follows:

$$\begin{aligned} err_{sp}(n) &= \frac{||\hat{x}_T(n) - x_T(n)||_2}{||x_T(n)||_2}, \\ err_{tm}(t) &= \frac{||\hat{x}_N(t) - x_N(t)||_2}{||x_N(t)||_2}, \\ n &= 1, 2, \cdots, N; \;\; t = 1, 2, \cdots, M \end{aligned} \tag{13}$$

where N and T represent the total number of OD flows and measurement moments, respectively; $||\cdot||_2$ is L_2 norm; $err_{sp}(n)$ and $err_{tm}(t)$ denote the SREs and TREs, respectively. To precisely evaluate the estimation performance of three methods, we examine the cumulative distribution functions (CDFs) of their SREs and TREs. Figs. 8, 9, and 10 show their CDFs in the Abilene, GÉANT, and campus networks, respectively. Figs. 8(a), 9(a), and 10(a) evidently show that the curves of the SREs' CDFs of TomoGravtiy and {1}-inverse are far below that of ENNTMI, while TomoGravtiy's and {1}-inverse's are very close. Furthermore, in the Abilene network, for ENNTMI, about 80% of OD flows are tracked with SREs less than 0.75, while about 72% for TomoGravity and less than 69.1% for {1}-inverse. In the GÉANT network, for ENNTMI, above 52% of OD flows are tracked with SREs less than 0.85, while 29% for TomoGravity and about 37.6% for {1}-inverse. Moreover, in the campus network, for ENTMI, above 88% of OD flows are captured with SREs less than 0.62, while 72% for TomoGravity and 76% for {1}-inverse. This shows that in the Abilene, GÉANT, and campus networks, the spatial estimation errors of ENNTMI are far lower than those of the other two methods, while those of TomoGravity and {1}-inverse is close in Abilene network and TomoGravity's are larger than {1}-inverse's in the GÉANT networks.

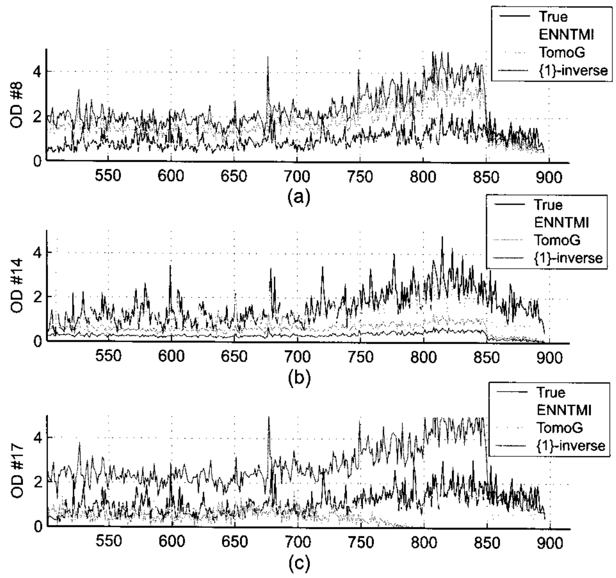


Fig. 7. Estimations of OD flows in campus.

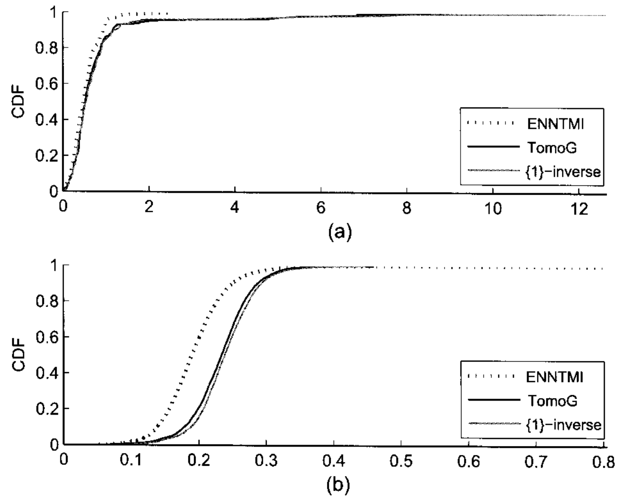


Fig. 8. CDF of spatial and temporal relative errors in Abilene: (a) x = L2 norm, spatial relative errors and x = L2 norm, temporal relative errors

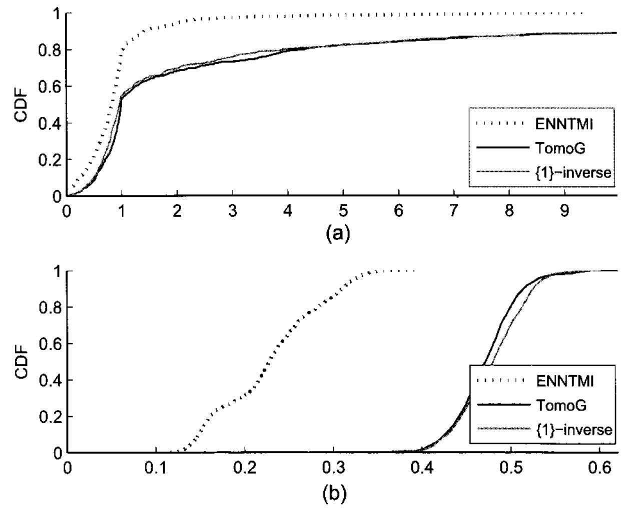


Fig. 9. CDF of spatial and temporal relative errors in GÉANT: (a) x = L2 norm, spatial relative errors and (b) x = L2 norm, temporal relative errors.

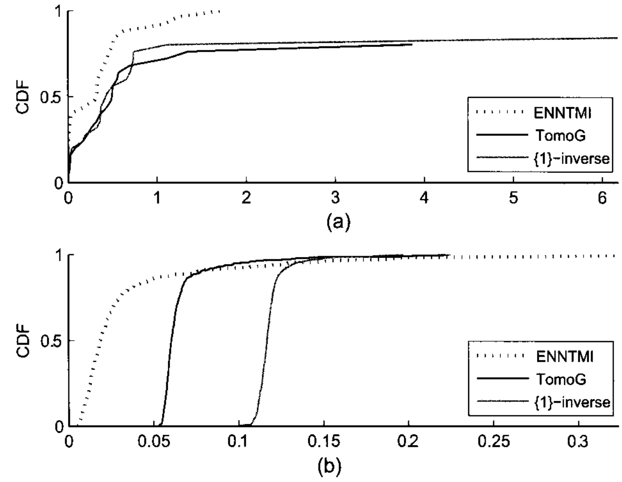


Fig. 10. CDF of spatial and temporal relative errors in campus: (a) x = L2 norm, spatial relative errors and (b) x = L2 norm, temporal relative errors.

For the campus network, when SREs are about less than 0.7, those of TomoGravity and {1}-inverse is close, whereas that of {1}-inverse is always larger than that of TomoGravity.

From Figs. 8(b), 9(b), and 10(b), we can see that the curves of the TREs' CDFs of TomoGravity and {1}-inverse are far below that of ENNTMI, while {1}-inverse's is under TomoGravity's. Moreover, we can also see that, in the Abilene network, about 90% of measurement moments, for ENNTMI, can be tracked with TREs less than 0.25, while about 65.2% for TomoGravity and less than 61% for {1}-inverse. In the GÉANT network, ENNTMI can track 90% of measurement moments with TREs less than 0.32, while TomoGravity and {1}-inverse with TREs 0.515, and about 0.526, respectively. Likewise, in the campus network, ENNTMI can track 80% of measurement moments with TREs less than 0.035, while TomoGravity and {1}-inverse with TREs 0.121, and about 0.067, respectively. This shows that in the Abilene, GÉANT, and campus networks, the temporal estimation errors of TomoGravity and {1}-inverse are much larger than those of TNNTMI. However, {1}-inverse's are larger than TomoGravity's. Therefore, ENNTMI can more accurately estimate traffic matrix than TomoGravity and {1}-inverse.

Fig. 11 shows that the average performance improvement for ENNTMI over TomoGravity is up to 24.2%, 54.1%, and 42.7% in the Abilene GÉANT and campus networks, respectively, while that for ENNTMI over {1}-inverse is equal to 26.6%, 52.7%, and 62.7% in three networks, respectively. In contrast to TomoGravity and {1}-inverse, the improvement of ENNTMI is more significant. This further indicates that for the ill-posed problem of traffic matrix estimation, ENNTMI is promising. More importantly, from the above analysis, we find that in contrast to TomoGravity and {1}-inverse, ENNTMI has more accurate estimation performance in not only larger networks such as Abilene and GÉANT but also smaller networks, for instance campus network.

To further evaluate the performance of the three methods,

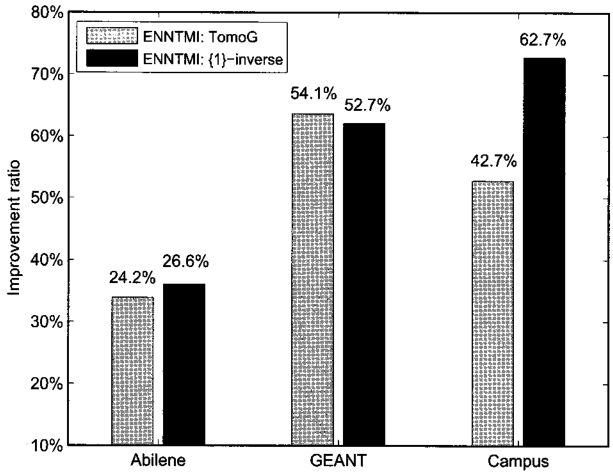


Fig. 11. Average performance improvement for ENNTMI over TomoGravity and {1}-inverse.

the impact of the noise on the three methods is discussed. We introduce an error term $\theta(t)$ to (1) and attain the equation $y_n(t) = \mathbf{A}x(t) + \theta(t)$, where $\theta(t) = y_n(t)\eta(0, \zeta)$, and $\eta(0, \zeta)$ denotes a normal distribution with zero mean and standard deviation. We discuss the robustness of three methods in three cases: $\zeta = 0.02$, $\zeta = 0.03$, and $\zeta = 0.05$. We use the following spatial root mean squared relative error (SRMSRE) and temporal root mean squared relative error (TRMSRE) to evaluate the robustness of the three methods in the Abilene, GÉANT, and campus networks:

$$\text{SRMSRE} = \frac{1}{N}\sum_{n=1}^{N}\frac{||\hat{x}_T(n) - x_T(n)||_2}{||x_T(n)||_2},$$
$$\text{TRMSRE} = \frac{1}{T}\sum_{t=1}^{T}\frac{||\hat{x}_N(t) - x_N(t)||_2}{||x_N(t)||_2}. \quad (14)$$

Tables 1 and 2 show the impact of the noise on the three methods in the Abilene, GÉANT, and campus networks, respectively. From the tables, we can see that, in the three cases: $\zeta = 0.02$, $\zeta = 0.03$, and $\zeta = 0.05$, the ENNTMI's SRMSRE and TRMSRE are still much lower than TomoGravity's and {1}-inverse's. Moreover, the SRMSRE's and TRMSRE's changes of ENNTMI are lower than those of other two methods. Hence, this indicates that ENNTMI is more robust to noise than TomoGravity and {1}-inverse.

IV. CONCLUSION

This paper has proposed a new method call ENNTMI to estimate an IP traffic matrix. On the basis of the conventional Elman neural network, we propose a modified Elman neural network model to overcome the highly ill-posed nature of traffic matrix estimation. We build traffic matrix estimation into an optimal inference process. By introducing the regular parameter into the optimal equation, we can further get rid of the ill-posed nature of this problem. Simulation results show that ENNTMI outperforms the previous methods.

Table 1. Impact of noise on three methods in Abilene.

Noise level		$\zeta = 0.02$	$\zeta = 0.03$	$\zeta = 0.05$
Link loads	SRMSRE	2.02%	3.12%	5.17%
	TRMSRE	1.99%	2.92%	4.99%
ENNTMI	SRMSRE	59.03%	59.97%	62.07%
	TRMSRE	19.13%	19.19%	19.31%
TomoG	SRMSRE	81.01%	82.83%	87.48%
	TRMSRE	24.56%	25.25%	27.55%
{1}-inverse	SRMSRE	87.83%	89.62%	92.37%
	TRMSRE	25.23%	25.85%	26.72%

Table 2. Impact of noise on three methods in GÉANT.

Noise level		$\zeta = 0.02$	$\zeta = 0.03$	$\zeta = 0.05$
Link loads	SRMSRE	2.01%	3.08%	5.11%
	TRMSRE	1.98%	2.99%	5.02%
ENNTMI	SRMSRE	106.57%	107.21%	108.21%
	TRMSRE	24.21%	24.55%	25.26%
TomoG	SRMSRE	155.89%	159.05%	160.91%
	TRMSRE	47.02%	47.46%	48.62%
{1}-inverse	SRMSRE	150.36%	150.98%	152.58%
	TRMSRE	95.73%	96.75%	97.69%

Table 3. Impact of noise on three methods in campus.

Noise level		$\zeta = 0.02$	$\zeta = 0.03$	$\zeta = 0.05$
Link loads	SRMSRE	1.90%	2.88%	4.77%
	TRMSRE	1.95%	2.91%	4.73%
ENNTMI	SRMSRE	51.69%	56.87%	66.69%
	TRMSRE	4.78%	5.26%	7.12%
TomoG	SRMSRE	139.62%	150.05%	169.91%
	TRMSRE	98.12%	104.23%	115.57%
{1}-inverse	SRMSRE	140.62%	145.68%	162.23%
	TRMSRE	47.31%	49.85%	55.39%

ACKNOWLEDGEMENT

The authors thank the reviewers for their helpful comments.

REFERENCES

[1] K. Papagiannaki, N. Taft, and A. Lakhina, "A distributed approach to measure IP traffic matrices," in *Proc. IMC*, 2004, pp. 161–174.

[2] M. Roughan, M. Thorup, and Y. Zhang, "Traffic engineering with estimated traffic matrices," in *Proc. SIGCOMM*, 2003, pp. 248–258.

[3] X. Wang, J. Cao, H. Cheng, and M. Huang, "QoS multicast routing for multimedia group communications using intelligent computational methods," *Computer Commun.*, vol. 29, pp. 2217–2229, 2006.

[4] L. Guo, "LSSP: A novel local segment shared protection for multi-domain optical mesh networks," *Computer Commun.*, vol. 30, pp. 1794–1801, June 2007.

[5] J. Zhang and B. Mukheriee, "A review of fault management in WDM mesh networks: Basic concepts and research challenges," *IEEE Network*, vol. 18, pp. 41–48, 2004.

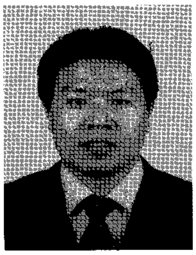
[6] L. Guo, J. Cao, H. Yu, and L. Li, "Path-based routing provisioning with mixed shared protection in WDM mesh networks," *J. Lightwave Technol.*, vol. 24, pp. 1129–1141, Mar. 2006.

[7] L. Paradis and Q. Han, "A survey of fault management in wireless sensor networks," *J. Netw. Syst. Manage.*, vol. 15, pp. 171–190, 2007.

[8] M. Gagnaire, M. Koubaa, and N. Puech, "Network dimensioning under

scheduled and random lightpath demands in all-optical WDM networks," *IEEE J. Sel. Areas Commun.*, vol. 25, pp. 58–67, 2007.
[9] Y. Bejeranoa, S. Hanb, and A. Kumarc, "Efficient load-balancing routing for wireless mesh networks," *Computer Netw.*, vol. 51, pp. 2450–2466, 2007.
[10] L. Guo, "A new and improved algorithm for dynamic survivable routing in optical WDM mesh networks," *Computer Commun.*, vol. 30, pp. 1419–1423, Mar. 2007.
[11] Q. Song and A. Jamalipour, "Network selection in an integrated wireless LAN and UMTS environment using mathematical modeling and computing techniques," *IEEE Wireless Commun.*, vol. 12, pp. 42–48, 2005.
[12] R. Rodrigues, G. Mateus, and A. Loureiro, "On the design and capacity planning of a wireless local area network," in *Proc. NOMS*, 2000, pp. 335–348.
[13] Q. Song and A. Jamalipour, "A network selection mechanism for next generation networks," in *Proc. ICC*, 2005, pp. 1418–1422.
[14] L. Guo and L. Li, "A novel survivable routing algorithm with partial shared-risk link groups (SRLG)-disjoint protection based on differentiated reliability constraints in WDM optical mesh networks," *J. Lightwave Technol.*, vol. 25, pp. 1410–1415, June 2007.
[15] L. Lau, J. Naor, M. Salavatipour, and M. Singh, "Survivable network design with degree or order constraints," in *Proc. STOC*, 2007, pp. 651–660.
[16] G. Borradaile and P. Klein, "The two-edge connectivity survivable network problem in planar graphs," *Lecture Notes in Computer Science*, vol. 5125, pp.485–501, 2008.
[17] L. Guo, L. Li, J. Cao, H. Yu, and X. Wei, "On finding feasible solutions with shared backup resources for surviving double-link failures in path protected WDM mesh networks," *J. Lightwave Technol.*, vol. 25, pp. 287–296, Jan. 2007.
[18] J. Jia, J. Chen, G. Chang, and Z. Tan, "Energy efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm," *Comput. Mathematics Appl.*, vol. 57, pp. 1756–1766, 2007.
[19] O. Younis, M. Krunz, and S. Ramasubramanian, "Node clustering in wireless sensor networks: Recent developments and deployment challenges," *IEEE Netw.*, vol. 20, pp. 20–25, 2006.
[20] J. Jia, J. Chen, G. Chang, Y. Wen, and J. Song, "Multi-objective optimization for coverage control in wireless sensor network with adjustable sensing radius," *Comput. Mathematics Appl.*, vol. 57, pp. 1767–1775, 2009.
[21] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees, "Deploying a wireless sensor network on an active volcano," *IEEE Internet Comput.*, vol. 10, pp. 18–25.
[22] J. Jia, J. Chen, G. Chang, C. Tian, and W. Qin, "Maximization for wireless sensor network lifetime with power efficient cover set alternation," in *Proc. ICCCAS*, 2008, pp. 439–443.
[23] D. Jiang, J. Chen, and L. He, "An accurate approach of large-scale IP traffic matrix estimation," *IEICE Trans. Commun.*, vol. E90-B, pp. 3673–3676, Dec. 2007.
[24] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of IP traffic matrices from link loads," *ACM SIGMETRICS Performance Evaluation Rev.*, vol. 31, pp. 206–217, 2003.
[25] Y. Zhang, M. Roughan, C. Lund, and D. L. Donoho, "Estimating point-to-point and point-to-multipoint traffic matrices: An information-theoretic approach," *IEEE/ACM Trans. Netw.*, vol. 13, pp. 947–960, 2005.
[26] L. Tan and X. Wang, "A novel method to estimate IP traffic matrix," *IEEE Commun. Lett.*, vol. 11, pp. 907–909, 2007.
[27] A. Soule, A. Lakhina, N. Taft, K. Papagiannaki, K. Salamatian, A. Nucci, M. Crovella, and C. Diot, "Traffic matrices: Balancing measurements, inference and modeling," *ACM SIGMETRICS Performance Evaluation Rev.*, vol. 33, pp. 362–373, 2005.
[28] C. Tebaldi and M. West, "Bayesian inference on network traffic using link count data," *J. American Stat. Ass.*, vol. 93, pp. 557–576, 1998.
[29] J. Cao, D. Davis, S. V. Weil, and B. Yu, "Time-varying network tomography: Router link data," *J. American Stat. Ass.*, vol. 95, pp. 1063–1075, 2000.
[30] I. Juva, "Sensitivity of traffic matrix estimation techniques to their underlying assumptions," in *Proc. ICC*, 2007, pp. 562–568.
[31] F. Lin, L. Teng, and H. Chu, "Modified Elman neural network controller with improved particle swarm optimisation for linear synchronous motor drive," *IET Electric Power Appl.*, vol. 2, pp. 201–214, 2008.
[32] [Online]. Available: http://www.cs.utexas.edu/ yzhang/research/Abilene-TM
[33] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, "Providing public intradomain traffic matrices to the research community," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, pp. 83–86, 2006.

Xingwei Wang received B.Sc., M.Sc., and Ph.D. in computer science from Northeastern University, China, in 1989, 1992, and 1998, respectively. He is currently a Professor in the College of Information Science and Engineering, Northeastern University, China. His research interests are mainly on routing algorithms and protocols, QoS control schemes, fault-tolerance and survivability models, mobility management mechanisms and resource assignment methods in NGI, and IP over DWDM optical Internet and mobile Internet. He has published over 100 technical papers in the above areas. As one of the co-investigators, he has won the national science and technology advancement award of China twice.



Dingde Jiang received Ph.D. 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. He 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 will receive B.Sc. in College of Information Science and Engineering, Northeastern University, Shenyang, China, in 2010. His research interests include network measurement and cognitive network.