A Linear Regression-Based Prediction Method to Traffic Flow for Low-Power WAN with Smart Electric Power Allocations

Bing Liu[1], Fanbo Meng[2], Yun Zhao[1], Xinge Qi[1], Bin Lu[2], Kai Yang[3], and Xiao Yan[3](✉)

[1] State Grid Dalian Electric Power Supply Company, Dalian 116011, China
[2] State Grid Liaoning Electric Power Company Limited,
Shenyang 110006, China
amengfb@163.com
[3] School of Aeronautics and Astronautics, UESTC, Chengdu 611731, China
yanxiao@uestc.edu.cn

Abstract. Currently power telecommunication access networks have many new requirements to meet the low-power WAN with smart electric power allocations. In such a case, network traffic in the low-power WAN has exhibited new features and there are some challenges for network managements. This paper uses the linear regression model to propose a new method to model and predict network traffic. Firstly, network traffic is modeled as a linear regression model according to the regression model theory. Then the linear regression modeling method is used to capture network traffic features. By calculating the parameters of the model, it can be decided correctly. Then, we can predict network traffic accurately. Simulation results show that our approach is effective and promising.

Keywords: Network traffic · Low-power WAN · Linear regression · Traffic modeling · Traffic prediction

1 Introduction

With current network technology development increasingly quickening and new applications quickly appearing in low-power WAN with smart electric power allocations, more and more new features have embodied in network traffic. This leads to a larger challenging for network engineering in low-power WAN [1–3]. To effectively guaranteeing electric power network performance, we need to accurately model network traffic characteristics. Low-power WAN traffic holds many properties, such as burst, self-similarity, spatio-temporal correlations and so on, which has a direct impact on network performance and management [4–7]. The low-power WAN traffic holds network-level behaviors. Hence, from a global view, network-level traffic modeling has received more and more attention from researchers, operators, and developer over the whole world [8–14]. This has become a hot research topic.

The traffic behaviors in low-power WAN for smart electric power allocations hold network-level nature, which is often used to describe kinds of network behaviors, such

H. Song et al. (Eds.): SIMUtools 2019, LNICST 295, pp. 125–134, 2019.
https://doi.org/10.1007/978-3-030-32216-8_12

as path loads, network throughput, network utilization, and so on. The statistical methods [1, 3], gravity model [4], generic evolvement [5–7], mix method [2], and compressive sensing [12] are utilized to capture the properties of the network-level traffic in low-power WAN. Although these methods attain better prediction and estimation performance for network-level traffic, they produced larger errors and often additional link load information. Hence, it is necessary to research new traffic prediction approach for low-power WAN with smart electric power allocations. The time-frequency analysis was used to describe multi-scale features of network traffic [1, 9]. Neural network models were utilized to model network-level traffic [10–15]. Moreover, network traffic prediction methods are extensively applications [16–20]. These methods still hold a larger prediction error.

In this paper, we propose a novel method to characterize and analyze network traffic accurately. Our method is based on the linear regression modeling theory. Generally, we have difficulties in modeling and describing network traffic because of their highly time-varying nature, which is difficult to be described via the analytical formulation. In this paper, we exploit the linear regression model to characterize network traffic. The linear regression theory is used to build the model parameters via the sample data about network traffic. To the end, firstly we denote a linear regression model over time. Secondly, by calculate the model parameters, we correctly create the prediction model for network traffic based on the linear regression model. Thirdly, we propose a new prediction algorithm to estimate and forecast network traffic accurately. Simulation results show that our approach is effective and promising.

The rest of this paper is organized as follows. Our method is derived in Sect. 2. Section 3 presents the simulation results and analysis. We then conclude our work in Sect. 4.

2 Problem Statement

The model of the time-varying network traffic for the power telecommunication access network is very hard to build. The traffic in the network is fluctuation along with the business volume and time, and the features of flow is very hard to express it directly, so it is a huge challenge to model the traffic in the power telecommunication access network. Here, we donate the traffic in network-level as $y = \{y(1), y(2), \ldots, y(t)\}$, where $y(t)$ is the traffic value of flow y at the time slot t. We assume that the traffic value $y(t)$ in the network satisfies the independent identically distributed random process.

According to the linear regression analysis theory, linear regression model can be written as follow

$$\begin{cases} y = b_0 + b_1 x_1 + \ldots + b_m x_m + \ldots + b_n x_n + \varepsilon \\ E(\varepsilon) = 0, 0 < D(\varepsilon) = \sigma^2 < +\infty \end{cases} \tag{1}$$

where b_0 represents the constant variable, ε represents the random error. b_m (where $m = 1, 2, \ldots, n$) represents the partial regression coefficient, x_m (where $m = 1, 2, \ldots, n$) represents the values of many experiments. $E(\varepsilon)$ is the mean value of the random error, and $D(\varepsilon)$ is the variance of the random error. The random errors is a normal distribution whose mean and variance are zero and σ^2, respectively. So, the distribution of the random error can be expressed as

$$p(\varepsilon) = \frac{1}{\sqrt{2\pi}}\exp(-\frac{\varepsilon^2}{2\sigma^2}) \tag{2}$$

In the network, we know that there are many users connect into the network at the same time, many flows are transmitted in the network from one node to another at the same time. The end-to-end traffic y is regarded as a statistical variable which can be expressed as the Eq. (1). According to the analysis in [10–12], we know that the traffic in the access network has the correlation over time. In order to retrieve the feature of traffic in the power telecommunication access networks, we express the network traffic with the linear regression theory, so the traffic in the network can be expressed as

$$y = b_0 + \sum_{i=1}^{n} b_i x(i) + \varepsilon, \varepsilon \sim N(0, \sigma^2) \tag{3}$$

where b_i (where $i = 1, 2, \ldots, n$) denotes partial regression coefficient. ε represents the residual error when process network traffic. $x(i)$ is the related features of flow traffic.

As we assumed earlier, $y(t)$ (where $t = 1, 2, \ldots$) is the traffic instance at time slot t. From Eq. (3), we know that the statistic traffic at slot t is correlation with characterizes of flows. Equation (3) shows the statistics of traffic $y(t)$ and the characterizes of flows $x(i)$ (where $i = 1, 2, \ldots, n$). $x(i)$ denotes the network traffic which can be obtain at time slot i. And the residual error between the estimation and the actual traffic is ε, so we can obtain the traffic $y(t)$ at slot t. Due to the residual error of the estimation is normal distribution. Based on the liner regression model, the traffic at slot t can be expressed as

$$y(t) = b_0 + \sum_{i=1}^{n} b_i x_t(i) + \varepsilon, \varepsilon \sim N(0, \sigma^2) \tag{4}$$

where b_0 and b_i (where $i = 1, 2, \ldots, n$) are the regression constant and partial regression coefficient, respectively. $x_t(i)$ is the characterizes of flow $x(i)$ at time slot t. So the mean of traffic satisfies that

$$E(y(t)) = E(b_0 + \sum_{i=1}^{n} b_i x_t(i)) \tag{5}$$

where $E(\cdot)$ is the expression of expectation value. If there are k measurements and the characterizes $n > k$, so the linear regression can be expressed as

$$\begin{cases} y(1) = b_0 + b_1 x_1(1) + b_2 x_1(2) + \ldots + b_n x_1(n) + \varepsilon_1 \\ y(2) = b_0 + b_1 x_2(1) + b_1 x_2(2) + \ldots + b_1 x_2(n) + \varepsilon_2 \\ \ldots \\ y(k) = b_0 + b_1 x_k(1) + b_1 x_k(2) + \ldots + b_1 x_k(n) + \varepsilon_k \end{cases} \tag{6}$$

Then, we express (6) as a matrix

$$Y = XB + \Theta \tag{7}$$

where $Y = [y(1), \ldots, y(k)]^T$, $B = [b_0, b_1, \ldots, b_n]^T$, $\Theta = [\varepsilon_1, \ldots, \varepsilon_k]^T$ and $X = \begin{bmatrix} 1 & x_1(1) & x_1(2) & \ldots & x_1(n) \\ 1 & x_2(1) & x_2(2) & \ldots & x_2(n) \\ \ldots & \ldots & \ldots & x_j(i) & \ldots \\ 1 & x_k(1) & x_k(2) & \ldots & x_k(n) \end{bmatrix}$.

The elements $x_j(i)$ of matrix X can be obtain from history data. We assume that the estimates of partial regression coefficients are $\{\hat{b}_0, \hat{b}_1, \ldots, \hat{b}_n\}$, so the Eq. (4) can be written as

$$\hat{y}(t) = \hat{b}_0 + \hat{b}_1 x_t(1) + \hat{b}_2 x_t(2) + \ldots + \hat{b}_n x_t(n) \tag{8}$$

where $\hat{y}(t)$ is the estimates of $y(t)$.

The residual error at time slot t is

$$\begin{aligned} \varepsilon_t &= y(t) - \hat{y}(t) \\ &= y(t) - (\hat{b}_0 + \hat{b}_1 x_t(1) + \hat{b}_2 x_t(2) + \ldots + \hat{b}_n x_t(n)) \end{aligned} \tag{9}$$

Then, we use the ordinary least square (OLS) to estimate the residual errors. Here, we firstly make some assumptions in the following.

Assumption 1: the mean value of residual errors is zero.

$$E(\Theta) = E([\varepsilon_1, \ldots, \varepsilon_k]^T) = [E(\varepsilon_1), \ldots, E(\varepsilon_k)]^T = 0 \tag{10}$$

Assumption 2: residual errors have the same distribution.

$$Var(\varepsilon_j) = E(\varepsilon_j^2) = \sigma^2, j = 1, 2, \ldots, k \tag{11}$$

Assumption 3: There is no correlation between residual errors.

$$Cov(\mu_i, \mu_j) = E(\mu_i \mu_j) = 0, i, j = 1, 2, \ldots, n \tag{12}$$

Assumption 4: The residual error and characteristics of flow $x_t(i)$ have no relevance.

$$Cov(x_j(i), \mu_j) = E(x_j(i)\mu_j) = 0, j = 1, 2, .., t, i = 1, 2, \ldots, n \tag{13}$$

Based on the least square method, we know that the regression constant and partial regression coefficients should minimize sum of squares of residual errors, so

$$\begin{aligned} RSS(\hat{b}_0, \hat{b}_1, \ldots, \hat{b}_n) &= \arg\min(\sum_{k=1}^{t} \varepsilon_k^2) \\ &= \arg\min(\sum_{k=1}^{t} (y(k) - \hat{y}(k))^2) \end{aligned} \tag{14}$$

For an example, we make experiments that when $n = 3$ here. So, the traffic at time slot t can be rewritten as

$$y(t) = b_0 + b_1 x(1) + b_2 x(2) + b_3 x(3) + \varepsilon, \varepsilon \sim N(0, \sigma^2) \tag{15}$$

and the estimates of $y(t)$ can be written as

$$\hat{y}(t) = \hat{b}_0 + \hat{b}_1 x(1) + \hat{b}_2 x(2) + \hat{b}_3 x(3) + \varepsilon, \varepsilon \sim N(0, \sigma^2) \tag{16}$$

According to Eqs. (14)–(16) and Eq. (2), we make a optimization to find the regression coefficients and the residual error.

$$\begin{cases} f(b_0, b_1, b_2, b_3, \varepsilon) \\ s.t.\ \ \hat{y}(k) = \hat{b}_0 + \hat{b}_1 x(1) + \hat{b}_2 x(2) + \hat{b}_3 x(3) + \varepsilon \\ \quad\ \ p(\varepsilon) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{\varepsilon^2}{2\sigma^2}) \\ \quad\ \ RSS(\hat{b}_0, \hat{b}_1, \ldots, \hat{b}_n) = \arg\min(\sum\limits_{k=1}^{t} (y(k) - \hat{y}(k))^2) \\ \quad\ \ \varepsilon = y(k) - \hat{y}(k) \\ \quad\ \ (b_0, b_1, \ldots, b_n) = (\hat{b}_0, \hat{b}_1, \ldots, \hat{b}_n) \end{cases} \tag{17}$$

Equation (17) is a multi-constraint and multi-object optimization issue. The first constraint of Eq. (17) indicates estimate $\hat{y}(k)$ of flow traffic at time slot k. The second one donates the distribution of residual errors, and the third one means the optimal estimates of partial regression coefficient and $y(k)$ is the measured traffic value at time slot k. The fourth equation in (17) calculate residual errors between the measured value of traffic and the prediction under the partial regression coefficient $(\hat{b}_0, \hat{b}_1, \ldots, \hat{b}_n)$. By training the model of (4) and adjusting the residual errors with the (2), we can obtain the prediction model and the set of parameters.

We present our prediction algorithm based on linear regression model, called Linear Regression Model Theory Traffic Modeling Algorithm (LMTMA). Based on the analysis and derivation above, the steps of algorithm LMTMA are as following.

Step 1: Given t initial measured value of the end-to-end network traffic in the network-level $y = \{y(1), y(2), \ldots., y(t)\}$ in the front t time slots.
Step 2: Based on the linear regression model theory and the statistical analysis methods, we initialize network traffic $y(t)$ and parameters of σ^2, respectively.
Step 3: Build the traffic prediction model (4) and distribution of the residual errors (2) to find the estimate of the partial regression coefficient b_0, b_1, b_2, b_3.

Step 4: In objective function (17), minimize the residual errors ε and update the partial regression coefficient b_0, b_1, b_2, b_3.
Step 5: obtain the optimal parameters $b_0, b_1, b_2, b_3, \varepsilon$ from objective function (17).
Step 6: The traffic prediction model is constructed over, then exist the process of modeling.

3 Simulation Results and Analysis

Now, we conduct many tests to demonstrate our algorithm LMTMA. In order to justify the accuracy of our algorithm, we need to use real network data. Here, the real data needed in the simulation experiment is collected by the network nodes deployed at different place; we use the real data from the real Abilene backbone network in the United States to validate LMTMA. PCA [3], WABR [7], and HMPA [2] algorithms for the network traffic modeling have been reported as the better performance. Here we compare LMTMA with them. In the following, the prediction results of the network traffic are analyzed for LMTMA algorithm. The average relative errors for the network traffic are indicated for four algorithms. Finally, we discuss the performance improvement of LMTMA against PCA, WABR, and HMPA. In our simulation, the data of the first 500 time slots are used to train the models of four approaches, while other data are exploited to validate the performance of all algorithms.

Figure 1 shows the prediction results of network traffic flows 78 and 118, where network traffic flows 78 and 118 are selected randomly from the 144 end-to-end traffic flow pairs in our simulation network. Without loss of generality, we only discuss the network traffic flows 78 and 118 in this paper. The network traffic flows is also called as the Origin Destination (OD) pair. Figure 1(a) indicates that LMTMA can effectively capture the dynamic changes of the network traffic flow 78. For different time slots, the real network traffic exhibits the significant time-varying nature. From Fig. 1(a), we have seen that LMTMA can seek the trend of the network traffic flow. Likewise, the network traffic flow 118 shows the irregular and dynamic changes over the time as indicated in Fig. 1(b). From Fig. 1(b), it is very clear that although LMTMA holds the larger prediction errors for the network traffic flow 118, it can still capture its change trend. This further indicates that LMTMA can effectively predict the change of the network traffic over the time.

Next, we discuss the predict errors of four algorithms. Generally, we have difficulties in seizing the dynamic nature of the network traffic over the time via the model. To further validate our algorithm, we compare the relative prediction errors over the time for all algorithms. To avoid the randomness in the simulation process, we perform 500 runs to calculate the average relative prediction errors.

The average relative prediction errors over the time for network traffic reflect the performance of the methods for predicting network traffic, they are donated as:

$$d(t) = \frac{1}{N}\sum\nolimits_{i=1}^{N} \frac{||\hat{y}_i(t) - y_i(t)||_2}{||y_i(t)||_2} \tag{18}$$

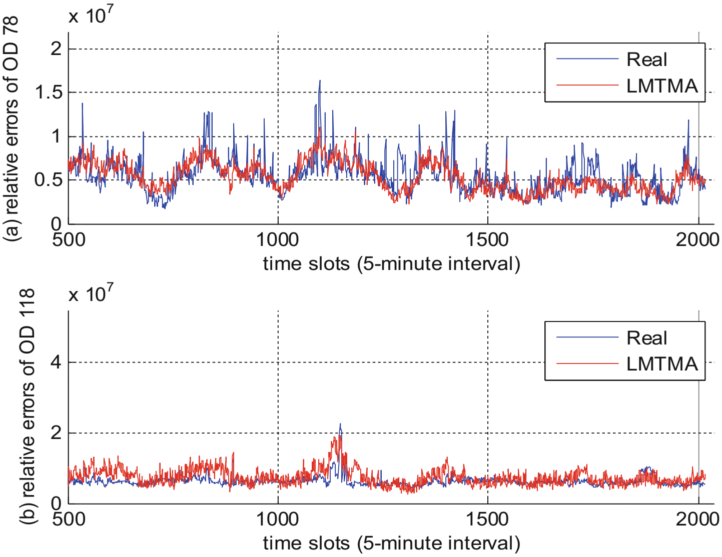

Fig. 1. Prediction results of network traffic flows 78 and 118.

where $i = 1, 2, \ldots, N$, N is the number of runs in the simulation process, $\|\cdot\|_2$ is the norm of L_2, and $\hat{y}_i(t)$ indicates the traffic prediction value of run i at time slot t.

Figure 2 shows the average relative prediction errors of four algorithm over the time for network-level traffic flows 78 and 118. We can find that for network traffic flows 78 and 118, WABR, HMPA, and LMTMA exhibit the lower relative errors while

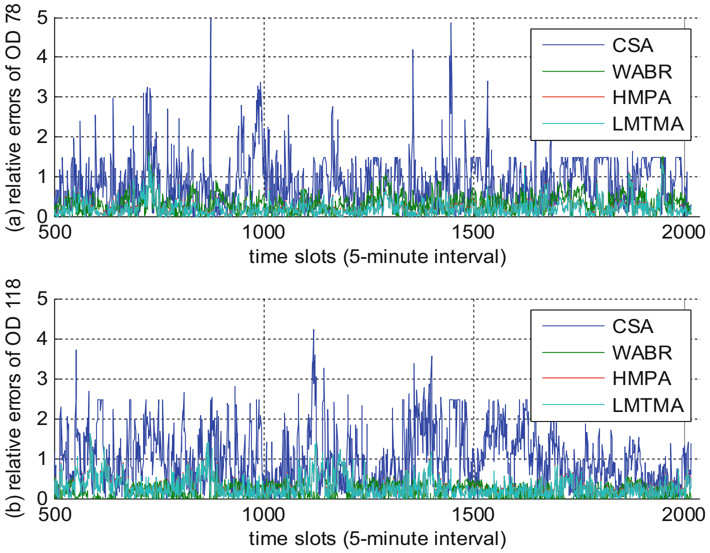

Fig. 2. Average relative errors for network traffic flows 78 and 118.

PCA hold the larger prediction bias. For Fig. 2, we can also see that that LMTMA holds the lowest relative errors. This tells us that in contrast to PCA, WABR, and HMPA, LMTMA holds the better prediction ability for the network traffic, while LMTMA holds the best prediction ability. More importantly, WABR, HMPA, and LMTMA indicate the lower fluctuation over the time in terms of relative errors than PCA. This shows that compared with other three algorithms, LMTMA can more effectively model the network traffic with time-varying and correlation features. Moreover, this also tell us that LMTMA can more accurately predict network-level traffic than previous methods.

Now, we analyze the performance improvement of LMTMA relative to other three algorithms for the network traffic. Figure 3 exhibits the performance improvement ration of network traffic flow 78 and 118. For network traffic flow 78, LMTMA attains the performance improvement against PCA, WABR, and HMPA, respectively. Similarly, for network traffic flow 118, LMTMA obtains the performance improvement against PCA, WABR, and HMPA, respectively. This clearly denotes that compared with PCA, WABR, and HMPA, our algorithm LMTMA can more accurately model the network-level network traffic. Moreover, Fig. 3 also tell us that relative to PCA and WABR, LMTMA can reach the larger performance improvement. Compared with HMPA, LMTMA also reaches to the better performance improvement. As mentioned in Fig. 2, this further illustrates that our algorithm LMTMA holds the better modeling capability for the network-wide network traffic. Moreover, LMTMA and HMPA hold the similar performance, while LMTMA exhibits the better performance improvement. This also shows that LMTMA can correctly model the network traffic and hold better modeling performance for network traffic.

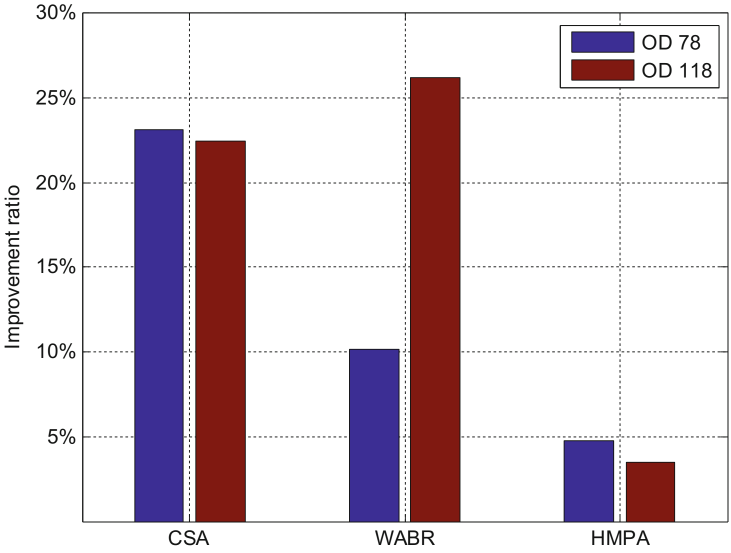

Fig. 3. Improvement ratio of network traffic flows 78 and 118.

4 Conclusions

This paper proposes a linear regression theory-based method to model and predict network traffic. Different from previous methods, the linear regression model is used to construct and determine the model parameters effectively. Firstly, the network traffic is described as an independent identically distributed exponential distribution process. Secondly, the linear regression method is exploited to capture the network-level network traffic. By calculating the parameters of the model, we build the corresponding network traffic model. Simulation results show that our approach is promising and effective.

References

1. Jiang, D., Xu, Z., Chen, Z., et al.: Joint time-frequency sparse estimation of large-scale network traffic. Comput. Netw. **55**(10), 3533–3547 (2011)
2. Jiang, D., Xu, Z., Xu, H.: A novel hybrid prediction algorithm to network traffic. Ann. Telecommun. **70**(9), 427–439 (2015)
3. Soule, A., Lakhina, A., Taft, N., et al.: Traffic matrices: balancing measurements, inference and modeling. In: Proceedings of SIGMETRICS 2005, vol. 33, no. 1, pp. 362–373 (2005)
4. Takeda, T., Shionoto, K.: Traffic matrix estimation in large-scale IP networks. In: Proceedings of LANMAN 2010, pp. 1–6 (2010)
5. Yingxun, F.: The Research and Improvement of the Genetic Algorithm. Beijing University of Posts and Telecommunications, Beijing (2010)
6. Jiang, D., Zhao, Z., Xu, Z., et al.: How to reconstruct end-to-end traffic based on time-frequency analysis and artificial neural network. AEU-Int. J. Electron. Commun. **68**(10), 915–925 (2014)
7. Jiang, D., Yuan, Z., Zhang, P., et al.: A traffic anomaly detection approach in communication networks for applications of multimedia medical devices. Multimedia Tools Appl. **75**, 14281–14301 (2016)
8. Jiang, D., Xu, Z., Nie, L., et al.: An approximate approach to end-to-end traffic in communication networks. Chin. J. Electron. **21**(4), 705–710 (2012)
9. Vaton, S., Bedo, J.: Network traffic matrix: how can one learn the prior distributions from the link counts only. In: Proceedings of ICC 2004, pp. 2138–2142 (2004)
10. Lad, M., Oliveira, R., Massey, D., et al.: Inferring the origin of routing changes using link weights. In: Proceedings of ICNP, pp. 93–102 (2007)
11. Jiang, D., Xu, Z., Li, W., et al.: Topology control-based collaborative multicast routing algorithm with minimum energy consumption. Int. J. Commun Syst **30**(1), 1–18 (2017)
12. Jiang, D., Nie, L., Lv, Z., et al.: Spatio-temporal Kronecker compressive sensing for traffic matrix recovery. IEEE Access **4**, 3046–3053 (2016)
13. Tune, P., Veitch, D.: Sampling vs sketching: an information theoretic comparison. In: Proceedings of INFOCOM, pp. 2105–2113 (2011)
14. Jiang, D., Li, W., Lv, H.: An energy-efficient cooperative multicast routing in multi-hop wireless networks for smart medical applications. Neurocomputing **220**(2017), 160–169 (2017)
15. Zhang, Y., Roughan, M., Duffield, N., et al.: Fast accurate computation of large-scale IP traffic matrices from link loads. In: Proceedings of SIGMETRICS 2003, vol. 31, no. 3, pp. 206–217 (2003)

16. Jiang, D., Wang, Y., Han, Y., et al.: Maximum connectivity-based channel allocation algorithm in cognitive wireless networks for medical applications. Neurocomputing **2017** (220), 41–51 (2017)
17. Jiang, D., Wang, W., Shi, L., Song, H.: A compressive sensing-based approach to end-to-end network traffic reconstruction. IEEE Trans. Netw. Sci. Eng. (2018). https://doi.org/10.1109/tnse.2018.2877597
18. Jiang, D., Huo, L., Song, H.: Rethinking behaviors and activities of base stations in mobile cellular networks based on big data analysis. IEEE Trans. Netw. Sci. Eng. **1**(1), 1–12 (2018)
19. Jiang, D., Huo, L., Li, Y.: Fine-granularity inference and estimations to network traffic for SDN. PLoS ONE **13**(5), 1–23 (2018)
20. Jiang, D., Huo, L., Lv, Z., et al.: A joint multi-criteria utility-based network selection approach for vehicle-to-infrastructure networking. IEEE Trans. Intell. Transp. Syst. **99**, 1–15 (2018)