An Adaptive and Efficient Network Traffic Measurement Method Based on SDN in IoT

Wansheng Cai[1], Xi Song[2], Chuan Liu[3], Dingde Jiang[4](✉), and Liuwei Huo[4]

[1] State Grid Electric Power Research Institute, Nanjing 210003, China
[2] State Grid Gansu Electric Power CORP., Lanzhou 730000, China
[3] Global Energy Interconnection Research Institute Co., Ltd., Nanjing 210003, China
liuchuan@geiri.sgcc.com.cn
[4] School of Astronautics and Aeronautics, University of Electronic Science and Technology of China, Chengdu 611731, China

Abstract. The Internet of Things (IoT) is a worldwide information network that connects thousands of technological gadgets. We incorporate the SDN network architecture into IoT networks and investigate the characteristics of SDN-based IoT networks in order to make the IoT more flexible and extendable. SDN (Software Defined Networking) is a logical control center with a centralized control plane that makes network management more flexible and efficient. For IoT network management, fine-grained and reliable traffic information is critical. Then, in SDN-based IoT networks, we construct a network traffic model by analyzing the self-similarity of network traffic in IoT network. Then, we collect some traffic statistics in OpenFlow-based switches as the source data and use it to train the proposed network traffic estimation model. Using the measured network traffic in the IoT network, we use the Kalman Filtering to measure and estimate each flow, this scheme just increases a little overhead. Then, we propose to an algorithm to search the more accuracy of traffic. Finally, we run additional simulations to ensure that the suggested measuring system is accurate. Simulation findings suggest that using intelligent optimization approaches, we can improve the granularity and accuracy of traffic data.

Keywords: Internet of Things · Software Defined Network · Optimization algorithm · Network measurement

1 Introduction

The Internet of Things (IoT) is a widely used communication technology that has changed people's lifestyles. It has been widely used in smart homes, intelligent transportation, intelligent logistics, smart medical, smart factories, intelligent agriculture, and other areas to create a more convenient life for citizens [1]. Cloud computing is at the heart of the IoT, processing and storing data. Hundreds of thousands of computers make up the cloud computing center, which allows customers to access application services from any location via a variety of terminals. Cloud computing is a general-purpose computing

D. Jiang and H. Song (Eds.): SIMUtools 2021, LNICST 424, pp. 64–74, 2022.
https://doi.org/10.1007/978-3-030-97124-3_6

platform with significant computing and storage capabilities that allows multiple programs to run concurrently [2]. Cloud resources can be sold based on consumer demand, and their scale can be dynamically adjusted to meet the needs of rising applications and users. In cloud computing, there is a lot of data to process and exchange, therefore accurate network traffic statistics is critical for network management.

The network architecture of the legacy IoT network is not scalable and does not meet the requirements of the increasing devices. The goal of Software Defined Networking (SDN) is to make network management easier and the IoT more flexible. To simplify network management policies and dynamically configure network rules, SDN isolates the control plane from the underlying forwarding device and incorporates it into the logically centralized controller [3–5]. Because the controller in SDN has a global view of the networks, it may build global traffic dispatching optimization rules. However, the applications process associated to cloud computing for requesting specific services, such as long-term data storage, applications requests, and resource requests, is susceptible to latency and bandwidth constraints. As a result, most cloud computing services are primarily concerned about latency and bandwidth. In a cloud computing network, fine-grained network traffic information acquisition is critical for traffic control. The importance of traffic engineering for cloud computing architecture cannot be overstated. Hardware support and remote monitoring agent software are required for each of them. SDN offers a revolutionary flow-based statistical measuring method that is both versatile and convenient for collecting traffic statistics from switches.

Through the OpenFlow protocol, SDN provides two techniques for collecting flow statistics: pull-based and push-based [6]. A pull-based technique is an active method of gathering statistical data that does not require any additional hardware or software. All that is required is for the controller to deliver commands to the OpenFlow-based switch. Programs for controlling devices to collect flow statistics or port statistics information, read-state messages and deliver them to OpenFlow-based switches [7]. Jiang et al. [8, 9] investigated network traffic aspects in a communications access network.

We discuss pull-based and flow-based network traffic measurement in cloud computing architecture based on the aforesaid analysis. In this research, we present a low-overhead traffic information acquisition scheme and build a unique cloud computing network measurement architecture. We directly measure some network traffic metrics and forecast fine-grained network traffic. Then, to reduce the fine-grained measurement error inferred, we propose an intelligent optimization model and a heuristic approach to find the model's optimal solution. We offer a model to forecast network traffic and use an optimization approach to achieve fine-grained network traffic in this system. The following are the primary contributions we make in this paper:

- We present a framework for measuring network traffic in IoT networks. We collect statistics of flow traffic to construct a traffic matrix and clean them as the training set of the proposed model.
- We proposed the NEKF algorithm to measure and estimate the network traffic in the IoT network.
- Finally, we run additional simulations to test the suggested measuring scheme's performance.

The following is how the rest of the paper is structured: The key security challenges of SDN are discussed in Sect. 2, after which we present a fine-grained measurement method and describe fine-grained measurement prediction and optimization in the IoT paradigm. The performance of fine-grained measurement with the scheme is simulated in Sect. 3. Finally, Sect. 4 brings our works to a conclusion.

2 Problem Statement

In the IoT network, there are huge amount of devices that connect into the network, and network architecture is dynamic, to measure and manage the network effectively, we think about the IoT network architecture based on SDN, and proposed the active network traffic measurement scheme in the SDN-based IoT network.

2.1 The Self-similarity of IoT Network Traffic

In the Internet, the network traffic obeys the Poisson distribution, then we can construct a mathematic model of Internet with Poisson distribution function. However, the traffic in SDN-based IoT network is different from the traffic in Internet, although the packet of devices in IoT network is relatively small, and the data volume is relatively large, so we need to analyze the flow traffic characteristic in IoT network.

The discovery of the self-similarity of flow traffic has led people to find the theoretical reasons for the actual network performance and the theoretical performance using the traditional Poisson model. First, analyze the causes of self-similarity. Aiming at the propagation characteristics of self-similar business between sensor nodes in the Internet of Things, the self-similarity analysis of the business after the network researches self-similar traffic entering the Internet of Things network through the access point, etc. Through these analyses, to comprehensively describe the business characteristics of the Internet of Things. Finally, analyze and study the impact of business self-similarity on the derived performance of network nodes and on the quality of network service.

The scale of networks has grown as the telecommunications business has grown, and user data needs have emerged; different sorts of services, such as voice data, video, and formal streaming media, have distinct features. IP addresses are used in a variety of business streams on the Internet. The expected result is quite different from the real situation when the Poisson service model of the data service flow in the network is utilized in the data transmission network. The enormity of the network and the intricacy of the service in this situation can no longer be evaluated through the lens of a single call. The establishment cannot be determined by making multiple calls; rather, it must be determined from a broad perspective, depending on the data flow method and rate. Data on Internet communication, video, and VBR traffic was thoroughly examined, and the following key results were reached:

(1) Business flow can be self-similar at any moment and in any network environment.
(2) Self-similarity is common in the IP network environment, whether at the network layer, transport layer, or application layer, and it is linked to certain applications and protocol networks.

(3) Network performance is influenced by self-similarity. The notion of business self-similarity, as well as an examination of the sources of similarity, as well as an examination of network performance based on self-similarity, estimation, optimization, necessity, and possibility.

The local features of the curve seem to have the same similar performance on different space or time scales. Self-similarity refers to network traffic that has the same burstiness and similarity in different time scales and different locations. When the business flow in the network is increasing rapidly, the suddenness of the flow growth is easy to be noticed. The larger the number of samples, the more it can reflect the characteristics of the network traffic, and there are many similarities.

The flow traffic in IoT network is a generalized stationary random process $\mathbf{X} = \{X_n, n = 1, 2, 3, ...\}$, where X_n represents the k-th network traffic entities (such as data packets, bytes, bits). $N[t]$ is the number of the network traffic arrived between the time period $[0, t]$, and can be written as

$$N(t) = \int_0^t dN\tau \tag{1}$$

where $dN\tau$ is the process of each packets arrived.

The mathematical expectation and variance of a stationary time series X_n are

$$\mu = E[(X_n)] \tag{2}$$

$$\sigma^2 = E[(X_n - \mu)^2] \tag{3}$$

The autocorrelation function is

$$r(k) = \frac{E[(X_n - \mu)(X_{n+k} - \mu)]}{\sigma^2} \tag{4}$$

The random process obtained after time aggregation with a block size of m has the same autocorrelation function as the original random process. When the condition $k \to \infty$, so

$$\lim_{k \to \infty} r(k) = H(2H - 1)k^{2H-2} \tag{5}$$

where $H \in (0.5, 1)$ is the Hurst parameter [10]. The actual network traffic is positively correlated, so the value range of H is in the range (0.5, 1). The larger the H value, the higher the degree of self-similarity of the process, the slower the variance reduction rate of the random process after time averaging, and the stronger the correlation of the business flow. When $H = 0.5$, the self-similar process degenerates into a Poisson process. The transmission of the flow in the IoT network has short correlation, that is, the data volume of the packet in the arrival time interval is self-similar.

As we all know that the network traffic is the rate of packets/bytes/bites arrived, so it has the rate change and can be regarded as the acceleration of the flow. So, we use the Nosie Estimation Kalman Filtering (NEKF) to predict the network traffic. NEKF overcomes the limitation that the state transition matrix in the classic Kalman filter model needs to be reversible, and expands the application range of the Kalman filter. Using the statistical characteristics of noise estimation, open source to achieve higher prediction accuracy. The core function of Kalman filter is that

$$\mathbf{X}_k = \mathbf{K}_k \cdot \mathbf{Z}_k + (1 - \mathbf{K}_k) \cdot \mathbf{X}_{k-1} \tag{6}$$

where $\mathbf{K}_k$ is the k-th Kalman Gain. $\mathbf{Z}_k$ is the measurement results and $\mathbf{X}_{k-1}$ is the (k − 1)-th estimation results. The state equation and observation equation of Kalman filter can be written as follows

$$\mathbf{X}_k = \mathbf{F}_k \cdot \mathbf{X}_{k-1} + \mathbf{w}_{k-1} \tag{7}$$

$$\mathbf{Y}_k = \mathbf{H}\mathbf{X}_k + \mathbf{v}_k \tag{8}$$

where Eq. (7) represents the state equation, $\mathbf{X}_k$ is the $M \times 1$ dimensional state vector, which is unobservable; $\mathbf{F}_k$ is the $M \times M$ dimensional state transition matrix, which is used Describe the transition of the system from the state at time $k-1$ to the state at time k; $\mathbf{w}_{k-1}$ is the $M \times 1$ dimensional process error vector. Equation (8) represents the observation equation, $\mathbf{Y}_k$ is an $N \times 1$ dimensional observation vector; $\mathbf{H}$ is an $N \times M$ dimensional observation matrix, and the unobservable state vector $\mathbf{X}_k$ undergoes the action of $\mathbf{H}$, It becomes the observation vector $\mathbf{Y}_k$; $\mathbf{v}_k$ is the $N \times 1$ dimensional observation error vector.

Assuming that the observation time interval is T, the problem of concern is to predict the traffic arrival volume in the next T based on the current and previous observations. Let x_k denote the amount of traffic arriving at time n, and use x'_{k-1} to denote the first-order differential of x_k. When the observation time interval T is not too large, the relationship can be obtained using the first-order difference equation:

$$x_k = x_{k-1} + Tx'_{k-1} \tag{9}$$

where x'_k is the rate of flow change, namely the flow rate. In the same way, we can get that

$$x'_k = x'_{k-1} + Tx''_{k-1} \tag{10}$$

where x'_{k-1} is the rate of flow change.

There are many reasons can lead to changes in the rate of flow traffic at the node, such as the randomness of channel status, the uncertainty of terminal access, the distribution of service duration, the role of protocols, and various interferences. Consider these factors as "noise" with unknown statistical characteristics that cause the flow velocity change, and set it as w_k, so we can get that

$$x''_k = w_k \tag{11}$$

Then, we combine Eqs. (8)–(10) into a vector, then

$$\mathbf{X}_k = \begin{bmatrix} x_k \\ x'_k \\ x''_k \end{bmatrix} \tag{12}$$

The Eq. (6) can be written as

$$\mathbf{X}_k = \mathbf{F}_k \mathbf{X}_{k-1} + \mathbf{w}_k \tag{13}$$

where $\mathbf{F}_k$ is the state transition matrix (as $\mathbf{F}$), namely

$$\mathbf{F}_k = \mathbf{F} = \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & T \\ 0 & 0 & 0 \end{bmatrix} \tag{14}$$

w_k is the state noise vector, and it can be expressed as

$$\mathbf{w}_k = \begin{bmatrix} 0 \\ 0 \\ w_k \end{bmatrix} \tag{15}$$

The observation equation can be expressed as

$$y_k = \mathbf{H}\mathbf{X}_k + v_k \tag{16}$$

where $\mathbf{H}$ is the observation matrix, $\mathbf{H} = [1\,0\,0]$; v_k represents the observation noise. Note that $\mathbf{H}$ is 1×3-dimensional, and $\mathbf{X}_k$ is 3×1-dimensional, so the matrix form $\mathbf{Y}_k$ in Eq. (7) becomes the scalar form y_k.

2.2 Estimation Model

The traffic noise in IoT network obeys Gaussian distribution with zero mean, namely $G(0, \sigma)$. In multi-dimensional network traffic forecasting, to comprehensively consider the degree of deviation of each dimension from its mean, it is necessary to introduce a covariance matrix. $\mathbf{w}_k$ and $\mathbf{v}_k$ are the noise variables. The means of $\mathbf{w}_k$ and $\mathbf{v}_k$ are both 0, and the covariance matrix of $\mathbf{w}_k$ and $\mathbf{v}_k$ are $\mathbf{Q}_k$ and $\mathbf{R}_k$, respectivley. The covariance matrix estimation $\mathbf{P}_{k,k-1}$ at k-th can be expressed at that

$$\mathbf{P}_{k,k-1} = \mathbf{F}_k \mathbf{P}_{k-1,k-1} \mathbf{F}_k^T + \mathbf{Q}_k \tag{17}$$

Then, the Kalman Gain can be written as that

$$\mathbf{K}_k = \frac{\mathbf{P}_{k,k-1}\mathbf{H}_k^T}{\mathbf{R}_k + \mathbf{H}_k \mathbf{P}_{k,k-1} \mathbf{H}_k^T} \tag{18}$$

The estimation results can be written as that

$$\mathbf{X}_k = \mathbf{F}_k \cdot \mathbf{X}_{k-1} + \mathbf{K}_k[\mathbf{Y}_k - \mathbf{H}_k \mathbf{X}_{k-1}] \tag{19}$$

The updated minimum mean square error matrix is that

$$\mathbf{P}_{k,k} = [1 - \mathbf{K}_k \mathbf{H}_k]\mathbf{P}_{k,k-1}\beta \tag{20}$$

Base on the above analysis, we proposed an algorithm (NEKF) to measure and estimate network traffic in IoT network, the process of the algorithm as follows:

Step 1: Measuring the network traffic in the IoT network, and construct the traffic matrix $\mathbf{X}$; Initialing the noise error $\mathbf{w}_k$, $\mathbf{v}_k$ and β; construct the transfer matrix $\mathbf{F}_k$ and observation matrix $\mathbf{H}$.
Step 2: Calculate the covariance matrix $\mathbf{Q}_k$ and $\mathbf{R}_k$; Then, with the Eq. (16) to calculate the covariance matrix estimation $\mathbf{P}_{k,k-1}$;
Step 3: Based on the Eq. (17), calculating the Kalman Gain $\mathbf{K}_k$;
Step 4: With the Eqs. (17) and (18) to calculate the estimation results $\mathbf{X}_k$ and update the minimum mean square error matrix $\mathbf{P}_{k,k}$;
Step 5: Go back to step 2 until the maximum repeat times.

3 Simulation Result and Analysis

3.1 Simulation and the Simulation Metrics

In order to verify the algorithm proposed in this article, we built an SDN test platform based on the Ryu controller and Mininet network simulation tool, and wrote the network traffic measurement and prediction module in the Ryu controller with python language. Absolute error (AE) and relative error (RE) are two commonly used indicators to measure network traffic estimation errors. The AE of the flow rate reflects the deviation between the actual flow rate and the measurement result. RE is the ratio of AE to the actual traffic value, which reflects the reliability of the measurement. AE and RE can be expressed as

$$\mathrm{A}E_i = \left|x_i - \hat{x}_i\right| \tag{21}$$

$$RE_i = \left|x_i - \hat{x}_i\right| / x_i \tag{22}$$

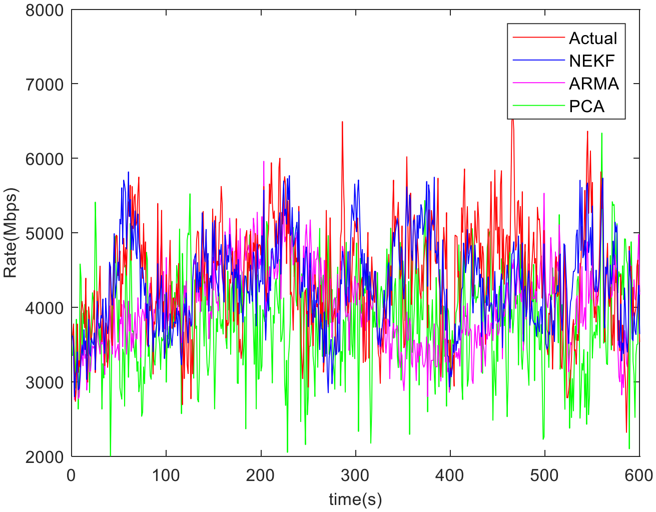

Fig. 1. The network traffic of actual flow and three traffic estimation results. (Color figure online)

3.2 Simulation Evaluation

We use Iperf to simulate network traffic in the network and randomly select a business flow for analysis. Traffic in the network. In this article, we use the traffic model (NEKF) mentioned in this article to estimate and optimize network traffic, and compare and analyze it with real network traffic, ARMA and PCA and other network traffic estimation methods.

Figure 1 shows the results of network traffic measurement and estimation with a sampling interval of 1 s in 10 min. The red line uses the real traffic value generated by the simulation tool, while the blue line, pink line and green line represent the results of network traffic measurement and estimation by NEKF, ARMA, and PCA, respectively. From the figure, we can see that the three network traffic measurement and estimation methods can basically reflect the trend of traffic changes in the network as a whole. It can be seen from the whole that our proposed algorithm can still more accurately reflect the changes in network traffic compared with the other two methods.

To reflect the accuracy of network traffic estimation more accurately, we use absolute error and relative error to further analyze the network traffic. Figure 2 shows the absolute error of the network flow estimation error of several different network flow estimation methods. We also find that the error of our proposed method is relatively small compared with the other two network traffic estimation methods, and the average is less than 500 Mbps.

Figure 3 shows the CDF distribution diagram of the absolute error of network traffic. From the figure, it can be seen that 80% of the network traffic error of our proposed algorithm is less than 720 Mbps, while the other two algorithms are less than 1080 Mbps and 1470 Mbps, respectively, which fully demonstrates the stability and the accuracy of network traffic estimation of our proposed methods. In Fig. 3, we also find that the

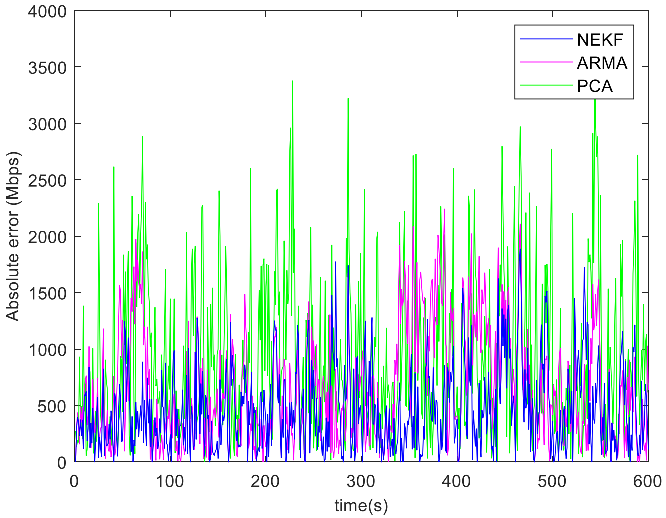

Fig. 2. The AE of three network traffic estimation results.

slope of NEKF is the largest, that is, the probability distribution of absolute error of NEKF is very concentrated, which shows that the performance of our proposed method is relatively stable.

Figure 4 shows the distribution of the relative error of network traffic with the boxplot. T The red line in the box represents the median value of RE ranking, while the upper and lower edge distributions of the blue box represent the distribution points of RE at 75% and 25%. Then, compare with the other two methods, the median value of RE of

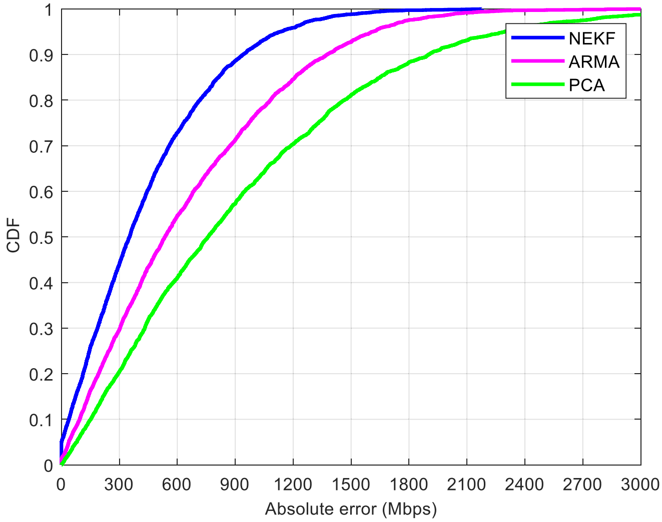

Fig. 3. The CDF of absolute errors of three network traffic estimation results.

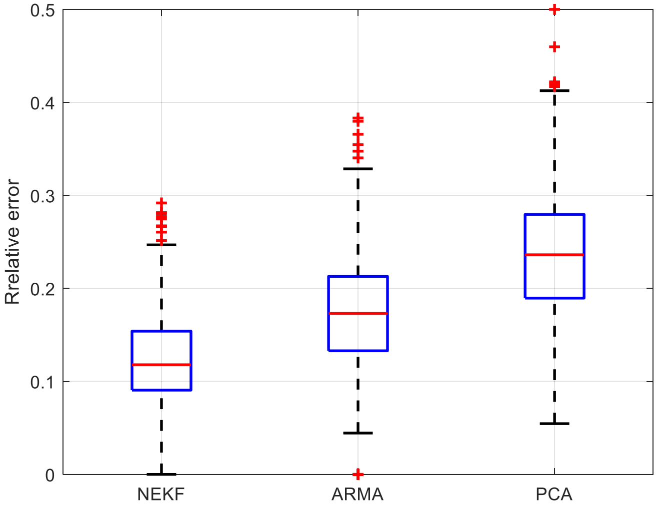

Fig. 4. The mean and variance of relative errors. (Color figure online)

our proposed method is the smallest, and the value is 0.118, namely the RE of traffic estimation results is about 11.8%, and the RE of the traffic estimation results is about 17.3% and 23.6%, respectively. The black line represents the distribution area of most of the measured values. It can be seen from the figure that the black interval of our proposed algorithm is smaller than that of the other two methods, which fully illustrates the stability of our proposed algorithm. It can be seen that our proposed algorithm is not only stable, but also has high network traffic estimation accuracy. The RE of traffic results is about 11%, which has strong usability.

4 Conclusions

As the basic network for information collection, the Internet of Things allows millions of electronic devices to be connected to the network. The traditional architecture of the SDN-based Internet of Things network has strong flexibility and strong scalability, and can allocate resources to users in a personalized manner and perform network management and scheduling according to user needs. In order to better manage the network, the measurement and estimation of network traffic is the basis to ensure the efficient prototype of the network. In order to overcome the high cost of traditional measurement methods and poor deployment flexibility, we propose a novel network traffic measurement and measurement method in this article.

By measuring the network traffic, the network traffic is estimated, so as to obtain the dynamic change trend of the network traffic, and provide data support for the efficient management of the network. Finally, we built a network simulation verification platform to analyze the performance of the algorithm proposed in this paper. The simulation results show that our proposed measurement approach is feasible and effective.

References

1. Memos, V., Psannis, K., Ishibashi, Y., et al.: An efficient algorithm for media-based surveillance system (EAMSuS) in IoT smart city framework. Futur. Gener. Comput. Syst. **83**(4), 619–628 (2018)
2. Hossain, M., Muhammad, G., Abdul, W., et al.: Cloud-assisted secure video transmission and sharing framework for smart cities. Futur. Gener. Comput. Syst. **83**, 596–606 (2018)
3. Ali, I., Gani, A., Ahmedy, I., et al.: Data collection in smart communities using sensor cloud: recent advances, taxonomy, and future research directions. IEEE Commun. Mag. **56**(7), 192–197 (2018)
4. Suarez-Varela, J., Barlet-Ros, P.: Towards a NetFlow implementation for OpenFlow software-defined networks. In: Proceedings ITC 2017, vol. 1, pp. 187–195 (2017)
5. Huang, L., Zhi, X., Gao, Q., et al. Design and implementation of multicast routing system over SDN and sFlow. In: Proceedings ICCSN 2016, pp. 524–529 (2016)
6. Yu, C., Lumezanu, C., Zhang, Y., Singh, V., Jiang, G., Madhyastha, H.V.: FlowSense: monitoring network utilization with zero measurement cost. In: Roughan, M., Chang, R. (eds.) PAM 2013. LNCS, vol. 7799, pp. 31–41. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36516-4_4
7. Xu, H., Zong, X., Su, J., et al.: Formalization of SNMP messages using composite-elements based on extenics for software-defined networking. In: Proceedings the 9th International Conference on Communication Software and Networks, May 2017, pp. 989–992 (2017)

8. Jiang, D.D., Huo, L.W., Li, Y.: Fine-granularity inference and estimations to network traffic for SDN. PLoS ONE **13**(5), 1–23 (2018)
9. Jiang, D.D., Huo, L.W., Lv, Z.H., et al.: A joint multi-criteria utility-based network selection approach for vehicle-to-infrastructure networking. IEEE Trans. Intell. Transp. Syst. **19**(10), 3305–3319 (2018)
10. Kermarrec, G.: On estimating the Hurst parameter from least-squares residuals. Case study: correlated terrestrial laser scanner range noise. Mathematics **8**(5), 1–23 (2020)
11. Wang, J., Wen, R., Li, J., et al.: Detecting and mitigating target link-flooding attacks using SDN. IEEE Trans. Dependable Secur. Comput. **1**, 1–10 (2018)