A Network Traffic Measurement Approach for Edge Computing Networks

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

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

Abstract. Edge computing is one of the key technologies in 5G networks, it can collect and process data on the access network and decrease the transmission load of the network. The data exchange in the Edge computing network Software Defined Networking (SDN) decouples the control plane and forwarding plane in traditional switches and plans to route in the global view, making network management more flexible and efficient. The accurate and comprehensive network traffic measurement is the key to traffic management of edge computing networks. Then, we propose a novel edge computing network traffic measurement approach to SDN. The proposed measurement methods use the in SDN by collecting statistics in OpenFlow-based switch and utilize the LSTM model and GNN method to infer the fine-grained measurement. Then, we construct an objective function to optimize the estimation results. Finally, we conduct a series of simulations to evaluate the performance of the proposed scheme. Simulation results show that our approach is feasible and has low measurement cost.

Keywords: Edge computing · Internet of things · Software defined networking

1 Introduction

Smart cities use billions or even trillions of devices to collect various information in urban areas, such as transportation, power plants, water supply networks, schools, and hospitals [1, 2]. The data of the smart city is generated by a large number of distributed devices and transmitted to the cloud computing server through the network. This information is used to improve the management and utilization efficiency of urban resources and improve the quality of life of residents [3]. In the traditional Internet of Things architecture, smart city data is integrated and processed in cloud computing. The cloud computing center consists of hundreds of thousands of servers, allowing users to use various terminals to access application services at any location. The storage location of the requested resource in the cloud is not fixed. Cloud computing is a general computing platform that

D. Jiang and H. Song (Eds.): SIMUtools 2021, LNICST 424, pp. 29–37, 2022.
https://doi.org/10.1007/978-3-030-97124-3_3

has powerful computing and storage capabilities and supports simultaneous operation of different applications [2]. The scale of the cloud can be dynamically expanded to meet the growing needs of applications and users, and cloud resources can be sold according to customer needs. In the IoT architecture of smart cities, cloud computing collects and centrally processes a large amount of monitoring data from distributed devices. Therefore, citizens can obtain more comprehensive and effective information from the cloud. Although the data volume of each device is small, a large number of devices simultaneously transmit data to cloud computing, which will cause congestion and delay, and pose a major challenge to the fast transmission capability of cloud computing servers, data processing.

Chen et al. [4] Designed a scalable, accurate, and fast measurement scheme in SDN using packet-level statistical data, and proposed a low-latency load-aware two-layer measurement platform that can estimate link utilization. Aslan et al. [5] studied the influence of active and passive network state collection methods in SDN. Liu et al. [6] shows a flow measurement and reasoning framework that performs adaptive measurement through online learning. Shu et al. [7] proposed a reference framework for traffic engineering in SDN, and proposed a framework for traffic measurement. These programs try to use estimation methods to measure network traffic, however, measurement errors are not unpredictable. Jiang et al. [8, 9] studied the characteristics of network traffic in wireless access networks.

Based on the above analysis, we have considered pull-based and flow-based network traffic measurement in the edge computing architecture. In this article, we propose a low-overhead measurement scheme and build a new edge computing network measurement architecture. We directly measure some data of network traffic and predict fine-grained network traffic. Then, we propose an objective optimization model to reduce the fine-grained measurement error of reasoning, and propose a heuristic algorithm to find the optimal solution of the model. Our main contributions in this article are as follows:

(1) The measurement overhead and accuracy of flows as the core factor in the edge computing network. To obtain the measurement results with low overhead and high accuracy, we propose to measure the coarse-grained traffic of flows and fine-grained traffic of links.
(2) Multiple decision factors, including bandwidth, link load and flow conservation principle, are jointly considered in networking, an objective function is proposed to decrease the estimate errors.
(3) We present a LSTM and GNN algorithm to obtain the optimal solution of the fine-grained measurement, and conduct some simulations to verify the validate the proposed measurement scheme.

The rest of this paper is organized as follows. Section 2 presents a novel lightweight measurement architecture of edge computing and analyzes the traffic matrix. We propose a coarse-grained measurement and optimization to measure the fine-grained traffic of the edge computing network. Section 3, we use a simulation to verify the novel measurement architecture and the performance of the proposed method and then analyze simulation results. Finally, Sect. 4 is the conclusion of our work.

2 Problem Statement

Edge computing frequently requests the resource from cloud computing to decrease the network. Network measurements such as load balancing, path planning, and anomaly detection are required, and SDN is based on traffic measurement is easier and more flexible than traditional networks.

2.1 Traffic Matrix Construction

OD (Origin-Destination) traffic refers to the amount of data transmitted from any node to another node in the network, and it describes the dynamic changes of data transmitted between the source and the destination. In the edge-computing network, flow traffic is dynamically changing and can be expressed as

$$\mathbf{x} = \{x_1, x_2, ..., x_t, ...\} \tag{1}$$

The flow traffic in the network is very important for the network management, but it is difficult to measure the traffic of each flow in the edge computing network. However, the link load in the network can be directly measured through SNMP(simple network management protocol). Therefore, we use the network traffic matrix to invert the network traffic in the edge-computing network. The relationship between flow traffic and link load can be expressed as a linear equation that

$$[y_1, y_2, ..., y_n]^T = [a_{ij}]_{n\times k}[x_1, x_2, ...x_k]^T \tag{2}$$

where $[y_1, y_2, ..., y_n]^T$ is a column vector representing link traffic, $[x_1, x_2, ...x_k]^T$ is also a column vector representing the traffic matrix and $[a_{ij}]_{n\times k}$ is the routing matrix, $i \in \{1, 2, ..., n\}$ $i \in \{1, 2, ..., n\}$, and $a_{ij} \in \{0, 1\}$.

The network traffic flow forecasting problem can be regarded as describing the non-linear mapping function f that maps historical traffic data to future traffic data.

The flows in the edge computing network have periodic changes in both time and space dimensions. To obtain complex spatial and time dependence is an important issue for network traffic inversion. In a non-dynamic network topology, the device nodes in the network constitute a point set, and the connections between nodes constitute a combination of edges. Therefore, the network topology can be expressed as a graph composed of points and edges $G = (N_p, E_p)$, where $N_p = \{n_1, n_2, ..., n_{NP}\}$, $E_p = \{e_1, e_2, ..., e_{EP}\}$.Using traditional convolutional neural networks (CNN) can obtain local spatial features, but CNN is limited to processing European data (such as images, speech, etc.). However, the network topology exhibits a non-Euclidean topology, which indicates that the CNN model cannot directly represent the complex topology of the network, that is, it cannot correctly capture the spatial correlation of traffic in the network topology. GCN (Graph convolutional network) is widely used to extract the spatial correlation of graph-based data.

Many researchers only use the first few time intervals (usually a few minutes or hours) to predict the network traffic. These methods ignore long-term correlations (such as periodicity), and periodicity is regarded as an important feature of spatiotemporal forecasting problems. Edge computing network data exhibits periodic changes in temporal and spatial correlation, considering not only short-term information, but also long-term periodic information. After extracting the spatial characteristics of the data, LSTM is usually used to obtain the time series dependency. Edge computing network traffic data is constantly changing with time and space, showing strong uncertainty and complexity. Therefore, in the measurement and prediction of edge computing network traffic, it is necessary to consider the impact of these complexity and uncertainty on the prediction results.

$$h_{i,t} = LSTM([x_{i,t}; e_{i,t}], h_{i,t-1}) \tag{3}$$

where $h_{i,t}$ is the network traffic i at time t. $e_{i,t}$ is the external influence variable. The $h_{i,t}$ contains spatial and short-term time information. This method only uses historical time series fragments adjacent to the forecast period, because the traffic data of a node at the previous moment will inevitably have a greater impact on the traffic at the next moment. This kind of network only uses the most recent time interval. To make better long-term predictions, we need consider the periodic information. Since people's lives and work are usually regular, usually everyone's living habits and work content are relatively fixed, and have a certain degree of repetition, which makes network business traffic also have time characteristics. In addition, different types of business flows occur during working hours or during breaks, and they are located in different spatial locations during working hours and off hours. That is, the network traffic data also has an obvious weekly cycle pattern. We can find that the network traffic pattern of the previous Monday is usually similar to the network traffic pattern of the previous Monday, but is slightly different from the network traffic pattern of the weekend.

Training LSTM to handle long-term information is a difficult task. As the length of the time series increases, the periodic effects reduce significantly. To solve this problem, the relative time period of the forecast target should be modeled. However, it is not enough to consider only the relative time period, because it ignores the traceability of time changes, that is, the data of network traffic is periodic. During daily working hours, the traffic on the network will suddenly increase, and during the rest period, the network traffic will decrease, which shows that network traffic sequence changes periodically.

Therefore, we designed a periodic attention mechanism to constrain the predicted value of network traffic. For daily time changes, these times can solve potential time changes. Using LSTM to process daily sequence information, the formula as shown

$$h_{i,t}^{p,q} = LSTM([x_{i,t}^{p,q}; e_{i,t}^{p,q}], h_{i,t}^{p,q-1}) \tag{4}$$

where $h_{i,t}^{p,q}$ is the network traffic i at time t at precious p days at time period q. $e_{i,t}^{p,q}$ is the external influence variable. The $h_{i,t}$ contains spatial and short-term time information. We use LLAMM as our proposed network traffic measurement methods.

Not every previous network traffic measurement has the same contribution to the target prediction. Then, we introduce the attention mechanism to capture the changes in time and obtain a weighted representation of each period in the previous measurement. The representation of each period of the previous measurement is the weighted sum of each selected period, and its definition as shown

$$h_{i,t}^{p} = \sum_{q \in Q} \alpha_{i,t}^{p,q} h_{i,t}^{p,q} \tag{5}$$

where $\alpha_{i,t}^{p,q}$ is the weighted factor which used to represent the importance of p days at time period q. $\alpha_{i,t}^{p,q}$ is obtained by comparing the space-time representation obtained from LSTM with the previous hidden state $h_{i,t}^{p,q}$, and its calculation uses the attention mechanism, and its calculation as shown that

$$\alpha_{i,t}^{p,q} = \frac{\exp(score(h_{i,t}^{p,q}, h_{i,t}))}{\sum_{q \in Q} \exp(score(h_{i,t}^{p,q}, h_{i,t}))} \tag{6}$$

The definition of attention score can be viewed as a content-based function that

$$score(h_{i,t}^{p,q}, h_{i,t}) = v^T \tanh(W_H h_{i,t}^{p,q} + W_X h_{i,t} + b_X) \tag{7}$$

where W_H, W_X, b_X, v are parameters and v^T is the transpose of v. For each previous periodic p, we get a period representing $h_{i,t}^{p}$. Then, we use another LSTM to use these period representations as the input and save the time sequence, as shown that

$$h_{i,t}^{p} = \mathrm{LSTM}(h_{i,t}^{p}, h_{i,t}^{p-1}) \tag{8}$$

The output $h_{i,t}^{p}$ of the last period as a representation of time dynamic similarity.

We will concatenate short-term dependence $h_{i,t}$ and long-term dependence, $h_{i,t}^{p}$ to obtain $h_{i,t}^{c}$, as shown in the formula that

$$h_{i,t}^{c} = [h_{i,t} : h_{i,t}^{p}] \tag{9}$$

where the symbol: means splicing. For the predicted link and time, both short-term and long-term dependence are retained. We input $h_{i,t}^{c}$ to the fully connected layer. To obtain the final predicted value of traffic flow for each link, and represent it as $y_{i,t+1}$, so the final prediction function is defined as follows:

$$\hat{y}_{i,t+1} = \tanh(W_{fa} h_{i,t}^{c} + b_{fa}) \tag{10}$$

where W_{fa} and b_{fa} are parameters. Due to the normalization operation, the output range of the model is (1,1). Then, the predicted value is normalized and reversed to bring it back to the actual range.

In the training phase, the goal is to minimize the error between the actual traffic flow on links and the predicted value. The loss function of the model can be written as

$$L = \sum_{i=1}^{n} \left\| \hat{y}_{i,t+1} - y_{i,t+1} \right\| + \lambda L_{reg} \tag{11}$$

where $\hat{y}_{i,t+1}$ and $y_{i,t+1}$ are the predicted result and actual network traffic, respectively. λ is a hyperparameter. The first term of formula (11) is used to minimize the error between the actual traffic flow and the predicted flow. The second term L_{reg} is the L2 regularization term, which can effectively prevent over-fitting.

3 Simulation Result and Analysis

3.1 Simulation Environment

In order to evaluate the performance of the measurement scheme proposed, we built an SDN test platform and wrote the measurement module in python. We import the measurement module programmed import into the controller and call the Periodic Sampling module to send a Read-state message to switches via OpenFlow protocol. To verify the performance of the measurement scheme proposed, we introduce some common error evaluation metrics, such as the Relative Error (RE). The AE of the flow traffic reflects the deviations between the actual traffic and the measurement results. The RE is the ratio of measured absolute error to the actual value, reflecting the credibility of the measurement.

3.2 Simulation Evaluation

Figure 1 shows that the transmission rate of LLAMM is generally relatively stable, and is higher than the transmission rate of R60 and U60, and is closest to the transmission rate of the real network stream, indicating that the measurement performance of LLAMM is more stable.

It can be seen from Fig. 2 that the relative error produced by U60 at the beginning of the measurement is relatively large, and the relative error is relatively stable over time, and the average relative error of most network traffic measurement is about 0.2. The relative error of R60 is relatively small in the early stage, while the relative error of the later measurement is always high. Although the measurement relative error of LLAMM will suddenly increase at some moments. LLAMM still maintains a relatively stable relative error, and the measurement result of LLAMM is relatively stable and similar

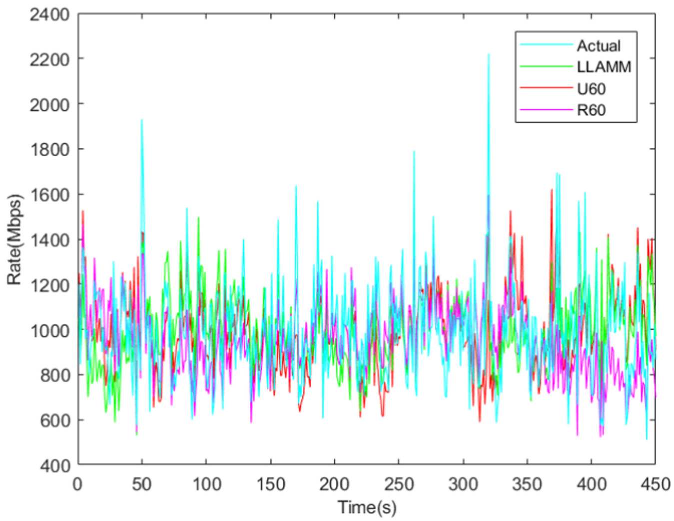

Fig. 1. The network traffic measurement of different methods.

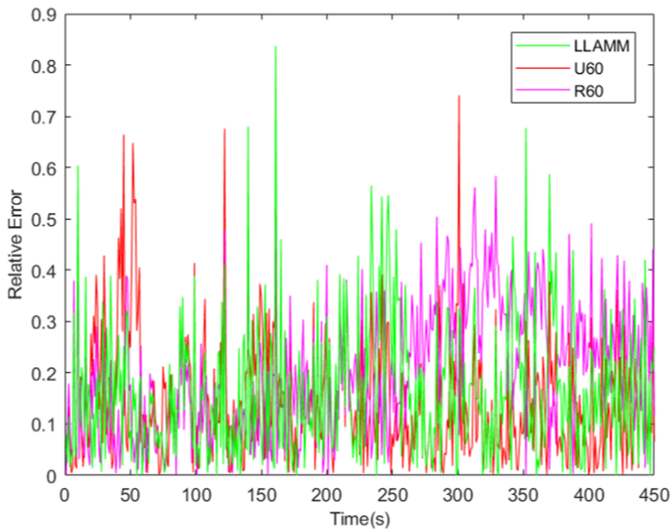

Fig. 2. The RE of network traffic measurement.

with the U60. This shows that our proposed method can continuously and steadily measure network traffic.

We introduce the cumulative distribution function (CDF) to describe the relative error more intuitively. Figure 3 shows the CDF of the relative error produced by different sampling schemes. The relative error is the ratio of the absolute error of the measurement to the actual value, which reflects the credibility of the measurement. It can be seen from

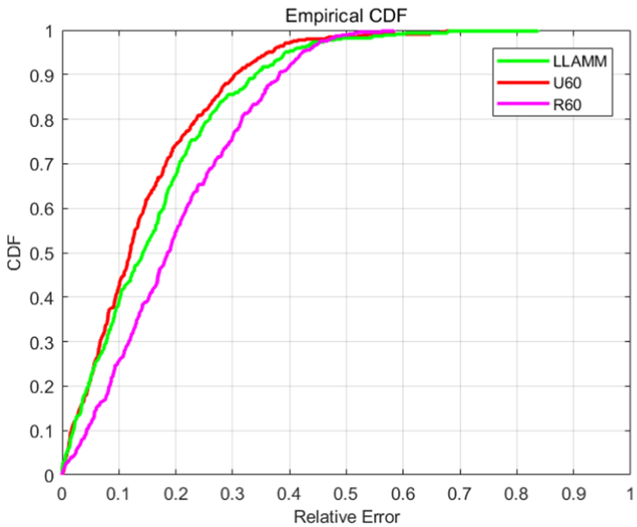

Fig.3. The CDF of RE of network traffic measurement.

Fig. 3 that the relative error of LLAMM and U60 is about 90% lower than 0.3, and R60 is even lower. Only about 75% of the relative error is lower than 0.3, indicating that the measurement of R60 is credible. The degree is not as high as LLAMM and U60. From the CDF curve of the relative error between LLAMM and U60, it can also be seen that the measurement accuracy of U60 is slightly better than that of LLAMM. However, it can be seen that the measurement overhead of U60 is abnormally high, resulting in the overall measurement performance is still not as good as LLAMM.

4 Conclusions

Fine-grained flow-based network measurement has a great impact on network traffic management of edge computing networks. In this paper, we propose a novel measurement method for flow traffic of SDN. The proposed scheme has two main stages. The first stage, the sampling method is used to collect the statistics information of links and flows in OpenFlow-based switches to obtain the coarse-grained measurement; the second stage performs the LSTM model to predict the network traffic and utilize the LLAMM method to optimize the fine-grained estimation results to decrease the measurement errors. Finally, the simulation results show that the proposed measurement method is feasible and effective.

References

1. Memos, V.A., Psannis, K.E., 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.S., 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. Chen, M.H., Tien, Y.C., Huang, Y.T., et al.: A low-latency two-tier measurement and control platform for commodity SDN. IEEE Commun. Mag. **54**(9), 98–104 (2016)
5. Aslan, M., Matrawy, A.: On the impact of network state collection on the performance of SDN applications. IEEE Commun. Lett. **20**(1), 5–8 (2016)
6. Liu, C., Malboubi, A., Chuah, C.N.: OpenMeasure: adaptive flow measurement and inference with online learning in SDN. In: Proceedings of the INFOCOM 2016, pp. 47–52 (2016)
7. Shu, Z.G., Wan, J.F., Wang, S.Y., et al.: Traffic engineering in software-defined networking: measurement and management. IEEE Access **4**, 3246–3256 (2016)
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. Pappas, S.S., Ekonomou, L., Karamousantas, D.C., et al.: Electricity demand loads modeling using AutoRegressive Moving Average (ARMA) models. Energy **33**(9), 1353–1360 (2008)
11. Xu, C., Li, Z., Wang, W.: Short-term traffic flow prediction using a methodology based on autoregressive integrated moving average and genetic programming. Transport **31**(3), 343–358 (2016)

12. Li, X.L., Qian, J.X.: Studies on artificial fish swarm optimization algorithm based on decomposition and coordination techniques. J. Circ. Syst. (2003)
13. Roughan, M., Zhang, Y., Willinger, W., et al.: Spatio-temporal compressive sensing and internet traffic matrices. IEEE/ACM Trans. Netw. (ToN) **20**(3), 662–676 (2012)