Mobile Networks and Applications (2021) 26:716–725
https://doi.org/10.1007/s11036-019-01423-3

A New Traffic Prediction Algorithm to Software Defined Networking

Yuqing Wang[1] · Dingde Jiang[1] · Liuwei Huo[2] · Yong Zhao[2]

Published online: 20 December 2019

Abstract

Traffic prediction is significantly important for performance analysis and network planning in Software Defined Networking (SDN). However, to effectively predict network traffic in current networks is very difficult and nearly prohibitive. As a new cutting-edge network technology, SDN decouples the control and data planes of network switch devices to enable the flexibility of network measurements and managements. The SDN architecture of the flow-based forwarding idea brings forth a promising of network traffic capture and prediction. We propose a lightweight traffic prediction algorithm for SDN applications. Firstly, different from traditional network traffic measurements, our method uses the flow-based forwarding idea in SDN to extract traffic statistic from data plane. The statistical variable describes network flow information forwarded in SDN and enables more accurate measurements of flow traffic via a direct and low-overhead way compared with traditional traffic measurements. Secondly, based on the temporal nature of traffic, the time-correlation theory is utilized to model flow traffic, where the time-series analysis theory and regressive modeling approach are used to characterize network traffic in SDN. A fully new method is proposed to perform traffic prediction. Thirdly, we propose the flow-based forwarding traffic prediction algorithm to forecast to SDN traffic. The detailed algorithm process is discussed and analyze. Finally, sufficient experiments are presented and designed to validate the proposed method. Simulation results show that our method is feasible and effective.

Keywords Software defined networking · Regressive model · Traffic prediction · Simulation analysis · Network measurements

1 Introduction

With the rapid development of the Internet and the explosive growth of network traffic, the traditional network architecture based on TCP/IP protocol has exposed various problems [1], such as scalability, controllability, security, mobility, Quality of Service (QoS) and green energy saving etc. In traditional networks, the traffic prediction is an important research direction. It is important for network performance analysis and network planning [2]. However, the traditional network is a distributed network, and it cannot be flexibly controlled, which leads to the traffic prediction algorithm cannot be applied to industry very well. In such a network environment, we need to find a more efficient and flexible network deployment model to adapt to more and more flexible network requirements, reduce network complexity, and accelerate the pace of network innovation. So, the concept of Software Defined Networking (SDN) is proposed and gradually recognized [3, 4]. SDN is a new type of network architecture [5]. The feature of the architecture is that the data plane is separated from the control plane and can be directly controlled by the program [6]. In the traditional TCP/IP network architecture, the traffic control layer and data layer are tightly coupled a switch or router. However, with the core technology OpenFlow, SDN extracts the control plane from the network device and makes a separate controller for centralized network control, which reduces the complexity of the network.

SDN Traffic prediction is very important for network performance analysis and network planning. Through traffic prediction, traffic trends can be known in advance, so that traffic can be controlled or adjusted in time to achieve load balancing, avoid network congestion, and improve network performance. At the same time, it guides network planning and design, which makes it easy to manage network resource. However, because the data traffic of the SDN network has the characteristics of fractal and paroxysm, the modeling and analysis of the SDN network traffic is also complicated and difficult. Traditional prediction models are no longer suitable for SDN. Based on the long correlation and self-similarity of

✉ Dingde Jiang
merry_99@sina.com

1 School of Astronautics and Aeronautic, University of Electronic Science and Technology of China, Chengdu 611731, China

2 School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China

traffic data, we propose a flow modeling and prediction method based on autoregressive moving average (ARMA) model. With the ARMA model, the prediction calculation is small and can be calculated in real time [7, 8].

We make the following contributions:

- We describe in detail the principles of network traffic prediction and the SDN traffic measurement process.
- We introduce an equally spaced sampling algorithm and use the sampling algorithm to obtain a flow sequence. Then we propose a lightweight traffic prediction algorithm for SDN applications.
- We use time series analysis theory and regression modeling method to characterize SDN network traffic and propose a traffic prediction method. The simulation results show the feasibility and effectiveness of the proposed method.

The remainder of this paper is organized as follows: Related work is discussed in section 2. The process of SDN traffic measurement, the establishment of the ARMA model, and the process of traffic prediction are described in section 3. Section 4 is simulation and results analysis. And the paper is concluded by section 5.

2 Related Work

The research of traffic prediction method is changing with the change of network scale and network application. The development of its prediction method has gone through four stages: traditional model, self-similar model, intelligent algorithm and combined model.

In order to accurately estimate the network traffic in the large-scale IP network, Jiang et al. proposed the BPNN estimation in the large-scale IP network [9]. Li et al. proposed a model that combines wavelet transform with artificial neural network to significantly reduce the prediction error and improve the prediction accuracy [10]. To improve the setting method of the parameter and network structure of neural network, Wei proposed a prediction model based on RBF neural network optimized by improved gravitation search algorithm [11]. In order to improve the quality of service of cellular networks, Zhang et al. proposed the Spatial-Temporal Cross-domain neural Network (STCNet) [12]. To adapt to the suddenness of network traffic and improve prediction accuracy, Lu et al. proposed an Long Short-Term Memory (LSTM) model based on the wavelet transform theory [13].

With the development of SDN, researchers have shifted the research focus from traditional network to SDN. Therefore, the research of traffic prediction based on SDN is also widely concerned by researchers. In the traditional TCP/IP network, the distributed network architecture makes the network less flexible and intelligent, which leads to the poor application of traffic prediction. In SDN, the control plane is extracted to be a controller, and the centralized control of the network is realized. Therefore, the deployment and application of traffic prediction algorithm has been well applied in SDN.

In recent years, with the rapid development of artificial intelligence and machine learning, many of these algorithms have also been applied to SDN [14]. In order to accurately predict network traffic, Dai et al. proposed the EMD-based multi-model Prediction (EMD-MMP) for network prediction [15]. Traffic Matrix (TM) prediction is defined as the problem of estimating future network traffic matrix from the previous and achieved network traffic data. On the one hand, in order to improve the performance of TM prediction, Azzouni et al. presented NeuTM TM prediction based on Long Short-Term Memory Recurrent Neural Networks (LSTM RNNs) [16]. On the other hand, in order to reduce the resource consumption of TM estimation, Li et al. proposed the Online Information Gain Maximization (IGME) based SDN traffic matrix estimation method [17]. Furthermore, in order to overcome the limitations of the traditional model, effectively predict the controller's control channel traffic, Yu et al. build a seven-function model to predict control channel usage [18]. Our previous work can be found in [21–25].

3 Algorithm Design

3.1 Network Rraffic Prediction Principle

The network predicts the trend of traffic changes. According to the predicted results, the network load can be predicted in advance, and control or adjustment can be made in time to avoid network congestion and achieve load balancing. In turn, it greatly improves network performance and service quality. Therefore, using mathematical modeling theory and flow measurement technology to establish an appropriate model is of great significance for network performance improvement and network planning and design. This section will first describe the basic principles of network traffic prediction. Network traffic refers to the total amount of data transmitted over a network within a unit time. In the network traffic collection process, network traffic is usually collected at the same time interval, so that the time series traffic in time sequence is obtained.

Network traffic prediction refers to training the model based on historical traffic data collected and establishing an appropriate traffic model. Therefore, the future development trend of network traffic is predicted, which provides a basis for network planning and decision-making. The general prediction principle of network traffic is shown in Fig. 1.

In the network traffic prediction, the prediction model is a key link, and the result of model selection will directly affect

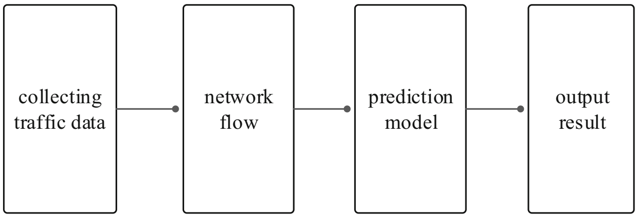

Fig. 1 The fundamental principle of traffic prediction

the final traffic prediction results. An appropriate network traffic prediction model can capture the characteristics of network traffic quickly and effectively. Conversely, if the selected model is not appropriate, it can not capture the characteristics of network traffic very well, and resulting in poor network performance because it misestimated the performance of the network. Network traffic is affected by a variety of complex external factors, and traffic data has characteristics of nonlinearity and multi-scale changing, so it is difficult to describe the network traffic and its influencing factors by accurate mathematical model.

3.2 SDN Traffic Measurement and Sampling Algorithm

3.2.1 SDN Traffic Measurement Process

To predict traffic, we firstly collect the flow statistics information required by simulation. According to the introduction of OpenFlow, we use the controller to send OpenFlow messages to query the information stored in the counters in the flow entries of the OpenFlow switch, and obtain the statistics of the flows. The counter records how many messages belonging to this flow have been received and other statistics data (such as the number of packets, the number of bytes and query time).

We studies the modeling and analysis for a network flow. So, we firstly should select the flow and establish a new flow table item for this flow, and then do real-time measurement. The real-time measurement [19] is the collection of statistics on the counter for this flow.

When a new flow arrives, because the new data packet group have no match in the flow table, the OpenFlow switch will package the Packet-in message to send to the controller. When packaging, if the local cache of the switch is large enough, the data packet is temporarily placed in the cache, and both the control information (128Byte) in the packet header and the serial number in the switch cache will be sent to the controller. If the switch does not support the local cache or the cache capacity is insufficient, the entire data packet will be encapsulated into the Packet-in message and sent to the controller. Then, when the controller receives the Packet-in message, it will send the Flow-mod message and send the packet again by the Packet-out message. The Flow-mod message

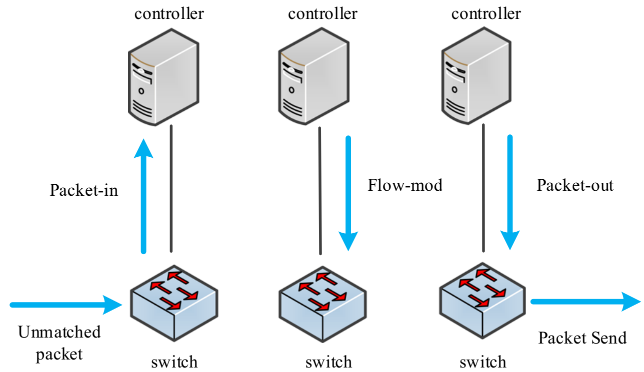

Fig. 2 The installation procedure of a new flow

gives the survival time of this flow table item. The flow creation process is shown in Fig. 2.

When a new flow table is established, it is time to carry out real-time measurement. The controller sends the OpenFlow message to the switch. The specific process is shown in Fig. 3. For example, the controller sends FlowStatisticsRequest to the switch to query the flow information. In response, the switch sends the FlowStatisticsReply message to the controller. The FlowStatisticsReply message contains the statistics information of flow counter. We further process the statistics of the traffic counters and then performed simulation analysis.

3.2.2 Equal Interval Sampling Algorithm

In the *nth* sampling period, selected sampling time is $t_n = nT + t_0$, where T represents the sampling interval, t_0 represents the initial time, in SDN, time is uses as a sampling element, for each time slice, it will be decided whether to sample it according to the sampling rules, if the time slice is sampled, the data packet groups that arrive at the time slice can be selected, the time slice can be expressed as $\{t_n : n = 0, 1, 2, \ldots\}$, the data packet group arrives at t_n will be sampled.

The pseudo code for the equal interval sampling algorithm is shown in Table 1.

For SDN traffic, after sampling at equal intervals, we get the time sequence as follows:

$$x = \{x(t)|t = 0, 1, 2, \ldots, n-1\} \tag{1}$$

where $x(t)$ stands for the amount of SDN traffic at time t and n represents the duration length of SDN traffic.

3.3 ARMA model parameter estimation

ARMA model is an important method for studying stochastic time series, and it is also a method of forecasting time series with high accuracy. It was proposed by Box and Jenkins, so it is also called the B-J method. The basic idea of ARMA model is: some time series is a set of time variable depending on time t, though the value of the time sequence is uncertain, the

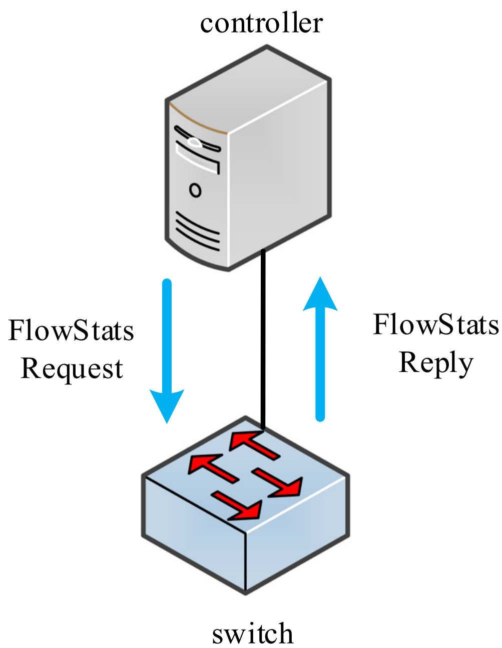

Fig. 3 The exchange process of flow statistics

change of the whole sequence has a certain regularity, which can be described approximately by the corresponding mathematical model. Through the study of the mathematical model, we can clearly understand the structure and characteristics of the time series, so as to achieve the best prediction in the sense of minimum variance. ARMA model combines the function of regression analysis and time series analysis, and has achieved good results in network traffic prediction [20].

The ARMA consists of AR and MA, it describes predictive value of the present time and it is directly related to the previous time self value and the error disturbance.

$ARMA(p, q)$ model is:

$$x(t) = \phi_1 x(t{-}1) + \ldots + \phi_p x(t{-}p) + \varepsilon(t) + \theta_1 \varepsilon(t{-}1) + \ldots + \theta_p \varepsilon(t{-}p) \tag{2}$$

where $\phi_1, \phi_2, \ldots, \phi_p$ is the autoregressive parameter and $\theta_1, \theta_2, \ldots, \theta_p$ is the sliding average parameter.

ARMA is a linear model, and establishing a linear model requires a set of data to determine the order of the model and the value of the unknown parameter. These parameters are ϕ_1, ϕ_2, ..., ϕ_p, θ_1, θ_2, ..., θ_q, σ^2. σ^2 is the variance of $\varepsilon(t)$. The order

Table 1 The pseudo code of the equal sampling algorithm with the fixed interval

Equal interval sampling algorithm

```
INPUT:
C : the controller of software defined networking
S : the switches of software defined networking
t_0 : the initial time of measurement on switch
t_i : the ith time of measurement on switch
T : the measurement period of switch
f : the flow that needs to be measured
P_src : the source node that needs to be transmitted
bytenum : the number of bytes in flow counter
PROCEDURE:
Select flow that needs to be measured;
Select the switch that needs to be measured;
if detected flow is f
  for t_i = t_0 + nT
    Send a FlowStatisticsRequest to the switch;
  end for
else
   continue;
end if
if e is a FlowStatisticsReply event for flow f then
  Measure flow f information of the switch and obtain number of bytes in counter bytenum ;
  x = bytenum / T ;
  Get x and store it;
else
   break;
end if
OUTPUT:
Flow statistic information of equal time sample.
```

of the model is determined based on the AIC criterion. The AIC criterion is a method of determining the order of ARMA model given by H.Akaike. The AIC function is defined as:

$$AIC(s) = \ln\hat{\sigma}^2 + \frac{2s}{N} \tag{3}$$

Where, $\hat{\sigma}^2$ is the estimate value of $\varepsilon(t)$ variance; s is the total number of unknown parameters in the model, and is the sum of $\hat{\sigma}^2$, p and q, that is $s = p + q + 1$; N is the size of known observational data samples. The AIC function of the ARMA model is:

$$AIC(s) = \ln\hat{\sigma}^2 + \frac{2(p+q+1)}{N} \tag{4}$$

The use of the AIC criterion to determine order refers to finding the point $(\hat{p}, \hat{q})$ which makes the static variable $AIC(s)$ minimal in the certain change range of p and q, which is used as an estimate of (p, q). After the order of the model is determined, we use the inverse function method to estimate the value of each parameter. The following is the steps of using inverse function method to estimate the parameters of the $ARMA(p, q)$ model.

First, introduce the backward shift operator B, that is $Bx(t) = x(t-1)$, $B^k x(t) = x(t-k)$, then the equation will be simplified:

$$\left(1-\phi_1 B-\ldots-\phi_p B^p\right)X(t) = \left(1-\theta_1 B-\ldots-\theta_q B^q\right)\varepsilon(t) \tag{5}$$

Next, using inverse function represent $\varepsilon(t)$ as the linear combination of $\{x(t-k), k = 0, 1, 2, \ldots\}$, expression is:

$$\varepsilon(t) = x(t)-\sum_{j=1}^{\infty} I_j x(t-j) = \left(1-I_1 B-I_2 B^2-\ldots\right)x(t) \tag{6}$$

And substitute eq. (6) into eq. (5), the identity equation of B is:

$$\begin{aligned}&1-\phi_1 B-\ldots-\phi_p B^p \\ &\quad= \left(1-I_1 B-I_2 B^2-\ldots\right)\left(1-\theta_1 B-\ldots-\theta_q B^q\right)\end{aligned} \tag{7}$$

According to the principle that corresponding coefficients are equal:

$$\begin{cases}\phi_1 = I_1 + \theta_1 \\ \phi_2 = I_2-\theta_1 I_1 + \theta_2 \\ \vdots \\ \phi_j = I_j-\theta_1 I_{j-1}-\theta_2 I_{j-1}-\ldots-\theta_{j-1} I_1 + \theta_j \\ \vdots\end{cases} \tag{8}$$

where $\theta_j = 0, j > q$; $\phi_j = 0, j > p$. When $j > \max(p, q)$:

$$I_j-\theta_1 I_{j-1}-\ldots-\theta_q I_{j-q} = 0 \tag{9}$$

If the estimated value of I_j is known, the estimated value of the parameter $\theta_1, \theta_2, \ldots, \theta_q$ can be obtained by eq. (9), and then the estimated value of the parameter $\phi_1, \phi_2, \ldots, \phi_p$ can be obtained by eq. (8).

In order to obtain the estimated value of I_j, set $\hat{\rho}_i$ as the autocorrelation function of the network traffic data sample, $\hat{I}_j$ is the estimated value of the parameter $I_j (j = 1, 2, \ldots, p)$, and the value $\hat{I}_j$ can be calculated according to the eq. (9):

$$\begin{pmatrix}\hat{I}_1 \\ \hat{I}_2 \\ \vdots \\ \hat{I}_p\end{pmatrix} = \begin{pmatrix}1 & \hat{\rho}_1 & \cdots & \hat{\rho}_{p-1} \\ \hat{\rho}_1 & 1 & \cdots & \hat{\rho}_{p-2} \\ \vdots & \vdots & & \vdots \\ \hat{\rho}_{p-1} & \hat{\rho}_{p-2} & \cdots & 1\end{pmatrix} \tag{10}$$

After calculating $\hat{I}_j$, estimate the variance of $\varepsilon(t)$ by eq. (11):

$$\hat{\sigma}^2 = \hat{\gamma}_0-\sum_{j=1}^{p} \hat{I}_j \hat{\gamma}_j \tag{11}$$

where $\hat{\gamma}_j$ is the auto-covariance function of the sample of the network traffic data $(j = 1, 2, \ldots, p)$.

So far, the parameters of the ARMA model have been determined, and the modeling is completed.

3.4 Traffic Prediction Based on ARMA Model

3.4.1 Pretreatment of Traffic Data

$ARMA(p, q)$ introduced above describes the stationary sequence. If the input data of the model are non-stationary time series, the input data needs to be smoothed. SDN traffic is non-stationary time series. In order to smooth it, we use multiple differential method to convert network traffic into homogeneous nonstationary sequence. After the difference, the nonstationarity of the new sequence is suppressed, and the amplitude of the new sequence is changed. The number of difference is called homogeneous order.

Let ∇ be a difference operator:

$$\nabla x_t = x_t-x_{t-1} \tag{12}$$

$$\nabla^2 x_t = \nabla\left(x_t-x_{t-1}\right) = x_t-2x_{t-1} + x_{t-2} \tag{13}$$

And we can figure out that:

$$\nabla^k = (1-B)^d x_t \tag{14}$$

If x_t is a dth order homogeneous non-stationary sequence, after difference d times, the new sequence is generated:

$$y_t = (1-B)^d x_t \tag{15}$$

This is a stationary sequence.

If the sequence shows a significant linear trend, A first-order difference can achieve steady trend; sequence shows the trend of curve. To increase the influence of the difference order on the extraction of the trend of curve, the 2nd order or 3rd order difference is usually used. The effect of the differential processing is to convert the non-stationary data sequence into a stationary random sequence with a mean of zero.

3.4.2 Traffic Prediction Algorithm Based on ARMA Model

The steps for SDN traffic prediction are as follows:

- **Step 1:** Use equal interval sampling algorithm to sample flow data;
- **Step 2:** Obtain sampling sequence $x(t)$;
- **Step 3:** Verify the stationarity of the original sequence. If the sequence does not satisfy the stationary condition, the sequence can meet stationary condition by differential transformation;
- **Step 4:** Determine the ARMA order p and q according to the AIC criterion;
- **Step 5:** Use the inverse function method to find the parameters (θ_1, θ_2, ..., θ_q) and (ϕ_1, ϕ_2, ..., ϕ_p) of the model $ARMA(p, q)$, then test the significance of the parameters and the rationality of the model;
- **Step 6:** Diagnose and analysis, prove that the obtained model is consistent with the observed data characteristics;
- **Step 7:** Use the fitted model to predict and analyze, and obtain the predict results;
- **Step 8:** Further analyze the simulation results.

Figure 4 indicates a flow chart of the proposed traffic prediction algorithm.

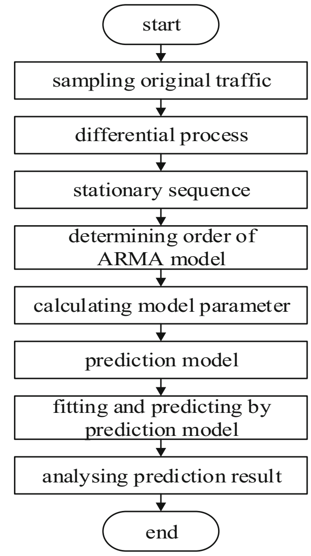

Fig. 4 Flow chart of traffic prediction algorithm

4 Simulation Analysis

4.1 Simulation Environment

The SDN simulation platform is introduced. We install POX controller on a PC equipped with Linux system, and install the Mininet platform on the virtual machine. Using the equal interval sampling algorithm and equal data packet sampling algorithm introduced above to query the data required for the simulation of the switch counter. At last, we use MATLAB to model and analyze the experimental data. Table 2 is some software used in the simulation environment.

Mininet is a widely used SDN simulation software that can quickly build a large-scale SDN prototype system on a common computer with limited resources to simulate a real network environment. Since the controller is the core of SDN, the controller used in this article is the POX controller, which is written entirely in Python. The Python language is more easily accepted by researchers because it is easy to learn, and has received widespread attention and application. The POX controller provides network visualization, topology discovery, and network measurements.

4.2 Equal Interval Sampling Simulation

We firstly obtain the simulation traffic data required for the simulation: 1 min of sampling data, 1 min of real data, 5 min of sampling data, 5 min of real data, 10 min of sampling data, and 10 min of real data. Figure 5 is a comparison of the sampled data of 1 min, 5 min and 10 min with corresponding real data respectively. Figure 6 shows the relative error between the sampled data and the real data at different sampling intervals.

The calculation result shows that when the sampling interval is 1 min, the average relative error between the real sample and the sample is 0.0483. When the sampling interval is 5 min, the average relative error between the real sample and the sample is 0.0822. When the sampling interval is 10 min, the average relative error between the real samples and the sample is 0.1502. Therefore, as the sampling interval increases, the relative error between the sampled data and the actual data also increases.

Table 2 simulation environment

software	function	version
Mininet	SDN	2.2.2
OpenvSwitch	Switch	1.4.3
POX	SDN Controller platform	Betta ranch
Vmware workstation	Virtual software	10.0.4
Linux(Ubuntu)	Operating system	16.04
Python	Programing language	2.7

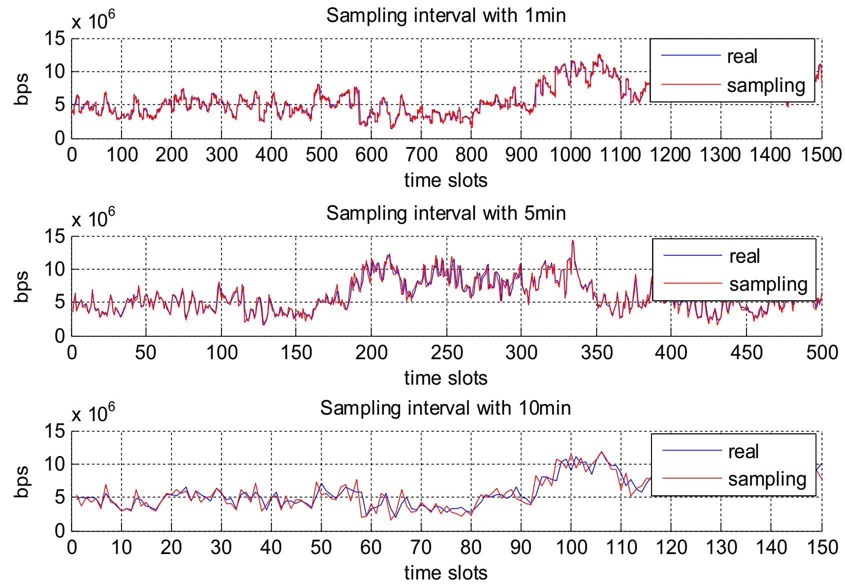

Fig. 5 Comparison of sampling values with the fixed interval

Then, we analyze the prediction results of sampling data and real data, and compare the simulation results at each sampling interval. Figure 7 is a predictive comparison chart of the real data and the sampled data (sampling interval 1 min). Figure 8 indicates a predictive comparison chart of the real data and the sampled data (sampling interval 5 min). Figure 9 shows a predictive comparison of the real data with the sampled data (sampling interval 10 min). From Fig. 7, Fig. 8 and Fig. 9, both the prediction of the real data and sampled data can accurately predict the changing trend of real data and sampled data, and the prediction result is also very good. However, as the sampling interval is coarsened, the prediction effect becomes worse and worse, and the prediction result becomes unstable. It shows that smaller is the sampling interval, more accurately can ARMA predict the network traffic changes, and higher the stability of the prediction results will be. Then we will analyze the prediction error of each sampling interval quantitatively by the cumulative distribution of relative error and relative error, and then compare the prediction performance.

We analyze the prediction value of the real data and the sampled data, and compare the simulation results at each sampling interval. Figure 10, Fig. 11 and Fig. 12 compare the predicted values of the real data and the predicted values of the sampled data respectively (sampling intervals 1 min, 5 min, 10 min). From Fig. 10, Fig. 11 and Fig. 12, the predicted values of the sampled data can well approximate the flow variation trend of the real data. However, as the sampling interval is coarsened, the approximation between them becomes worse and worse, and the stability also deteriorates. It

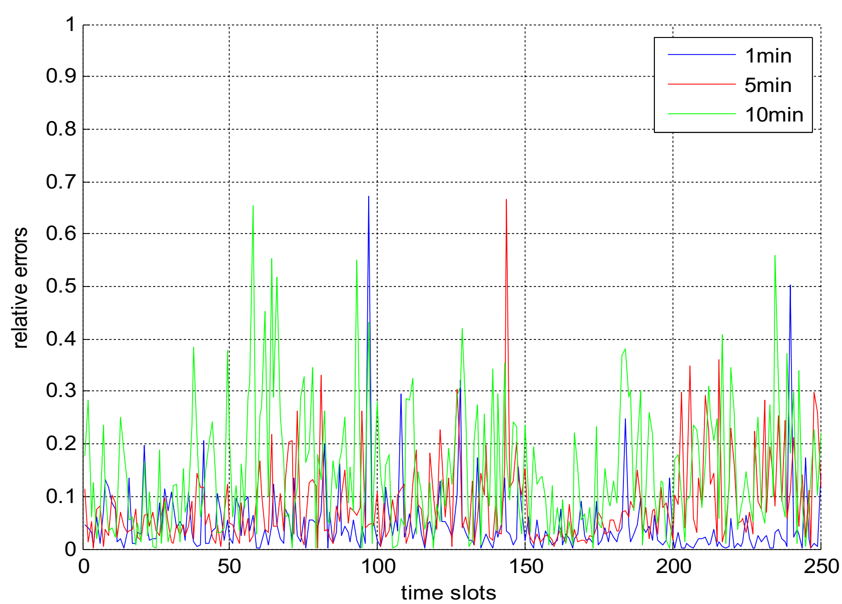

Fig. 6 Relative error of equal interval sample

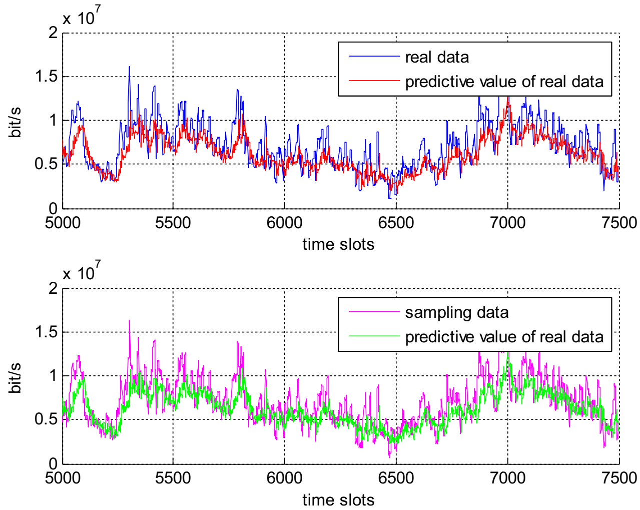

Fig. 7 Comparison of real value prediction with sampling value prediction(1 min)

shows that, at a certain sampling interval, the predicted value of sample data can be used to replace the predicted value of the real data, that is, the prediction values of the sample data can be used to predict traffic variation trends of the real network. Thereby, it is possible to predict the trend of traffic change with a small network measurement overhead. However, when the sampling interval exceeds a certain threshold, the approximate relationship between the predicted value and the true value is weakened. Therefore, the predicted value of the sampled data cannot predict the trend of network traffic.

In summary, when the sampling interval is small, on the one hand, the prediction value can predict the change trend of the original data accurately, the prediction performance of ARMA model is better, and the stability of the prediction results is better. On the other hand, the predictive value of the sampled data can be well approximated by the real data.

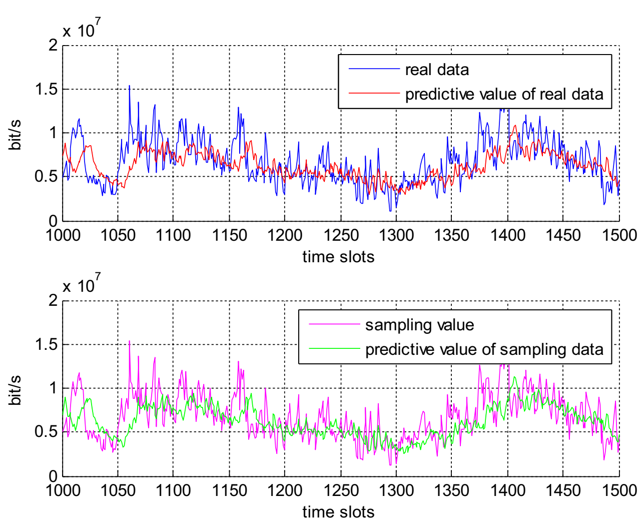

Fig. 8 Comparison of real value prediction with sampling value prediction (5 min)

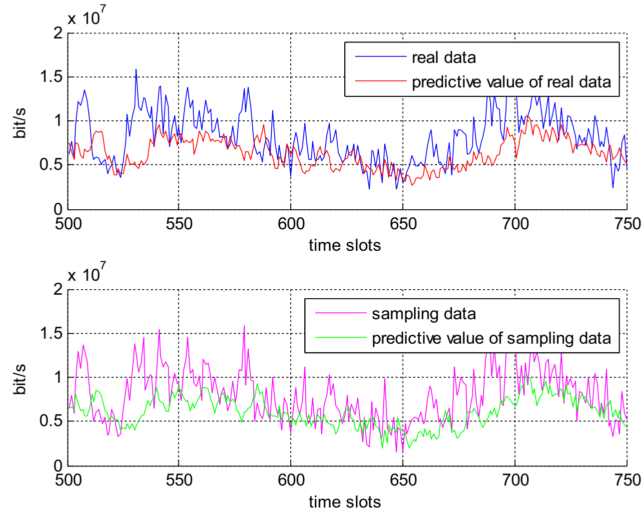

Fig. 9 Comparison of real value prediction with sampling value prediction (10 min)

Therefore, the predictive value of the sampling data can be used to predict the change trend of real network traffic, so as to predict traffic variation trends with smaller network measurement cost. When the sampling interval is large, the correlation between the data is weakened, and the prediction result of network traffic is poor. The prediction performance of ARMA model decreases, and the stability of the prediction algorithm is also decreased, sometimes resulting in large prediction errors.

Next, we analyze the real data and predicted value, the relative error between the real data and the predicted value. Figure 13 is the relative error of traffic prediction. Here, we have calculated the average relative error value respectively: when the sampling interval is 1 min, the average relative error of the real data and the predictive value is 0.2454, the average relative error of the sampling data and the prediction value is

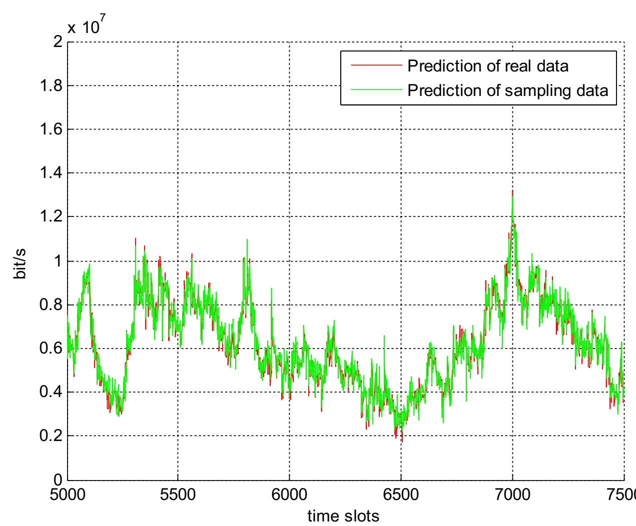

Fig. 10 Comparison of the predicted value of the true value with the predicted value of the sample value (1 min)

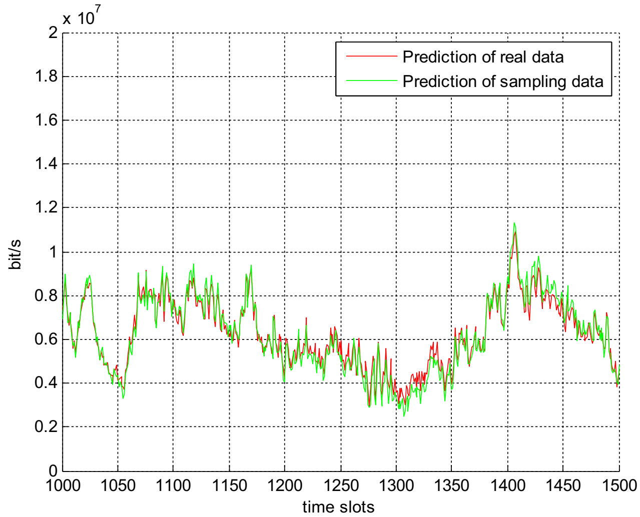

Fig. 11 Comparison of the predicted value of the true value with the predicted value of the sample value (5 min)

0.2577. When the sampling interval is 5 min, the average relative error of the real data and the predictive value 0.3385, the average relative error of sampling data and the prediction value is 0.3458. When the sampling interval is 10 min, the average relative error of the real data and the predictive value is 0.5122, the average relative error of the sampling data and the prediction value is 0.5360. According to the calculation, the relative error will increase with the increase of sampling interval, and the effect of traffic prediction will also decrease.

Finally, the cumulative distribution function is introduced to describe the relative error more intuitively. Figure 14 is the cumulative distribution function of relative error. Figure 14 shows that the relative error accumulation curve of 1 min is.

above that of 5 min and that of 10 min, which further shows that when the sampling interval is 1 min, the relative error is

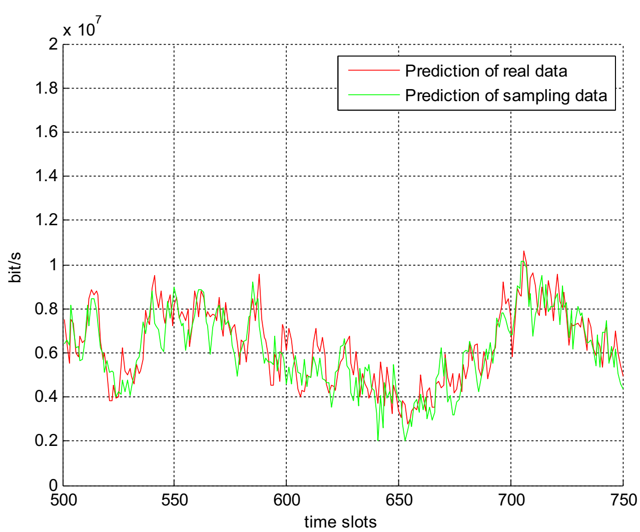

Fig. 12 Comparison of the predicted value of the true value with the predicted value of the sample value (10 min)

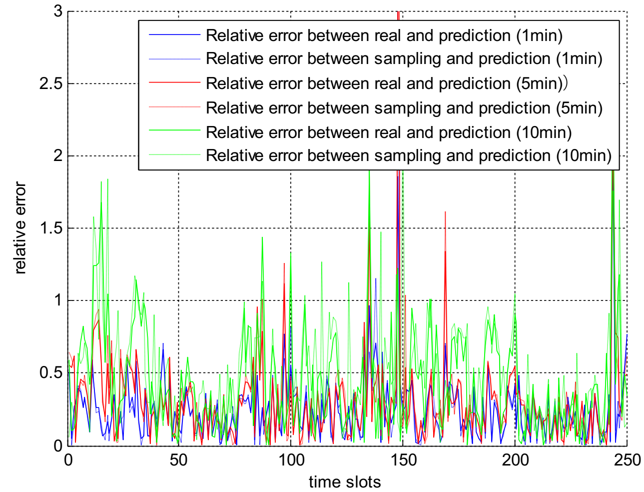

Fig. 13 Relative error of traffic prediction (1 min, 5 min, 10 min)

minimal. When the relative error is 0.4, the cumulative distribution function value of the relative error (1 min) of the real data and the predictive value of is 0.9, the cumulative distribution function value of the relative error (1 min) of the sampled data and the predictive value of is 0.9, the cumulative distribution function value of the relative error (5 min) of the real data and the predictive value of is 0.7, the cumulative distribution function value of the relative error (5 min) of the sampled data and the predictive value of is 0.7, the cumulative distribution function value of the relative error (10 min) of the real data and the predictive value of is 0.53, the cumulative distribution function value of the relative error (10 min) of the sampled data and the predictive value of is 0.47. This indicates that when the relative error is 0.4, the algorithm can accurately predict 90% of real data and sampled data (1 min). When the sampling error is 0.4, the algorithm can accurately predict 70% of real data and sampled data (5 min). When the relative error is 0.4, the algorithm can accurately predict 53% of real data (10 min), and accurately predict 49% of sampled data

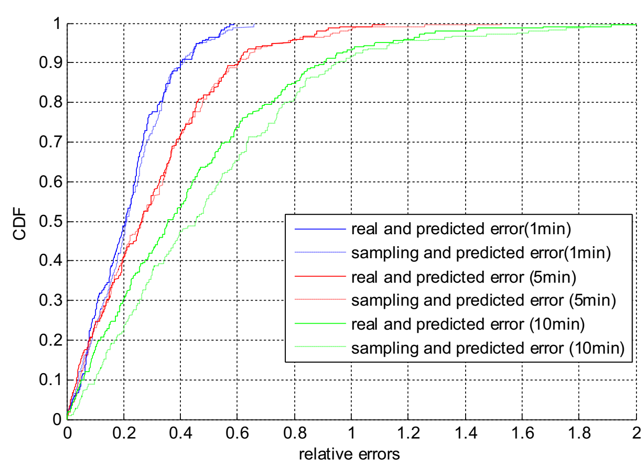

Fig. 14 Cumulative distribution function of sampling error in equal interval sampling

(10 min) at the same time. This shows that the smaller the sampling interval, the better prediction accuracy of the algorithm.

5 Conclusion

Traffic prediction is of great significance for network performance analysis and network planning. SDN provides a good platform for the application of traffic prediction algorithm. The equal interval sampling algorithm is used respectively to obtain the simulation data from SDN. Then, the ARMA prediction model is established by using simulation data, and a SDN traffic prediction algorithm based on ARMA model is proposed. The simulation results show that the ARMA model can predict the network traffic behavior trends accurately. The smaller the sampling interval is, the better the flow forecasting effect becomes, the higher the prediction accuracy is, and the higher the stability of the algorithm is.

Acknowledgments This work was supported by National Natural Science Foundation of China (No. 61571104), Sichuan Science and Technology Program (No. 2018JY0539), Key projects of the Sichuan Provincial Education Department (No. 18ZA0219), Fundamental Research Funds for the Central Universities (No. ZYGX2017KYQD170), and Innovation Funding (No. 2018510007000134). The authors wish to thank the reviewers for their helpful comments. Dr. Dingde Jiang is corresponding author of this paper (email: merry_99@sina. com).

References

1. Sezer S, Scott-Hayward S, Chouhan P et al (2013) Are we ready for SDN? Implementation challenges for software-defined networks. IEEE Commun Mag 51(7):36–43
2. Kuang X, Xu L, Huang Y et al (2010) Real-time forecasting for short-term traffic flow based on General Regression Neural Network. Proc. ICA'10, 2010, pp 2776–2780
3. Nunes B, Mendonca M, Nguyen X et al (2014) A survey of software-defined networking: past, present, and future of programmable networks. IEEE Commun Surv Tutor 16(3):1617–1634
4. Kreutz D, Ramos F, Verissimo P et al (2015) Software-defined networking: a comprehensive survey. Proc IEEE 103(1):14–76
5. Al-Najjar A, Layeghy S, Portmann M (2016). Pushing SDN to the end-host, network load balancing using OpenFlow. In: Proc. PCCW'16, 2016, pp 1–6
6. Saxena M, Kumar R (2016) A recent trends in software defined networking (SDN) security. In: Proc. CSGD'16, pp 851–855
7. Ning C, Wang J (2015). Auto regressive moving average (ARMA) prediction method of bank cash flow time series. In: Proc. CCC'15, 2015, pp 4928–4933
8. Tan Y, Cheng J, Zhu H et al. Real-time life prediction of equipment based on optimized ARMA model. In: Proc. PSHMC'17, 2017, pp 1–6
9. Jiang D, Wang X, Guo L et al (2011) Accurate estimation of large-scale IP traffic matrix. AEU-Int J Electr Commun 65(1):75–86
10. Li Y, Liu H, Yang W et al (2017) Predicting inter-data-center network traffic using elephant flow and sublink information. IEEE Trans Netw Serv Manag 13(4):782–792
11. Wei D (2017) Network traffic prediction based on RBF neural network optimized by improved gravitation search algorithm. Neural Comput & Applic 28(8):2303–2312
12. Zhang C, Zhang H, Qiao J et al (2019) Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data. IEEE J Selected Areas Commun 37(6):1389–1401
13. H Lu, F Yang. A network traffic prediction model based on wavelet transformation and LSTM network, in Proc. ICSESS'18, 2018, pp. 1–4
14. Xie J, Yu F, Huang T et al (2019) A survey of machine learning techniques applied to software defined networking (SDN): research issues and challenges. IEEE Commun Surv Tutor 21(1):393–430
15. Dai L, Yang W, Gao S et al. EMD-based multi-model prediction for network traffic in software-defined networks. In: Proc. MASS'14, 2014, pp 539–544
16. Azzouni A, Pujolle G (2018) NeuTM: A neural network-based framework for traffic matrix prediction in SDN. In: Proc. NOMS'18, 2018, pp 1–5
17. Li D, Xing C, Dai N et al (2019) Estimating SDN traffic matrix based on online adaptive information gain maximization method. Peer-to-Peer Netw Appl 12(2):465–480
18. Yu B, Yang G, Yoo C (2018) Comprehensive prediction models of control traffic for SDN controllers. In: Proc. NSW'18, 2018, pp 262–266
19. Adrichem N, Doerr C, Kuipers F (2014) OpenNetMon: Network monitoring in OpenFlow Software-Defined Networks. In: Proc. NOMS'14, 2014, pp 1–8
20. Cao A, Qiao Y, Sun K et al (2018) Network traffic analysis and prediction of Hotspot in cellular network. In: Proc. IC-NIDC'18, 2018, pp 452–456
21. Jiang D, Wang Y, Lv Z et al. Big data analysis-based network behavior insight of cellular networks for industry 4.0 applications, IEEE Transactions on Industrial Informatics, 2019, online available. https://doi.org/10.1109/TII.2019.2930226
22. Jiang D, Huo L, Lv Z et al (2018) A joint multi-criteria utility-based network selection approach for vehicle-to-infrastructure networking. IEEE Trans Intell Transp Syst 19(10):3305–3319
23. Jiang D, Huo L, Song H (2018) 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
24. Jiang D, Wang W, Shi L et al (2018) A compressive sensing-based approach to end-to-end network traffic reconstruction. IEEE Trans Netw Sci Eng 5(3):1–12
25. Jiang D, Huo L, Li Y (2018) Fine-granularity inference and estimations to network traffic for SDN. PLoS One 13(5):1–23