Research on Traffic Optimization Scheme of SDN Network Based on ME-HMM

Yanchun Yang* and Hongfeng Sun
School of Data and Computer Science, Shandong women’s University, Jinan, China
*Corresponding author email: 34009@sdwu.edu.cn

Abstract. In this paper, a traffic optimization scheme based on Maximum Entropy- Hidden Markov Model (ME-HMM) in Software-Defined Networking (SDN) networks is proposed in order to resolve the load imbalance in SDN architecture. First, the data flow generated by the host in the SDN network was monitored, and denoising and analyzing would be carried out with the maximum entropy (ME) model to predict the network traffic for a period of time in the future, then the controller would be deployed. Next, the traffic flow was evaluated with Hidden Markov Model (HMM) and the deployment was optimized in the SDN control plane. With the feedback information, the scheme could optimize SDN network traffic and promote load balancing.

Keywords: Software-Defined Networking; Maximum entropy; Hidden Markov; Optimization.

1. Introduction
With the techniques of cloud computing and big data being developed, traditional IP networks have been unable to meet the needs of dynamically changing networks. In order to adapt to the rapid growth of network traffic and deploy business rapidly, Software Defined Networking came into being. SDN is actually a new architecture of network that separates the network controlling function with forwarding function [1], which can improve the programmability and flexibility of the whole network and simplify the network management.

In traditional IP networks, devices such switches and routers, have both the data plane and the control plane which were implemented in one hardware device, while in SDN, the control plane is isolated from the data plane. The control plane was implemented as a software application. SDN has the characteristics of openness, logical centralized control, programmability, scalability, abstraction and virtualization [2].
SDN forwards data based on flow which unifies all the network appliances such as switches, middleware, firewalls, routers, and making network management more flexible. As shown in Figure 1, from top to bottom, an SDN network architecture can be divided into three layers, application plane, control plane and data plane. In control plane there are north programming interface and south programming interface included. A SDN controller is the core which is an application, and it based on protocols, for example, OpenFlow. OpenFlow can enable intelligent networking by managing flow control to and permit servers to talk with switches about where to send packets [3].

2. Relevant Research
Currently, there is an important quantity of studies that are related to the traffic optimization of SDN networks. For example, LABERIO equalization algorithm [4], was proposed, which used the maximum and minimum residual bandwidth method to calculate the optimal next-hop link and distributes the traffic in the high load link to the low load link to balance the load of the link. Agarwal et al proposed an approach [5] which made load balancing become a problem of linear programming, which optimize the data flow of SDN nodes with only one shortest path, and limited the performance of network. A flow-based partitioning strategy [6] was proposed for the load balancing problem, which will be transmitted after the large flow segmentation of the controller. A stage dynamic controller load balancing strategy (SDCLB) [7] was proposed to solve the problem of controller load imbalance, which completed the whole task in two phases. First, the candidate set of immigration controller was determined in phase one, with the aim of load equalization, and the delay and load were considered comprehensively, then, an index function was designed to determine which switch would be migrated. In phase two, it an improved EMD model was proposed with minimum migration cost considering the connectivity of network nodes, and the linear approximation algorithm was used to solve it quickly. Traffic prediction and deployment strategy of SDN based on machine learning [8] was studied, in which a traffic prediction and controller pre-deployment model (PPME) based on hidden Markov optimization is proposed. But none of them considered the problem that the deployment of the controller will affect the ability of the controller to handle network events, according to which, the network administrator should focus on the influence of the location of the controller on the recovery delay, thus saving the communication overhead between switches and controllers, and the switch migration cost. Based on this point, a traffic optimization scheme based on Maximum Entropy Hidden Markova Model in SDN networks was proposed in this paper.

3. Traffic Optimization Scheme Based-on ME-HMM
3.1. SDN Network Model
According to graph theory, an SDN network is modeled into an undirected graph. The graph is noted as $G = \{V, E\}$. $G$ stands for the network, and $V$ stands for the set of nodes of switches and controllers, and $V= \{v_1, v_2, ...v_m\}$, while $E$ stands for the set of links between controllers and switches, or between controllers, and $E= \{e_1, e_2, ...e_n\}$. Suppose there are $K$ controllers and $L$ switches in the SDN network $G$, $C= \{c_1, c_2, ...c_k\}$ represents the controller set, the switch set is expressed as $S= \{s_1, s_2, ..., s_l\}$. Each of the controllers and switches under its management make up a subdomain. All controllers communicate with each other from different domains, and they share the same view of network. Every switch in the SDN network has one unique master controller while many slave controllers. OpenFlow protocol works as an intermediate between the data plane and the control plane, which not only needs to reflect adding and updating of each data stream path truly onto the infrastructure switch and needs to be the channel helps the controller collect data flow information [9]. There are three different roles of master, equal and slave for the controller in OpenFlow [10]. The slave controller can be transformed from transition state to master controller, thus to obtain the smooth migration of the switch. Most studies took all slave controllers as incoming controllers and didn’t take the effect of connectivity between nodes on the network state too much into account, leading to idealized migration strategies. But in fact, it should base on the actual network topology, and the node connectivity should be considered.
3.2. ME-HMM Model

Suppose, each host in the network has n states, which can be noted by $S_1, S_2, \ldots, S_n$ respectively which are assumed to be independent of each other, and the probability of a particular event is $P_1, P_2, \ldots, P_n$.

Definition 1 Assume $U=\{u_1, u_2, \ldots, u_n\}$ is a discrete random variable, and the entropy of a user can be expressed by the average of uncertainty in a certain state as Eq. 1.

$$H(U) = E(-\log P_i) = -\sum P_i \log P_i \tag{1}$$

In Eq. 1, $P_i$ represents the possibility of obtaining the $i$-th signal of event number $i$. If $P_i=0$, that means the possibility of $U=u_i$.

HMM model is a graphical model. In general, HMM is used to predict hidden states with sequential data, such as text, speech, or weather etc. as shown in Figure 2, in which $S(t)$ means the state at the $t$ moment, while $O(t)$ represents the observable state at the $t$ moment. A complete set of states for each subdomain can be expressed in the form of a set: $\Sigma = (O_1, O_2, \ldots, O_M)$. Here $O_i$ ($0 \leq t \leq M$) represents the network real-time status of internal subdomain at $m$ moment.

![Figure 2. State diagram of HMM](image)

In networks, traffic changes over time and space. For example, day traffic is usually higher than night traffic, and network traffic can also be drastically change in a very short time. In this research, data was captured as a generated sequence, and data features include generation time, protocol for transport control layer, destination address, source address, packet configuration information, such as packet size, port number, Media Access Control (MAC) address, etc. According to the ordinary protocols in TCP/IP, such as Hyper Text Transfer Protocol (HTTP), Hyper Text Transfer Protocol over Secure Socket Layer (HTTPS), Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Internet Control Message Protocol (ICMP), ratio was calculated which was shown in Table 1.

| Percentage | HTTP | HTTPS | TCP | UDP | ICMP | Others |
|------------|------|-------|-----|-----|------|--------|
| 21.33      | 17.62| 33.29 | 16.56| 10.73| 0.47 |        |

3.3. Scheme

Step 1, historical network data flow analysis.

Historical network data flow was monitored by OpenFlow protocol according to an adaptive polling algorithm to switches with query request, then the flow rate is monitored according to the received reply message, while it will interact with the routing and flow scheduling mechanism.

Step 2, with maximum entropy algorithm, the data flow for a period of time in the future was predicted.

Step 3, the controller was deployed in advance to cut down the number of controller migration and cut down the delay of network.

With advanced Louvain Community detection algorithm [11], first according to the node similarity the chain weight was redefined, the subdomain modularity was calculated, and then the whole network was divided by the load difference of the device independently. Finally, deploy a single controller for each subdomain that has been partitioned considering the transmission delay between switch and controller, inter-controller transmission delay and control link reliability. Energy saving [12] would be considered also.

Step 4, HMM was used to evaluate the prediction results.
Step 5, the controller deployment was optimized according to the evaluation feedback. Repeat step 3, step 4 and step 5 to reach an optimized status. The whole workflow of the traffic optimization scheme based on ME-HMM is show as Figure.3 as below.

Historical data flow analysis

Prediction of future data flow with ME model

Deployment of controllers for each subdomain

Evaluation with HMM model

Optimization

Feedback

Figure 3. Workflow of traffic optimization scheme based on ME-HMM

4. Summary
In this paper, a scheme was proposed to introduce the way for traffic optimization based-on the maximum entropy and hidden Markov model in the control layer. First, the historical network data flow is monitored and analyzed. Then the data flow for a period of time in the future was predicted with maximum entropy model, and the controller was deployed in advance to cut down the number of controller migration and cut down the delay of network. At last, Hidden Markov Model was used to evaluate the prediction results and provide feedback to optimize the controller deployment. Further research would be taken on this scheme about the efficiency in larger scale of SDN networks.

Acknowledgement
This research was financially supported by the Cooperative Education Project of High Education Department of Ministry of Education of China (Grants No. 201702020004).

References
[1] A.T. Tuan, L. Mhamdi, D. McLernon, et al, Deep learning approach for network intrusion detection in software defined networking, 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM), IEEE, 2016, pp. 258-263.
[2] T.D. Nadeau, K. Gray, SDN: Software Defined Networks: an authoritative review of network programmability technologies, 1th ed. Sebastopol: O'Reilly Media, Inc., 2013, pp. 117-156.
[3] H. Kim, N. Feamster, Improving network management with software defined networking, IEEE Communications Magazine, 2013, pp. 114-119.
[4] L. Hui, S. YAO, M. GUO, et al, LABERIO: Dynamic load balanced Routing in OpenFlow enabled Networks, IEEE International Conference on Advanced Information Networking & Applications, 2013.
[5] S. Agarwal, M. Kodialam, T.V. Lakshman, Traffic engineering in Software Defined Networks, Proceedings IEEE Infocom, IEEE, 2013, pp. 2211-2219.
[6] W. Braun, M. Menth, Load dependent flow splitting for traffic engineering in resilient Open-Flow networks, Proc of 2015 International Conference and Workshops on Networked Systems (NetSys), 2015, pp. 1-5.
[7] Information on http://kns.cnki.net/kcms/detail/51.1196.TP.20191024.1016.033.html.
[8] Information on http://kns.cnki.net/kcms/detail/31.1289.tp.20191217.1447.003.html
[9] Y.G. Lu, X.W. Wang, F.L. Li, et al, Dynamic load balancing and energy saving mechanism in software defined networks, Chinese journal of computers, 2019, pp. 125:1-15

[10] T.T. Ren, Y.W. Xu, Analysis of the new features of OpenFlow 1.4, Proc of the 2nd International Conference on Information, Electronics and Computer, Amsterdam: Atlantis Press, 2014, pp. 73-77.

[11] V.D. Blondel, J.L. Guillaume, R. Lambiotte, et al, Fast unfolding of communities in large networks, Journal of Statistical Mechanics: Theory and Experiment, 2008, pp.155-168.

[12] Z. Chkirbene, A. Gouissem, R. Hadjidj, et al, Efficient techniques for energy saving in data center networks, Computer Communications, 2018, pp.111-124.