Experimental assessment of traffic prediction assisted data center network reconfiguration method

Xiaotao Guo, Xuwei Xue, Fulong Yan, Bitao Pan, Georgios Exarchakos, Nicola Calabretta

Institute of Photonic Integration, Eindhoven University of Technology, Eindhoven, the Netherlands, x.guo@tue.nl

Abstract *We experimentally demonstrate a traffic prediction assisted network reconfiguration method (TPANR) for data center networks based on deep reinforcement learning (DRL). Traffic prediction model performs the lowest MSE of 2.64E-4. Exploiting one-step ahead traffic prediction and DRL-based automatic network reconfiguration, TPANR achieves 17.3% latency improvement.*

Introduction

Cloud applications proliferate and increasingly differentiate imposing severe requirements to current data center networks (DCN) like low latency/jitter, high bandwidth, and flexible network resource provision. Thus, a real-time network reconfiguration is necessary to support the dynamic network traffic and satisfy diverse quality of service (QoS) of these applications. Some heuristic methods like open shortest path first (OSPF)[1] were proposed for network traffic routing, but these methods usually provide sub-optimal solutions and may degrade application QoS. Recently, we proposed and assessed an automatic real-time network reconfiguration method (DCR2L) based on deep reinforcement learning (DRL)[2], which can explore the optimal solution directly from the network without a-priori environment model. However, this reactive framework can only reconfigure the network after monitoring the QoS degradation, resulting in an inevitable network performance deterioration. To avoid this issue, a proactive framework with a traffic prediction model is required to reconfigure the network in advance.

Several artificial intelligence prediction models have been proposed for applications scheduling and allocation[3,4]. However, the time-consuming iteration processing and requirement of a-priori environment models limit the applicability of these methods on complex and dynamic DCNs. Meanwhile, an adaptive model was developed based on machine learning technologies for DCN resource utilization prediction[5]. Benefiting from the development of deep neural network (DNN), many DNN based methods were proposed for time-series data prediction, such as recurrent neural network (RNN)[6] and gate recurrent unit (GRU)[7]. These models can estimate the near-future traffic as part of proactive reconfiguration methods. However, there has not been yet an experimental demonstration of DNN-based network reconfiguration method and automatic operation.

In this work, we experimentally demonstrated a traffic prediction assisted automatic data center network reconfiguration method (TPANR) based on DNN and DRL, empowered by a software-defined networking (SDN) controller. Comparing the predicted traffic with the actual traffic, experimental results show that the GRU based traffic prediction model achieves the lowest mean square error (MSE) of 2.64E-4. Exploiting the traffic prediction model, the performance of TPANR method is experimentally assessed. It is demonstrated that the TPANR method achieves a network latency improvement of up to 17.3% compared with current methods.

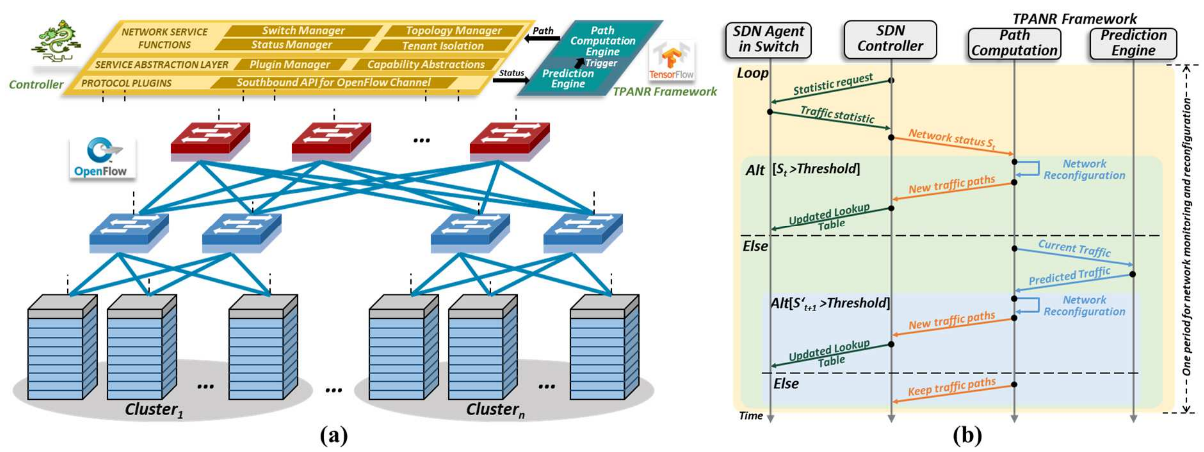

Fig. 1: (a) Architecture of the SDN-enabled DCN and (b) Schematic of the TPANR method

978-0-7381-4679-9/21/$31.00 ©2021 IEEE

TPANR architecture and operation

The SDN-enabled DCN architecture comprising the data/control plane and TPANR framework is depicted in Fig. 1(a). The information about data plane layout is stored in the Topology Manager, while SDN agents of all the switches send network statistics to the RYU controller via the Southbound Interface. Note that the leaf-spine topology is shown in the data plane, but TPANR method is also applicable to other topologies. The TPANR framework, consisting of a Traffic Prediction Engine (TPE) and a Path Computation Engine (PCE), cooperates with Status Manager to reconfigure the data plane network. Fig. 1(b) shows the schematic of TPANR method. The RYU SDN controller periodically sends requests to SDN agents for network statistics. Network statistics are pre-processed by SDN controller and forwarded to PCE of TPANR framework. If current network status s_t violates application QoS requirements, PCE is trigged, generates new paths of corresponding traffic, and sends the results to the SDN controller. Packet lookup tables are updated based on new traffic paths and distributed to the switches in the data plane by Switch Manager. Otherwise, current network status is sent to TPE. Combining the stored historical traffic, TPE forecasts the near-future traffic. According to the predicted status s'_{t+1}, PCE reconfigures the data plane network or keeps the current configuration.

The DNN based model is applied in the TPE. Assumed *N* neuron units in the input layer, at time *t*, the *i*-th neuron unit represents the historical traffic at time t-N+i (i=1, ···, N). The neuron unit of output layer represents the predicted traffic at time t+1. Thus, the target of DNN based model is to minimize the error between the model output and real traffic at time t+1. To train the DNN based model and assess the performance of traffic prediction, the MSE is selected as the loss function. For the PCE, because the network states and reconfiguration actions are continuous values, we choose deep deterministic policy gradient (DDPG) model. The input layer of actor network is network state s_t, which is defined as s_t = (P, src, dst, U). P is application QoS priority, src/dst is the traffic source/ destination address, and U is the normalized utilization of link bandwidth. The output layer of actor network in DRL agent is the action a_t of traffic allocation, defined as a vector a_t = (w_1, ···, w_m), where w_m is the traffic ratio allocated to m-th available traffic path. The input layer of critic network includes the action a_t and state s_t, while the output layer is estimated Q-value function. The expected Q-value function is based on the reward R_t:

$$R_t = -\sum_{p=1}^{P}(k_1^p La_p + k_2^p Lp_p) \qquad (1)$$

where La_p and Lp_p are the normalized average end-to-end latency and packet loss of application at time t with QoS priority p, while k^p_i is weighted factors between 0 and 1 (i=1, 2). The target of actor network in DRL agent is to maximize the Q-value function of policy $\mu(s)$ based on given state s_t. The target of critic network in the DRL agent is to minimize the error between estimated and expected Q-value functions based on the MSE. The target neural network is applied in the DRL agent training, which shares the same structure as the main neural network and is updated at a slower pace. More details about the DRL agent and training can be found in [2].

Experimental setup and results

The experimental setup for assessment of the TPANR method is shown in Fig. 2(a). It consists of 8 Dell R210 servers grouped in 4 racks and 2 clusters, and each server comprises one 3400 series Quad-core Intel Xeon processor, 16GB memory, dual 10G Intel NIC for data plane connection, and 1G NIC for control plane connection. For the network connection, there are 4 customized SDN-enabled top of rack (ToR) switches based on Broadcom chip and one 128-port Broadcom BCM956846K-02 electrical switch interconnected as a leaf-spine topology. The servers use SFP+ multimode fiber transceivers,

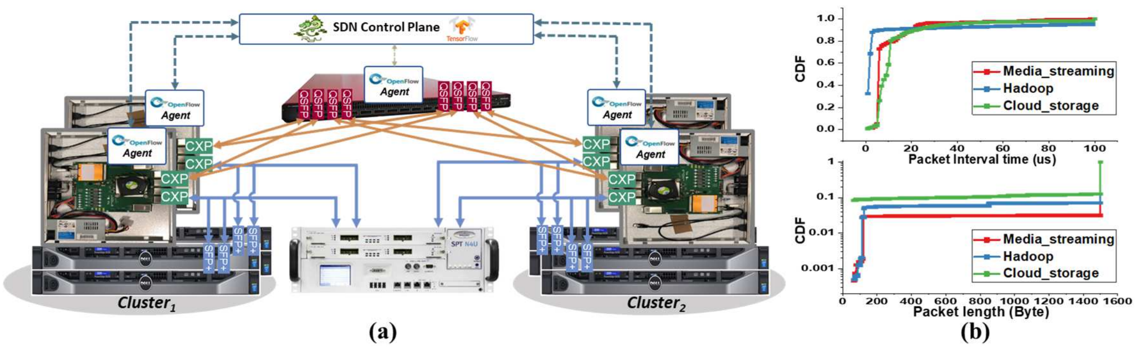

Fig. 2: (a) Experimental demonstration setup (b) Statistics CDF of deployed applications

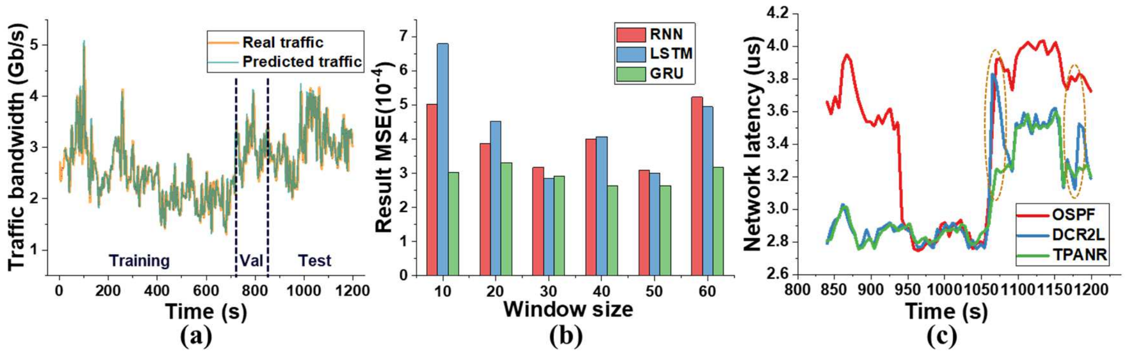

Fig. 3: (a) Real and predicted bandwidth of the overall traffic (b) MSE comparison under different window sizes (c) Average network latency with different methods

while ToR and spine switches use 120G CXP and 40G QSFP transceivers respectively. The TensorFlow framework based TPANR method is deployed in the RYU controller based control plane. The hardware of control plane is equipped with 2 12-core Intel Xeon 5118 processors, one NVIDIA Quadro 16 GB P5000 GPU, and 128GB memory. The SPIRENT Ethernet Testing Center is also connected to all the ToR switches to measure the network performance. The RYU SDN controller sends statistic request every second, and three applications are deployed in different racks to generate network traffic, including media streaming, Hadoop, and cloud storage. The statistic cumulative distribution function (CDF) of applications (packet length and interval time) are depicted in Fig. 2(b). It is shown that media streaming has the most packets of 1500 Bytes, while Hadoop has the most packets with an interval time of longer than 100us.

The real-time bandwidth of the total network traffic in the data plane is illustrated in Fig. 3(a). Three different DNN based models (RNN, LSTM, and GRU) are applied in the TPE to predict the network traffic. The batch size is set to 32 for model training, and the learning rate is set as 0.001. There are two hidden layers in the model, while each layer has 30 units. The collected traffic is divided into two parts: 85% for training, 15% for validation. Trained models are deployed in the control plane for testing. The MSE losses of three models in the testing under different window sizes are shown in Fig. 3(b). The window size is defined as the length of historical data used for traffic prediction of the coming period. It is shown that the window size has a significant impact to prediction performance, especially for RNN and LSTM. Compared with other two models, GRU model achieves the lowest MSE under most of window sizes. The best prediction performance, the lowest MSE of 2.64E-4, is obtained based on the window size of 50 and GRU model with the test data. Thus, we use the GRU model in the TPE of TPANR method. The predicted traffic in the experimental demonstration based on GRU model is also shown in Fig. 3(a). It can be observed that the predicted traffic keeps in line with the real traffic in the data plane.

Exploiting the GRU based TPE, the TPANR method is also experimentally demonstrated and compared with the DCR2L and OSPF methods. In the comparison, the DRL agent of TPANR and DCR2L methods is configured as two hidden fully connected layers (256 units at first and 128 units at second hidden layers). The batch size is set to 20, and the capacity of replay memory is set to 1600 transitions. During the DRL agent training, an episode is set as 20 steps, and the training episode is up to 2000. Because the electrical switches can buffer all the packets, zero lost packet are measured in the experiment. Thus, the data plane network latency is set as the reward of DRL agent and the performance metric. The weighted factor $k^1{}_1$ =0.5, $k^2{}_1$= 0.3, $k^3{}_1$= 0.2, then the application of higher QoS priority is assigned a higher weight during the network reconfiguration. The DCR2L and TPANR methods are triggered when the network latency increases by 10%. Fig. 3(c) illustrates the average network latency of data plane with three different methods. It is shown that, compared with the DCR2L method, the TPANR method achieves a latency improvement of 15.4% when network traffic increases, exploiting one-step ahead traffic prediction. Meanwhile, compared with the classical method OSPF, the TPANR method reduces the network latency by 17.3%.

Conclusions

We have experimentally demonstrated an automatic DRL-based data center network reconfiguration method assisted by the traffic prediction. With the network traffic generated by realistic applications, GRU model performs the lowest MSE of 2.64E-4 with the test data. In the experiment, the TPANR method achieves a latency improvement of up to 17.3% compared with current methods.

References

[1] V. Abhishek and B. Neha, "A Review on Routing Information Protocol (RIP) and Open Shortest Path First (OSPF) Routing Protocol," International Journal of Future Generation Communication and Networking, vol. 9, pp. 161-170, 2016.

[2] X. Guo, F. Yan, X. Xue, B. Pan, G. Exarchakos and N. Calabretta, "QoS-aware data center network reconfiguration method based on deep reinforcement learning," Journal of Optical Communications and Networking, vol. 13, no. 5, pp. 94-107, 2021.

[3] G. Kaur and A. Bala, “OPSA: an optimized prediction based scheduling approach for scientific applications in cloud environment,” Cluster Comput, pp. 1–20, 2021.

[4] F. Tseng, X. Wang, L. Chou, H. Chao and V. C. M. Leung, "Dynamic Resource Prediction and Allocation for Cloud Data Center Using the Multiobjective Genetic Algorithm," IEEE Systems Journal, vol. 12, no. 2, pp. 1688-1699, 2018.

[5] S. Baig, W. Iqbal, J. L. Berral, A. Erradi and D. Carrera, "Adaptive Prediction Models for Data Center Resources Utilization Estimation," in IEEE Transactions on Network and Service Management, vol. 16, no. 4, pp. 1681-1693, 2019.

[6] A. Yu et al., "Scheduling with Flow Prediction Based on Time and Frequency 2D Classification for Hybrid Electrical/Optical Intra-Datacenter Networks," 2019 Optical Fiber Communications Conference and Exhibition (OFC), pp. 1-3, 2019.

[7] Cho K, Van Merriënboer B, Gulcehre C, et al. “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” arXiv preprint arXiv:1406.1078, 2014.