A Deep-reinforcement Learning Approach for SDN Routing Optimization

Chunlei Xu
State Grid Jiangsu Electric Power Company
Nanjing, China
chunleixu@sina.com

Weijin Zhuang*
China Electric Power Research Institute (Nanjing)
Nanjing, China
zhuangweijin@ epri.sgcc.com.cn

Hong Zhang
China Electric Power Research Institute (Nanjing)
Nanjing, China
zhanghong2@epri.sgcc.com.cn

ABSTRACT

In order to realize a real-time and customizable routing optimization for software-defined network (SDN), the deep reinforcement learning method is integrated to the routing process of SDN. In this paper, we design a novel routing optimization mechanism based on deep reinforcement learning. This mechanism is capable of reducing the network maintenance and operational costs via improving network performance (e.g. delay, throughput), and it can also realize black-box optimization in continuous time. In addition, we conducted a series of experiments to evaluate the proposed routing optimization mechanism. The experimental results demonstrated that our proposed routing optimization mechanism has good effectiveness and convergence, which can not only achieve more stable performance than the traditional routing mechanism, but also provide better routing schemes.

CCS CONCEPTS

• Networks • Network protocols • Network protocol designs

KEYWORDS

Deep reinforcement learning, Software Defined Network (SDN), Routing optimization

1 Introduction

Software Defined Network (SDN) is a new network architecture that adopts centralized control and distributed forwarding, which simplifies operation and maintenance management process, and provides more possibilities for the evolution and innovation of the network. It is considered as one of the architectures of the future network. However, with the increasingly fine network control and the expansion of network scale, more and more applications are emerged. The network traffic grows exponentially and the demand is becoming more diverse. The traditional routing algorithm based on shortest path algorithm has the problem of slow convergence speed, which is not suitable for dynamic network, and the response to network changes may lead to serious congestion. Therefore, optimizing the routing process of SDN is crucial to ensure the service quality and promote the development and evolution of SDN.

In recent years, machine learning (ML) has attracted much attention from many researchers due to its capability in intelligent decision-making and large-scale data processing. These advantages make it possible to be an effective weapon to solve the current deadlock between network maintenance and management [1, 2]. In order to avoid the disadvantages of traditional routing methods, many researchers try to introduce intelligent algorithms such as machine learning into SDN routing mechanism. Unfortunately, it is a huge challenge to design a routing mechanism that employs ML to realize real-time and customizable optimization.

Many intelligent approaches are embedded into the SDN routing process. Li et al. [3] proposed a routing pre-design scheme based on the collaborative approach of multiple ML mechanisms. This scheme firstly used appropriate clustering algorithms such as k-means or Gaussian mixture model to extract data features, and then it made use of supervised learning mechanisms like limit learning machine to predict the traffic demand. On this basis, in line with the weights of different constraint factors of dataflow, the author used analytic hierarchy process to complete an adaptive multipath routing mechanism. Some heuristic algorithms (e.g. ant colony and genetic algorithm) [4, 5] were used to route selection. However, the heuristic algorithms have some limitations, the model is only suitable for specific issues, and changes in the network status will affect the readjustment of parameters, which limits the scalability. [6] and [7] both deployed reinforcement-learning modules in SDN. Jang et al. [6] apply Q-learning [8] to realize QoS adaptive routing in SDN by using reward function of QoS awareness. The authors in [7] proposed a pure network-based architecture using SDN, which is designed for monitoring the global performance information of network, they also proposed a Q-learning-based dynamic bandwidth allocation algorithm to achieve QoE fairness.

CSAE 2020 October 20-22, 2020, Sanya, China

ACM ISBN 978-1-4503-7772-0/20/10 $15.00
https://doi.org/10.1145/3424978.3425004

Reinforcement learning has obvious advantages in routing applications, which can realize adaptive routing, high throughput and low latency. However, in the traditional Q-learning algorithm, due to the fine-grained control of dataflow by SDN, the maintenance of [state, action] to reward information Q table will bring a large amount of storage resource overhead. With the expansion of the scale of Q table, the time cost of table lookup is also a factor that cannot be ignored. This creates significant limitations on the use of reinforcement learning in SDN routing. In order to address this issue, Sendra S [9] used neural network instead of the traditional Q table in reinforcement learning, and adopted the DQN (Deep Q-learning) [10] to optimize the path selection process. Nevertheless, DQN is not easy to converge, and can only achieve discrete time control, which is not well matched with the dynamic characteristics of the network.

In view of the above situation and challenges, in this work, we focus on the use of a deep reinforcement learning for routing optimization. We integrate the deep reinforcement learning mechanism Deep Deterministic Policy Gradient (DDPG) [11] into the routing process of SDN, which can effectively avoid the storage resources and time overhead of lookup tables brought by the maintenance of large-scale Q tables. The proposed mechanism has the advantages of low latency, high throughput, and it can realize black-box optimization in continuous time. In addition, the experimental results demonstrate that DDPG optimized SDN routing method not only converges well, but also realizes better stability and performance than previous static routing methods such as OSPF (Open Shortest Path First).

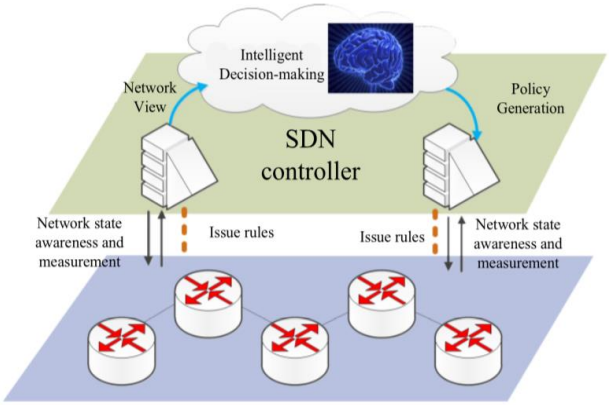

Figure 1: An SDN Framework based on Machine Learning.

2 Related Works

2.1 Combining SDN with Machine Learning

The structure of SDN integrated with machine learning is shown in Figure 1, that is, the intelligent decision-making layer based on machine learning is added at the SDN controller. Through the whole network view of SDN and network measurement technology, the intelligent decision-making layer can generate efficient policy, so as to achieve global, real-time and personalized intelligent network control. To be specific, on the basis of SDN, this framework is delivered through an "intelligent decision layer" that forms the network policy and then the control plane generates the corresponding flow table rules. It can be found that the architecture of the separation of forwarding and control in SDN network and the whole network view can well meet the demand of network intelligence, while the development of ML algorithms provides an opportunity for the realization of a "intelligent decision-making layer" which is lightweight and efficient.

2.2 Deep Reinforcement Learning DDPG

Reinforcement learning is the process by which agents learn the behavior of the environment in order to gain maximum reward value, while deep reinforcement learning is a learning method that combines the perceptual capabilities of deep learning with the decision-making capabilities of reinforcement learning. DeepMind proposed DDPG algorithm, which integrates deep learning neural network into DPG strategy learning method [12], as the basic algorithm of routing optimization. In the actor-critic structure of DDPG, the DPG is used as the actor module while the DQN is used as the critic module; each module is composed of the Target network for blocking the correlation of training data and the Online network for training. We assume that the Critic network is $Q(s,a,\omega)$, then the the Target Critic network is $Q(s,a,\omega^*)$; Likewise, the actor network is $\pi(s,a,\theta)$, then the the Target Actor network is $\pi(s,a,\theta^*)$. The parameters of DDPG are updated as follows:

(1) The Actor Module

The Actor module is used to update the policy functions of DPG. To evaluate the quality of the learned strategy, the policy objective function is defined as $J(\pi_\theta)$. Then, the purpose of strategy network is to maximize $J(\pi_\theta)$, as shown in Equation (1), $d^\pi(s)$ represents the distribution of the environmental state s. This model updates the parameters of the neural network in the direction of increasing the value of Q function. The gradient of $J(\pi_\theta)$ is shown in Equation (2).

$$\begin{aligned} J(\pi_\theta) &= \int_s d^\pi(s) Q\left(s, \pi_\theta(s)\right) ds \\ &= \mathrm{E}_{s\sim d_\pi}\left[Q\left(s, \pi_\theta(s)\right)\right] \end{aligned} \tag{1}$$

$$\nabla_\theta J(\pi_\theta) \approx \frac{1}{N}\sum_i \nabla_a Q^\pi(s,a)\big|_{s=s_i, a=\pi_\theta(s)} \nabla_\theta \pi_\theta(s)\big|_{s_i} \tag{2}$$

Where N represents mini-batch training data randomly sampled from the cache D.

(2) The Critic Module

The Critic module is utilized to approximate the Q value function, and the optimization objective is to minimize the Loss function. The Loss function is defined as the Equation (3). Then the parameters of neural network are updated using a gradient descent approach. It is worth noting that y_i is obtained by using the Target network to make the Critic network more stable and easier to convergence.

$$Loss = \mathrm{E}\left[\left(y_i - Q(s_i, a_i; \omega)\right)^2\right] = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - Q(s_i, a_i; \omega)\right)^2 \quad (3)$$

3 Algorithm

In this section, we deployed DDPG to SDN for SDN network routing optimization. The training flow chart of DDPG-based routing optimization algorithm proposed in this work is shown in Figure 2. There are three signals: action, state, and reward. Action a is taken to the environment by the agent with the goal of changing the weight of the network links, specifically, the path taken by the data flow of the source-destination node pair is changed by changing the weight of the network links. State s represents the Traffic Matrix (TM) of the current network load. Finally, reward r is related to the maintenance and operation strategy of the network, which can be an integrated strategy that balances multiple parameters or a single performance parameter (e.g. throughput, delay). By changing the settings of reward r, these control strategies can be adjusted. The Target network and the Online network have the same structure but different parameters. The parameters will be periodically pass to the Target network for updating from the online network, such as state s > state s'. In a training round, the information about each transformation process that interacts with the environment is stored in the experience pool, and by sampling and extracting some transformation process information from the experience pool, the samples of neural network is formed. The parameters of the online network are continuously updated through learning the training samples.

During the training process, DDPG updates parameters of the Actor network by the gradient of policy functions and parameters of the Critic network by the gradient of loss function, respectively. After that, soft update is used to update the parameters in the corresponding Target network, so that the Target Actor network and Target Critic network slowly learn the parameters from the Actor network and Critic network. To be specific, as shown in the Figure 2, the training process of the proposed algorithm can be described as follows: The parameters of the Actor and Critic network are initialized firstly; then accordingly the parameters of Target Actor and Critic network are updated; After initializing cache D, the Actor network selects action a by identifying state s, strategy π_θ and random noise; The environment routes the traffic based on the specified routing scheme according to a, the reward function is used to calculate r based on link utilization to return r and the new state s'; Then, (s, a, r, s') are stored in cache D for training the data set of neural network; N historical data (s, a, r, s') were randomly selected from cache D as data sets for training Actor network and Critic network; Computing the gradient of Loss function in the Critic network and the parameters ω are updated by using the Adam optimizer; Computing the gradient of policy functions in Actor network and the parameters θ are updated by using the Adam optimizer; Finally, the parameters of Target Actor network and Target Critic network are soft updated respectively until the parameters of Actor network converge to the optimal solution.

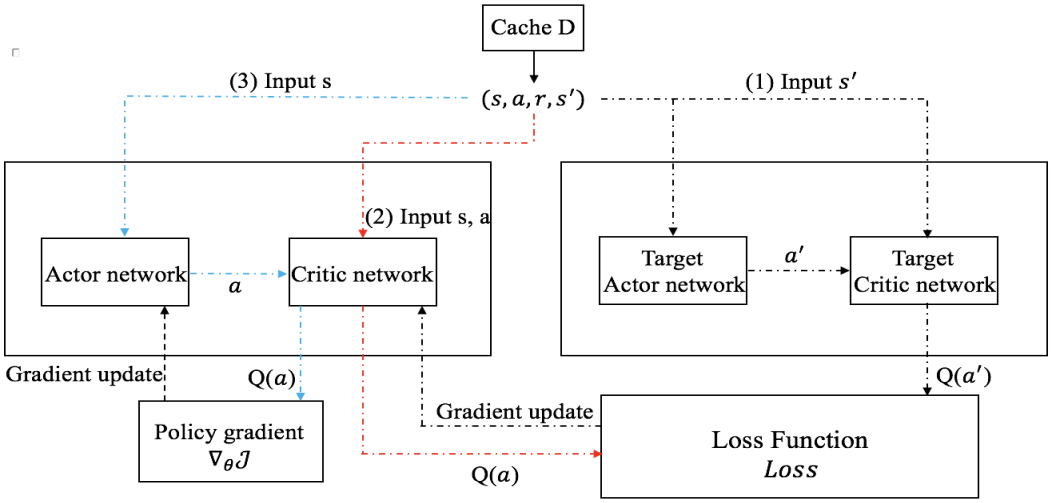

Figure 2: Overview of the Proposed SDN Routing Optimization Mechanism.

By running the proposed method, the SDN network can achieve optimization of customized performance parameters, such as minimizing delay, minimizing hops, and maximizing throughput. This is related to the selected network control strategy and the purpose of operation and maintenance of the network. Using DDPG to optimize SDN routing can achieve real-time control in continuous time, and it can realize black-box optimization that realizes automatic routing selection and effectively alleviates the pressure of operation and maintenance.

4 Experiments and Results

In this section, we conducted a series of experiments to evaluate the performance of DDPG optimal routing mechanism. We focus primarily on the effectiveness and convergence of DDPG agents and their performance advantages compared with other routing methods. We used a single NVIDIA GeForce RTX 2080 GPU, Tensorflow is adopted as the Deep Learning framework. We built the simulation network with OMNeT++ [13]. We tested the effectiveness and convergence of DDPG agents under the same topology. We also compare the performance of DDPG optimized routing with abundant randomly generated routing configurations and traditional mainstream routing protocol OSPF.

4.1 Experimental Setup

In the experiments, the backbone topology of Sprint network was adopted, which is composed of 53 links and 25 nodes. We assume that the bandwidth of link is the same. In order to approximate the scenario of actual network, multiple different Traffic Loads (TL) were set in the experiments, which account for different percentages of the total network bandwidth. Under the same TL, the gravity model [14] was used to generate a number of different traffic matrixes, that is, a number of differential TLs and distribution traffic matrixes are generated for training and testing.

Under different traffic intensity levels, the DDPG agent was trained in different steps, and the performance of the agent after training was tested to verify the convergence and effectiveness of the DDPG. In order to evaluate the performance advantages of DDPG optimized SDN routing, two sets of experimental schemes were designed: (1) Under different traffic loads, comparing DDPG optimized routing with the performance of a mass of randomly generated routing configuration. The performance of DDPG optimized routing is compared with the 100,000 randomly generated effective routing configurations, which are effective in this scenario because they are all accessible to each other without loops, the large quantity of comparative test data ensures the representativeness and validity. (2) In the same network environment, with the same weight of all links in the topology, we compared the routing performance of traditional OSPF protocol and DDPG.

4.2 Convergence and Effectiveness

Four different levels of TLs were used in the experiment: 100%, 70%, 40%, and 10% of the total network bandwidth, respectively. 250 traffic matrices were generated for each TL. We trained the DDPG agent with 2000, 4000, 10000, 20000, 50000, 80000 and 100000 times under different TLs by using the corresponding generative traffic matrices. Afterwards, we used 1000 traffic matrices as input to test the performance of DDPG agent and obtained the routing scheme selected by the agent after training. It is worth noting that given routing scheme and the traffic matrix, OMNeT++ simulation environment can obtain parameters of network performance (e.g. throughout, delay) as rewards for DDPG routing optimization mechanism. To ensure the validity of the data, the results of 1000 tests using the test flow matrix are averaged as a measure of the response training effect. The experimental results are shown in Figure 3.

The experimental results demonstrate that the DDPG agent can effectively reduce the network delay through training, and with the increase of training steps, the network delay is reduced and optimized. It is also proved that the performance of DDPG agent optimization network is constantly improved. As shown in Figure 3, under the traffic load of 70% network capacity, the optimization performance of DDPG agent training at 100,000 steps is 40.4% higher than that of training at 2000 steps, indicating that the proposed DDPG-based SDN optimization routing scheme has obvious effectiveness and good convergence.

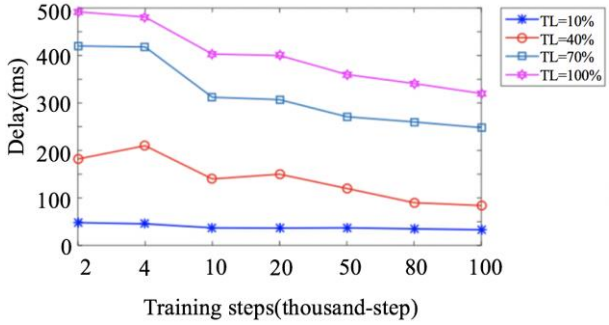

Figure 3: The Network Latency with Training Steps under Different Traffic Loads.

4.3 Performance Advantages

In order to evaluate the performance advantages of DDPG optimized SDN routing, we compared the performance of DDPG agents with abundant randomly generated routing configurations and OSPF routing algorithms.

Firstly, 100,000 randomly generated routing configurations are implemented into the test traffic matrix at each traffic load, and the corresponding delay is obtained through OMNeT++. Here, we used the DDPG agent with 100,000 training steps as the comparative baseline, and we compared the delay of random routing with the delay of routing strategy generated by DDPG routing optimization mechanism. The comparative results are shown in Figure 4 in the form of boxplots.

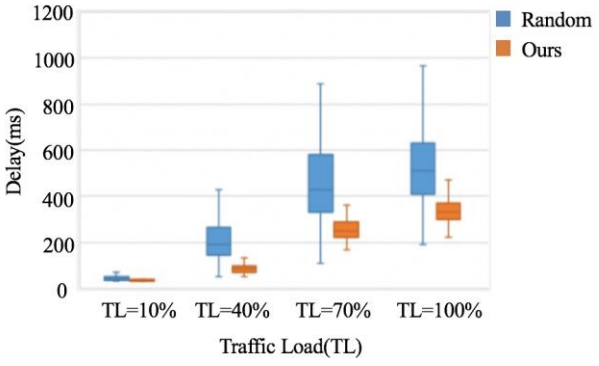

Figure 4: Results of Our Mechanism under Different Traffic Loads.

In Figure 4, the bottom and top of the rectangular module are on behalf of the lower and upper quartiles of the time-delayed observation data respectively. The middle horizontal line of the rectangle is on behalf of the observation data median. The top and bottom of a rectangular extension are on behalf of the minimum and maximum values of the observed data within the internal limits, respectively. In addition, invalid data outside the inner limit range are not displayed for simplicity. The experimental results show that the delay of most DDPG optimized routing configurations is smaller than the lower quartiles of a mass of randomly generated routing delays, and the minimum latency to optimize routing configuration by DDPG proxy is very close to the minimum delay obtained from a large number of random routing test sets. The experimental results fully validate the effectiveness and performance superiority of DDPG optimization SDN routing mechanism.
Then, under the condition that the traffic loads are the same and the link weight of the topology is set to the same value, the DDPG agent and OSPF are as the routing mechanism. The DDPG agent here has been trained for 100,000 times. The performance of DDPG optimized SDN routing mechanism is compared with that of traditional OSPF protocol, as shown in Figure 5.

In Figure 5, it presents the transmission delay of 8000 packets during the operation of two routing modes. Figure 5(a) shows the delay of packets when OSPF protocol is running. It can be found that the peak delay of most packets transmitted by OSPF protocol exceeds 300ms. While during the running time of DDPG optimized routing, as shown in Figure 5(b), only a few of the delay peaks exceed 300ms which indicates that the DDPG optimized routing scheme can achieve better performance than the traditional OSPF protocol. This result proves the effectiveness and performance advantages of our DDPG optimized SDN routing mechanism.

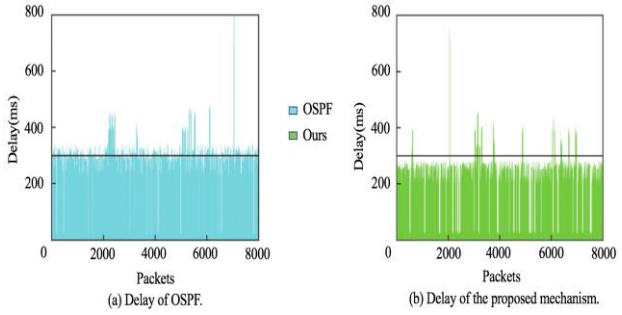

Figure 5: Network Latency under Different Traffic Loads.

5 Conclusions

In this work, a feasible architecture of ML mechanism for SDN is proposed, and DDPG deep reinforcement learning algorithm is added to SDN network to optimize the routing process. This method can realize black-box optimization in continuous time and simplify operation and maintenance process of the network. In addition, a series of experiments are conducted to evaluate the performance of DDPG-based optimized SDN routing mechanism. Experimental results show that the proposed mechanism in this paper has good convergence and effectiveness. Compared with the traditional and excellent routing protocol, it can provide more stable and superior routing schemes, which is of great practical value and significance for the operation and maintenance management of SDN network.

Future work. We will find an effective mechanism to reduce the training cost and cycle of agents. At present, the possible research directions are as follows: (1) Conducting online operation after the convergence of offline learning; (2) Increasing prior knowledge and speeding up the training process of transfer learning and imitation learning.

ACKNOWLEDGMENTS

This paper is partly funded by National Key R & D Program of China (2017YFB0902600) Research and Application of Key Technology for Intelligent Dispatching and Security Early-warning of Large Power Grid, Science and technology project of State Grid Corporation of China (SGJS0000DKJS1700840).

REFERENCES

[1] Boutaba R, Salahuddin M A, Limam N, et al. (2018). A comprehensive survey on machine learning for networking: evolution, applications and research opportunities[J]. Journal of Internet Services and Applications, 9(1), 16.
[2] Fadlullah Z M, Tang F, Mao B, et al. (2017). State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems[J]. IEEE Communications Surveys & Tutorials, 19(4), 2432-2455.
[3] Li W, Li G and Yu X. (2015). A fast traffic classification method based on SDN network[J]. Proc. 4th Int. Conf. Electron., Commun. Netw. 223-229.
[4] Wang F, Liu B, Zhang L, et al. (2017). Dynamic routing and spectrum assignment based on multilayer virtual topology and ant colony optimization in elastic software-defined optical networks[J]. Optical Engineering, 56(7), 076111.
[5] Parsaei M R, Mohammadi R and Javidan R (2017). A new adaptive traffic engineering method for telesurgery using ACO algorithm over Software Defined Networks[J]. European Research in Telemedicine/La Recherche Europeenne en Telemedecine, 6(3-4), 173-180.
[6] Lin S C, Akyildiz I F, Wang P, et al. (2016). QoS-aware adaptive routing in multi-layer hierarchical software defined networks: A reinforcement learning approach[J]. 2016 IEEE International Conference on Services Computing (SCC). IEEE, 25-33.
[7] Jiang J, Hu L, Hao P, et al. (2018). Q-FDBA: improving QoE fairness for video streaming[J]. Multimedia Tools and Applications, 77(9), 10787-10806.
[8] Sutton R S and Barto A G (2011). Reinforcement learning: An introduction[J].
[9] Sendra S, Rego A, Lloret J, et al. (2017). Including artificial intelligence in a routing protocol using software defined networks[J]. IEEE International Conference on Communications Workshops (ICC Workshops). IEEE, 670-674.
[10] Mnih V, Kavukcuoglu K, Silver D, et al. (2015). Human-level control through deep reinforcement learning[J]. Nature, 518(7540), 529.
[11] Lillicrap T P, Hunt J J, Pritzel A, et al. (2015). Continuous control with deep reinforcement learning[J]. arXiv preprint arXiv:1509.02971.
[12] Silver D, Lever G, Heess N, et al. (2014). Deterministic policy gradient algorithms[J].
[13] Varga A and Hornig R (2008). An overview of the OMNeT++ simulation environment[J]. Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 60.
[14] Roughan M (2005). Simplifying the synthesis of Internet traffic matrices[J]. ACM SIGCOMM Computer Communication Review, 35(5), 93-96.