Research on deep reinforcement learning multi-path routing planning in SDN

Zheng Wang*, Zhengyong Lu, Chuanhuang Li
Zhejiang Gongshang University, Hangzhou, China
378245336@qq.com

Abstract—Based on the research of current SDN traffic scheduling technology, combining the advantages of reinforcement learning in strategy optimization and the characteristics of SDN network resource centralized control, and using the neural network of depth learning to fit the Q-value table, this paper proposes an intelligent multi-path routing planning method based on depth reinforcement learning. Experiments show that the method can use the characteristics of SDN to route the network traffic in multi-path according to the current state information and traffic characteristics of the network; using the advantages of reinforcement learning, it can find multiple forwarding paths for different flows that conform to their traffic characteristics, and improve the utilization rate of the network link bandwidth; and using the neural network of deep learning to fit the Q value in the traditional reinforcement learning algorithm Table.

1. INTRODUCTION
In recent years, the rapid development of the Internet makes the rapid growth of network transmission business data. The Internet industry needs a new network architecture to solve the existing network problems, and the architecture needs to be more flexible and efficient than the traditional architecture to meet the growing demand of social traffic data.

Based on the above problems, software defined Networking (SDN) came into being. The concept of SDN was first proposed by the clean slate research group of Stanford University in the United States[1], by separating the control plane of network equipment from the data plane, and opening the programmable ability of network, thus realizing the flexible distribution of network traffic. With the development of SDN technology, its flexible and efficient features in the network attract the active participation of IT industry, such as Google, Facebook and other enterprises actively promote the deployment and transformation of SDN. Up to now, SDN has been put into practice. For example, Google and B4 network have successfully deployed SDN technology, which completely relies on OpenFlow to plan the flow path[2], solving the problem of flow scheduling. On the one hand, it makes the global flow control more efficient, on the other hand, it reduces the cost of network maintenance.

Routing is an essential part of both traditional networks and SDN networks. At present, Dijkstra (shortest path) algorithm is used in the mainstream SDN routing module. If all packets only depend on the shortest path algorithm, it is easy for data flow to cause link congestion due to selecting the same link, and other links will be idle, which greatly reduces Link utilization. On the other hand, the shortest path algorithm is the algorithm to find the shortest path in graph theory. When the algorithm is running, it will actually find the shortest path from the source node to all other nodes in the topology[3]. Therefore, the time complexity of the algorithm is very high. In addition, there are also some protocols supporting multi-path, such as ECMP[4], but these protocols do not consider the
quality of service requirements of different business flows. Therefore, in SDN network, a better routing strategy is needed to generate routes, improve the performance of the network, and ensure the quality of service of different traffic flows.

Deep reinforcement learning (DRL) will promote deep learning. DL is combined with reinforcement learning to learn and make decisions directly from high-dimensional raw data, which is more suitable for control and decision-making tasks. DL is responsible for extracting features of input environmental state and realizing the fitting of environmental state to action value function; RL is responsible for completing decisions according to the output of deep neural network and certain exploration, and realizing the mapping of state to action [5]. In recent years, DRL has made remarkable achievements in man-machine game, machine vision and other fields. For example, alphago go robot has defeated the best human opponent in the field of go. DRL is different from RL, RL is limited to low dimensional space, state action space is usually small and needs manual design features, DRL can intelligently adjust reward function, and use neural network to fit Q value table. When Q value table is large, DRL can save a lot of storage cost and search time cost.

Based on the characteristics of centralized control and easy access to global resource information of SDN network, and the advantages of deep reinforcement learning, a deep reinforcement learning multi-path routing algorithm based on SDN is proposed.

2. RELATED WORK

Hoang Hu [6] proposed a dynamic routing algorithm based on the maximum Association pair model to find a path from the beginning to the end in the unknown environment. Compared with the traditional dynamic routing algorithm, the experimental results show that the algorithm has a higher search efficiency.

Lin Yafei [7] and others optimized the traditional shortest path algorithm by improving the data storage structure of the algorithm. The optimized algorithm adopts the storage structure of object-oriented method type, and the mapping relationship between the storage location and key attributes of each object is established. When searching, only according to the key attributes of the object, the object to be searched can be obtained, thus reducing the time cost of searching.

Yan [8] and others proposed a multi-path QoS system based on SDN -- hiqos, which is divided into two parts: differentiated service component and multi-path routing component. In the differentiated services component, the controller distinguishes different services by identifying the IP address of the source node, and provides service guarantee for different services through the queuing mechanism on the SDN switch; the multi-path service is realized by monitoring the network status in real time and finding multiple paths that meet the QoS constraints. However, this method is still based on the idea of shortest path algorithm when implementing multi-path algorithm, so when the network topology is large, the efficiency of this method is low.

Due to the imbalance of network traffic distribution, the possibility of network link congestion is increased. In order to obtain better network performance and make full use of redundant paths in the network, a series of work has studied the multi-path routing algorithm in SDN, which can distribute the traffic to multiple paths reasonably and realize the load balance of the network.

Nonghuangwu [9] and others proposed a multi-path routing algorithm based on SDN for load balancing. The algorithm obtains the load information of the global link traffic through the controller. When a new flow enters the system, the controller will calculate all the repeatable paths between the source and destination nodes, then select a path with the least link load, and send the flow table to the router. This method can make the optimal load balance routing decision for each data flow entering the network. However, in the real network environment, the network state is very complex, and the calculation of this method is large.

At the same time, with the increasing demands of network services on the quality of service, the routing planning strategy to meet QoS constraints has been a hot topic in academic and industrial circles. In view of the demand of different traffic for quality of service, there are many researches related to QoS level.
Bueno [10] et al. Proposed a QoS policy configuration framework, which can provide end-to-end network services on demand according to the specific needs of online interactive applications, avoid excessive allocation of resources to business needs, improve the utilization of network resources, and ensure the quality of service.

Tomovic [11] and others proposed a QoS aware algorithm based on SDN, which can control QoS flow by using different routes for QoS flow and non QoS flow. In this algorithm, QoS routing takes bandwidth as the constraint and substitutes it into the shortest path algorithm to find the path. When the network cannot meet the quality of service requirements, non QoS flows are rerouted. The algorithm does not consider other QoS constraints, and does not further divide the QoS flow, so it has limitations.

In recent years, reinforcement learning has been paid more and more attention by researchers. It makes the best decision by exploring the environment. Cheng Cheng [12] proposes a routing planning algorithm based on reinforcement learning, which can effectively select the optimal path. However, when routing, all packets are treated in a unified way without differentiated services, and different types of packets should have different priority levels.

Deep reinforcement learning combines deep learning with reinforcement learning, which makes intelligent physical ability to make judgments in a way closer to people's thinking. Deep reinforcement learning is more and more used in games, robot control and other fields. In the field of traffic engineering, the traffic scheduling strategy based on deep reinforcement learning has also been preliminarily developed.

Stampa [13] and others designed and evaluated a deep reinforcement learning model in SDN network. Experiments show that this method has good performance. Compared with the traditional optimization algorithm, it can adapt to the current traffic situation and reduce the network delay. However, the QoS level of traffic is not considered in routing planning.

To sum up, the existing SDN routing planning methods still have the problems of low efficiency and high computational complexity. In view of the above problems, a multi-path routing planning algorithm based on deep reinforcement learning is proposed. The algorithm applies reinforcement learning to SDN traffic scheduling, and uses the advantages of neural network in feature extraction to fit the Q-value table of reinforcement learning.

3. MULTI-PATH ROUTING PLANNING BASED ON DEEP REINFORCEMENT LEARNING

3.1 Deeply Strengthen The Learning Of Routing Planning Framework

Reinforcement learning can get the maximum reward by learning the optimal path through continuous trial and error interaction with the environment, but it depends on the quality of artificial design features and has limitations. Deep reinforcement learning is a combination of multilayer neural network model and reinforcement learning model. It uses reward value of reinforcement learning to construct learning label, and replaces Q value table of reinforcement learning by neural network. According to the characteristics of SDN architecture and the advantages of deep reinforcement learning in feature learning and decision-making, this paper proposes a route planning method based on deep reinforcement learning to fit the Q-value table of reinforcement learning. The specific architecture is shown in Figure 1.
In the whole system architecture, the control layer mainly includes four parts: data collection module, data processing module, routing decision module and flow table information processing module. Through the data collection module to the data plane network traffic information and the current network topology information, when the data collection module collects a piece of data flow information that needs to be planned, it will input it into the data processing module for feature extraction, then input it into the routing decision module to implement the traffic scheduling algorithm, and finally through the flow table information processing module to the routing decision module. The execution result of is processed and distributed to the OpenFlow switch.

In this system architecture, the routing decision module is the core of the whole architecture. The routing decision module contains deep reinforcement learning neural network, which can obtain the environmental information of neural network from the data processing module, extract the characteristics of environmental information through deep learning neural network, and provide the status information of current environment as the decision input of reinforcement learning; deep reinforcement Learning neural network uses reinforcement learning to complete the decision-making function. It can map the current state into actions based on the corresponding strategies, and evaluate various action values. Finally, the Q-value table generated by reinforcement learning can be fitted into corresponding functions as the basis of routing decision.

Similar to the idea of multi-path planning based on reinforcement learning, in the routing planning technology based on deep reinforcement learning, the method of judging whether the traffic has been planned is used to generate multi-path to achieve high utilization of link bandwidth. After the data collection module collects a traffic forwarding demand, it inputs it to the data processing module to get the traffic information quadruple $flow = [s, d, \beta, B_{\min}]$, and the network topology information T matrix, then input the result to the routing decision-making module to get the optimal path, judge whether the optimal path can complete the forwarding of the flow, if it can end the routing decision; otherwise, divide the flow, then continue to execute the routing decision-making module to get the suboptimal path, and repeat the above process until the flow is allocated or No path forwarding for. Finally, the output of the routing decision module is input to the flow table information processing module to process and send the flow table to the router to complete the routing decision of the flow.

3.2 Model Algorithm Research

In this paper, a multi-path routing planning algorithm based on deep reinforcement learning is designed as follows:
SDN multipath routing algorithm based on deep reinforcement learning

S ← start point, D ← end point, β ← QoS level,
B available ← minimum available bandwidth
of current link, bmin ← traffic size, routing (s, d)
← route set, t ← current topology
while Bmin > 0
Packet initialization status s, initialize memory
pool memory _ size, set observation value (data
amount
in memory pool)
Loop traversal:
A ← the action selected by the data package,
execute action a to get r, s
Save s, a, r, s' to memory pool
If memory pool data enough
eval_ Net take out sample
calculation Q(s, a; \theta)
Every N steps from Eval_ Net to
target_ Net
target_ Net take out sample calculation
max_a Q(s, a; \theta^-)
Training neural network to get route
break
else
if s' == d
s ← starting point
else
s ← s'
Routing(s, d) ← Routing
Two nodes Si, SJ of the minimum available
bandwidth in routing
If B available <= bmin
T[si][sj] = 0, to update T
B_{min} ← B_{min} - \beta \sum_{i\neq j} B_{min}

The training of neural network is an optimization problem. In this paper, mean square error (MSE) is used as the loss function. The loss function is shown in formula (1):

\[ L(\theta) = E[(r \text{ arg et } Q - Q(s, a; \theta))^2] \quad (1) \]

The loss function indicates how close the predicted value of the sample is to the actual value. The lower the loss function value is, the closer the two values are. Where, \( \theta \) is a neural network parameter, \( s \) is the current status of the packet, \( a \) is the corresponding forwarding action, and TargetQ is the target Q value, as shown in formula (2):

\[ r \text{ arg et } Q = r + \gamma \max_a Q(s', a'; \theta^-) \quad (2) \]

Among them, \( r \) is the reward, \( s', a' \) are the state and action of the next step, \( \gamma \) is the discount rate. These parameters are determined according to the Q-learning update formula, and \( \theta^- \) is the parameter of the neural network that is not updated.
3.3 Experimental Simulation Design

The experimental system environment is Ubuntu 14.04, using Ryu controller and mininet to test the effectiveness of the routing scheme. Use mininet to build the network topology as shown in Figure 2, including 9 openflow switches and 5 hosts.

![Figure 2 SDN network topology](image)

The multi-path routing algorithm proposed in this paper uses the Markov decision process (MDP) to model. Therefore, the model MDP quadruple proposed in this paper is defined as follows:

1. State set: in the network topology, each switch represents a state. Therefore, according to the network topology, this paper defines the network state set as follows:

$$S = [s_1, s_2, s_3, \ldots, s_9]$$

2. Action space: in SDN network, the transmission path of packets is determined by the network state, that is, packets can only be transmitted at the connected network nodes. According to the network topology, this paper defines the network connection state as shown in formula (3):

$$T[s, s'] = \begin{cases} 0, & s, s' disconnected \\ 1, & s, s' connected \end{cases}$$

(3)

Because packets can only be transmitted in the connected network nodes, the action set of each state $s_i \in S$ can be defined according to the network state set and network connection state in this paper as follows:

$$A(s_i) = \{ s_j \mid T[s_i, s_j] = 1 \}$$

Indicates that when the current state is $s_i$, the optional action set of the state is represented as the node $s_j$ directly connected with $s_i$ on the network topology, that is, the next hop state of the current state $s_i$ will only select the state $s_j$ connected with the state. For example, the action set of state 1 is $A(s_1) = \{ s_2, s_3 \}$

3. State migration: in each round of training, when the packet is in the state $s_i$, after selecting an action in the action set, if the action is not in the selected state of the round, the packet will move to the next state.

4. Reward function: in the system model of this paper, each packet passing through a switch will get a negative reward, which indicates the cost of packet forwarding. The more switches it passes through, the more negative rewards it will accumulate, and the higher the cost will be. In order to increase the bandwidth utilization rate of the link and encourage the packet to select the link with high bandwidth utilization rate, the more each packet passes through a handover in order to force the packets to arrive at the destination node as soon as possible, when the packets arrive at the destination node, they will be rewarded with an additional size of 1, which is expressed by formula (4):

$$r(s_i, a) = \begin{cases} 0, & j \text{ is the destination node} \\ 1, & j \text{ is the non destination node} \end{cases}$$

(4)
4. EXPERIMENTAL RESULTS AND ANALYSIS

In this paper, the idea of using neural network to approach the Q-value table of reinforcement learning is adopted in the multi-path routing planning algorithm based on deep reinforcement learning. Therefore, this paper needs to carry out experiments to observe the fitting degree of neural network to the Q-value table of reinforcement learning. An ideal model result should be: the loss function value gradually decreases during training, and finally approaches 0. In this paper, by comparing the convergence performance and error of different learning rate, neural network layer number and neuron number models, the optimal parameters are selected. The final experimental results are shown in Figure 3.

![Figure 3 convergence of loss function of neural network](image)

In Figure 3, the x-axis represents the number of neural network training steps, and the y-axis represents the loss function value of neural network. According to the observation graph, the loss function of neural network decreases with the increase of training steps, and the loss value drops rapidly, and the loss value of neural network is close to 0 finally. The above experimental results show that the algorithm proposed in this paper can fit the reinforcement learning q-table well, so as to expand the reinforcement learning state action space. It solves the problem that reinforcement learning can't play its advantages because of many network states in the real network.

In order to measure the performance of the multi-path routing algorithm based on deep reinforcement learning proposed in this paper, this paper compares it with other routing planning algorithms based on deep reinforcement learning. In paper [14], a network traffic scheduling algorithm based on deep reinforcement learning is proposed. The main idea of this algorithm is to set the reward function. When the network traffic bandwidth loss is 0, the reward value is It is the minimum available bandwidth of all links in the network, which makes the distribution of network resources more balanced when dqn does not lose the traffic bandwidth; when the network traffic bandwidth loss is greater than 0, the reward value of algorithm reward function is the negative value of the traffic bandwidth loss, so that the network can reduce the traffic bandwidth loss as much as possible, so as to optimize the routing and achieve load balance.

In order to evaluate the performance of the two algorithms more effectively, this paper includes the loss of traffic bandwidth into the evaluation index of the algorithm. This paper defines the loss rate of traffic bandwidth as the quotient of the sum of the bandwidth loss of all traffic and the bandwidth demand of all traffic when the traffic is forwarding, as shown in formula (5):

\[
\text{Loss rate} = \frac{\sum B_{\text{loss}}}{\sum B}
\]

(5)
In the formula, $B'_i$ represents the bandwidth demand of the i-th traffic, $B'_{loss}$ represents the bandwidth loss of the i-th traffic, and the definition formula of the traffic bandwidth loss is shown in formula (6):

$$B'_{loss} = B'_i - \min_{\beta} R^i_\beta$$  (6)

In formula (6), $R^i_\beta$ represents all link bandwidth of the forwarding route selected by the i-th traffic, $\min_{\beta} R^i_\beta$ is the minimum link bandwidth of the route.

The experimental topology is shown in Figure 4. The bandwidth of each link is set to 200, in which each host sends traffic to other hosts with a probability of 20%. The traffic demand size sent is 30, and all hosts send 30 traffic in total. In order to simulate the network state in real environment, this experiment will also compare the loss rate of traffic bandwidth under different congestion of network links. The experimental results are shown in Figure 4.

In Figure 4, m-DQN represents the multi-path routing planning algorithm proposed in this paper, DQN represents the routing planning algorithm for comparison, X axis represents the congestion degree of the network topology link bandwidth, Y axis represents the loss rate of traffic bandwidth, the lower the loss rate of traffic bandwidth, the better the performance of the algorithm. As can be seen from the above figure, with the increase of the congestion degree of the network topology link bandwidth, the loss rate of the traffic bandwidth of the two algorithms also shows an upward trend. This is because the available bandwidth of the network link bandwidth determines the transmission capacity of the network itself, that is, the larger the available bandwidth of the network, the less the traffic loss, but it will not exceed the upper limit of the available bandwidth of the network. It can also be found from the above figure that the multi-path routing algorithm proposed in this paper can increase the network bandwidth utilization and reduce the bandwidth loss of traffic to a certain extent. From the essence of the algorithm, the comparison algorithm DQN only selects a path with large available bandwidth to forward the route. When the available bandwidth of the network topology link is small, if a data with large traffic bandwidth demand is to be forwarded at this time, it will cause great bandwidth loss. Moreover, the comparison algorithm DQN does not consider the QoS level of traffic, and all flows are treated uniformly, which may cause priority The small flow forwarding path is short, so the delay time is small; while the high priority flow forwarding path is long and the delay is large, which is obviously unreasonable in the real network.

5. Conclusion

In this paper, neural network and reinforcement learning are combined to propose a multi-path routing planning algorithm based on depth reinforcement learning. Experiments show that after a certain period of training, the algorithm can provide different services with routes that meet their QoS level, and when the link bandwidth is not enough, a large flow is divided into several small flows, so as to improve the link bandwidth utilization. The algorithm uses neural network to train the approximate
function to fit the Q value table. The experiment shows that after a certain period of training, the loss value of neural network is close to 0.

ACKNOWLEDGMENT
This work was supported by the Natural Science Foundation of China (61871468), the Zhejiang Provincial Natural Science Foundation of China (LY18F010006), Zhejiang Provincial Key Laboratory of New Network Standards and Technologies (NNST)(No.2013E10012), Zhejiang Provincial Key Research and Development Program of China (No.2020C01079).

REFERENCES
[1] Mckeown N, Anderson T, Balakrishnan H, et al. OpenFlow: Enabling innovation in campus networks[J]. ACM SIGCOMM Computer Communication Review, 2008, 38(2):69-74.
[2] Jain S, Kumar A, Mandal S, et al. B4: experience with a globally-deployed software defined wan[C]/ 2013.
[3] Yinghua Min. Survey of computer network routing[J]. Journal of Computer Science,2003,26(6):641-649.
[4] Zhang H, Guo X, Yan J, et al. SDN-based ECMP algorithm for data center networks[C]// 2014 IEEE Computing, Communications and IT Applications Conference (ComComAp). IEEE, 2014.
[5] Yao Dong, Yingying Ge, Hongyong Guo, Yongfeng Dong, Chen Yang.Path planning for mobile robots based on deep reinforcement learning [J]. Computer engineering and application,2019,55(13):15-19+157.
[6] Hong H,Sang H, Tae C. Extended Dyna-Q Algorithm for Path Planning of Mobile Robots[J].Journal of Measurement Science and Instruction, 2011, 2(3): 283-287.
[7] Yafei Lin, Huilin Liu. Design of routing algorithm based on OpenFlow[J], Academic talk about fund projects, 2015, 9: 30-32.
[8] Yan Jinyao, Zhang Hailong, Shuai Qianjun. HiQoS: an SDN-based multipath QoS solution[J]. China Communications, 2015, 12(5):123-133.
[9] Huangwu Nong, Chuanhe Huang, Xiaopeng Huang.Multi path routing algorithm for SDN based fat tree data center networks [J]. Computer science, 2015, 43(6):32-34.
[10] Bueno I, Aznar J I, Escalona E, et al. An OpenNaaS based SDN framework for dynamic QoS control[C]/ Future Networks and Services (SDN4FNS), 2013 IEEE SDN for. IEEE, 2013.
[11] Tomovic S, Prasad N, Igor Radusinović. SDN control framework for QoS provisioning[C]/ Telecommunications Forum Telfor (TELFOR), 2014 22nd. IEEE, 2014.
[12] Cheng Cheng. Research on reinforcement learning based routing planning algorithm in SDN [D], Zhejiang Business University, 2018
[13] Stampa G, Arias M, Sanchez-Charles D, et al. A Deep-Reinforcement Learning Approach for Software-Defined Networking Routing Optimization[J]. 2017.
[14] Qingyang Houyu, Jicheng Quan, Hongwei Wang. A review of the development of deep learning [J]. Naval electronic engineering, 2017, 37(4):5-9.