The Intelligent Routing Control Strategy Based on Deep Learning

Jinsuo Jia[1](✉), Yichun Fu[2], Guiyu Zhang[1], Xiaochen Liang[1], and Peng Xu[1]

[1] China Information Technology Designing and Consulting Institute Co., Ltd., Beijing 100048, China
{jiajs5,zhanggy,liangxch,xupeng}@dimpt.com

[2] International School, Beijing University of Posts and Telecommunications, Beijing 100876, China
2016213048@bupt.edu.cn

Abstract. The rapid development of computer hardware and software provides a suitable platform for machine learning, in which deep learning has become a breakthrough in machine learning technology in various fields in many disciplines. Some recent research efforts have focused on routing control based on deep learning. Therefore, this paper studies the problem of intelligent routing, and aims to propose an intelligent control strategy based on deep learning with the help of Software-Defined Network (SDN) and other new network technologies. The characteristics of SDN network that can easily obtain the network topology have laid the foundation for selecting different routing paths according to the different QoS levels of the flow. Nowadays, the routing modules in commonly used SDN controllers use the shortest path algorithm which is simple to implement and works effectively. However, the best path calculated by controllers may suffer from huge traffic load and result in congestion, and the controllers cannot learn from the previous experiences to intelligently switch to other paths. This paper present intelligent routing control strategy based on Deep Q-Learning (DQN) in SDN, which uses the Openflow to collect information from the network, and aggregates them to the SDN controller, and then uses DQN to generate the specific routing for forwarders.

Keywords: SDN · Deep learning · Routing strategy · Reinforcement learning

1 Introduction

Nowadays, the Internet has become a very important part of people's lives. Driven by strong market demand and the development of modern technology, the Internet has developed more and more rapidly which has been gradually inseparable from other industries. The initial design of the Internet was not considered that it will be so widely used in the future, and its role will be as important as this. Therefore, many designs at that time could no longer adapt to

B. Li et al. (Eds.): IoTaaS 2020, LNICST 346, pp. 585–598, 2021.
https://doi.org/10.1007/978-3-030-67514-1_46

the development trend of modern Internet [9,10]. At present, security, robustness and mobility have become the major problems of the Internet. In recent years, Software Defined Network (SDN) has been sought as the most likely solution of future network, and it has two characteristics: the control plane and data plane are separated, and the network is programmable [1]. Therefore, SDN can provide more effective configurations, better performance and higher flexibility to adapt to the future development of the network [3].

In 2017, the Alphago also announced the arrival of the AI revolution. The term "artificial intelligence" was originally born at Dartmouth society in 1956, and AI has become the hottest research direction in the field of computer currently due to the progress of algorithms and hardware. As the learning algorithm of artificial intelligence, Machine Learning (ML) and Deep Learning (DL) are also accepted to extensive attention and research.

As a branch of machine learning, reinforcement learning is the product of the combination of multidisciplinary. In the field of artificial intelligence, agent is generally used to represent an object with behavior ability, such as robot, unmanned vehicle, human and so on. The problem of reinforcement learning is the task of interaction between agent and environment. The essence of reinforcement learning is a "decision-making" problem, i.e., agent makes a decision by itself. Deep learning is an algorithm proposed after the development of "artificial neural network", which can be regarded as the evolution of traditional neural network. Deep Learning is also a new field in the process of machine learning, and it emphasizes the importance of model depth and feature learning. Compared with the traditional neural network, deep learning has similar structure but obvious improvement. The essence of deep learning is to build a neural network model with multiple hidden layers, and learn more useful features through massive training data, so as to ultimately improve the accuracy of model classification or prediction.

This paper introduces an intelligent SDN routing optimization method based on traffic classification, which is based on the characteristics of SDN, and combined with the advantages of reinforcement learning in strategy optimization and deep learning in data classification: when a network flow reaches a node in the network (Openflow switch), it can use the deep learning model to classify the network flow, and then according to the classification results to judge the QoS level, and then uses reinforcement learning model to plan its forwarding path, so that the flow can reach the destination node (Openflow switch) through the shortest path that meets its QoS requirements [4–8].

The concept of network layered architecture is the key to the rapid development of traditional Internet. However, in recent years, the scale of the network is getting larger and larger, and the traditional network equipment with closed network uses too many complex protocols, which is more and more difficult for major operators to optimize the network, and also difficult for researchers to deploy new protocols in the real environment. At the same time, with the continuous expansion of network scale, the rapid growth of traffic, the emergence of various new services, the improvement of network operation and maintenance costs, the traditional network architecture is difficult to meet the rapid development of network needs.

In order to meet the needs of the next generation of network development, Mckeown proposes the concept of SDN which is a kind of network architecture that the programmable control plane is separated from the data forwarding plane. The control plane is a programmable controller [2]. The R&D personnel and operators can customize the network or deploy new network protocols according to the needs of customers or their own products. The data layer is composed of some Openflow switches, which are different from the traditional two-layer switches. They only provide simple data forwarding function, quickly match data packets, and complete data forwarding to meet the growing network demand.

The most classic description of SDN architecture is the architecture diagram from ONF (Open Network Foundation).

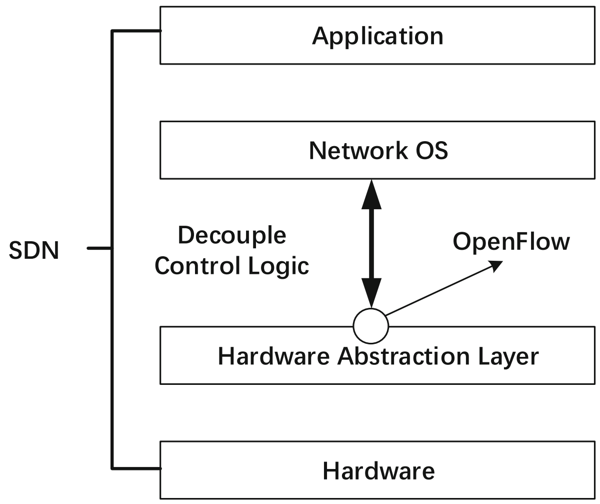

Fig. 1. SDN architecture diagram from ONF.

Figure 1 expresses the concept of hierarchical decoupling of SDN, including the general basic hardware layer, hardware abstraction layer, network operating system, and upper-layer applications. And the basic hardware and hardware abstraction constitute the physical network equipment which is the data forwarding level in the SDN architectures. The network operating system and upper-layer applications form the control level. The data forwarding level and the control level are decoupled with a standardized interactive protocol such OpenFlow. This decoupling architecture shows that the network operating system and network applications (such as routing control protocols, etc.) do not have to run on physical devices, but can run in external systems (such as X86-based servers), thereby achieving flexible and programmability of the network.

At present, the mainstream SDN controllers, such as Ryu, provides modules to complete packet forwarding, and the basic algorithm used is Dijkstra (shortest path) algorithm. Dijkstra algorithm finds the shortest path from the start node to the destination node of packet forwarding every time. However, if the forwarding of all packets only depends on the shortest path algorithm, it will bring a serious problem. The data flow is easy to gather because of choosing the same forwarding path, which greatly reduces the link utilization, and also easily leads to network congestion. In view of the shortcomings of pure Dijkstra algorithm, there are many researches related to SDN routing, and a variety of Dijkstra improved algorithms or new routing algorithms are proposed.

2 Related Work

2.1 Overview of SDN Architecture

The definition of SDN is divided into two types, in narrow sense, SDN refers to the new network architecture based on Openflow proposed by Stanford University. This new network architecture is derived from Ethane project of Stanford University, which has produced the concept of Openflow and has evolved a new network structure called SDN. In the initial SDN network architecture, OpenFlow was its core technology. With the development of the CleanStage project, the SDN network architecture expanded to separate the control plane and data plane of network devices, so as to flexibly control network traffic, develop new network applications on the platform of SDN, and make innovation to network applications. The SDN technology in narrow sense has the following characteristics:

(1) Separation of control plane and forwarding plane: traditional network equipment constitutes the forwarding plane and only forwarding tasks are performed, SDN controller constitutes the control plane to control the forwarding plane.
(2) Centralized control of the network, the SDN controller can obtain the entire network topology of the forwarding plane and the local state of each network device, the SDN controller can achieve centralized control of the entire network with this information.
(3) Open interface: The SDN controller develops different interfaces for the application layer and the device layer, then the application layer and the device layer are logically independent of each other. In this case, when we develop network applications, we just don't need to know the situation of the equipment layer.

As a new type of network technology, the SDN technology based on Openflow has many incomplete places, but shortly after its birth, it has become the top ten cutting-edge technology of MIT. It can be seen from this that the international society is optimistic about the prospects of SDN, and discussions on SDN are also in full swing. At present, the international standards organizations that focus on

SDN mainly include: ONF (Open Network Foundation), Open Daylight, ACM SICGCOM HotSDN Workshop, IRTF SDNRG and IETF.

The SDN architecture based on the Openflow specification was first proposed by ONF which is established in 2011 with objective to formulate SDN related standards and promote its proposed architecture. So far, ONF has proposed the Openflow standard and the Openflow configuration protocol. These work have accelerated the process of SDN standardization, and promoted the rapid development of SDN. The ultimate goal of ONF is to turn SDN into ordinary software, so that SDN can be deployed on general servers and common operating systems. In this way, ordinary programmers can implement network programming on SDN. The SDN architecture are composed of application layer, the control layer and the infrastructure layer. And for the different needs of the data center, access network and bearer network, different SDN architectures are proposed. Based on these architectures, research institutions have developed controllers and network node devices that can be used in SDN networks.

At the same time, for controllers and network devices, as well as controllers and upper-layer applications, there are also available interfaces. At present, the SDN controllers based on Open mainly include NOX, POX, Floodlight, Opendaylight, ONOS and so on. Regarding the switching equipment that can be used in the SDN architecture, NEC first introduced a number of SDN switches that support the Open protocol. Soon after that, companies such as Junifer, Hewlett-Packard, and Huawei also launched network devices such as routers, switches, and wireless network Access Points (AP) that support the Openflow protocol developed by their companies.

2.2 Development Trend of Network Architecture

The core of the SDN architecture is the controller. SDN can be considered as a set of client-server relationships between the SDN controller and other entities (administrators, applications, and SDN controllers, etc.). In its role as a server, the SDN controller can provide services to any number of clients, and the SDN controller acting as a client can call services from any number of servers.

Figure 2 illustrates some of its key functions and interfaces. The introduction of some nouns in the SDN architecture diagram shown in Fig. 2 is as follows:

- Client: Client is an entity that receives services from the server.
- Client context: The conceptual component of the server, which represents all the information of a given client and is responsible for participating the management and control operations with active server clients.
- Server: Server is the entity that provides services to the client.
- Server context: The conceptual component of the client, which represents all information for a given server and is responsible for participating the management and control operations with active server clients.
- Resource group: Network resources are grouped into layers according to a certain technology or management boundary. Different network resources and controllers can be controlled and managed with different protocols.

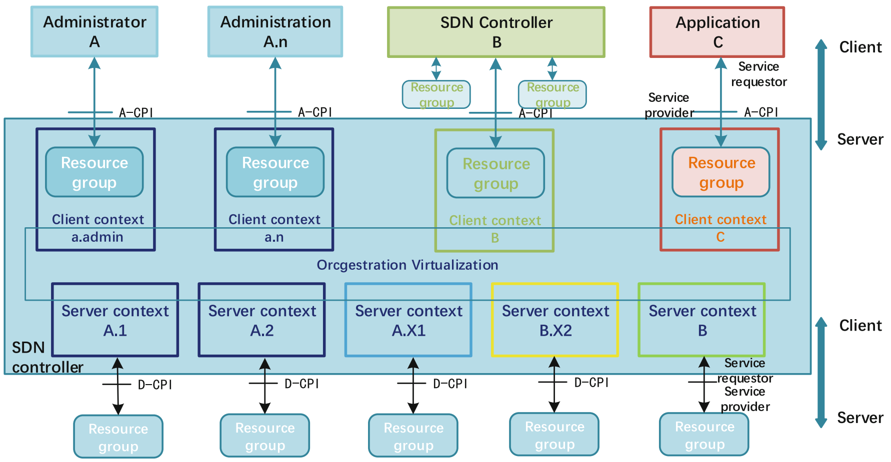

Fig. 2. The core of SDN architecture.

The core work of SDN controller is Orchestration and Virtualization. The control device satisfies client requests through virtualization and collaboration of its underlying resources. As the network environment and customer changes, the SDN controller is responsible for continuously updating the network and business status to achieve optimal policy-based configuration.

An SDN controller controls and manages network resources through a southbound interface (D-CPI, Data-Controller, Platform, and Interface), and through a northbound interface (A-CPI, Application-Control-Control-Control) Internet) to communicate with applications or higher-level controllers. Management and control are regarded as continuous behaviors, in which the role of administrator (Administrator) is different from that of general applications, and it has greater scope and privileges. The administrator has the right to configure the SDN controller itself and Client context and Server context. At present, ONF has defined the standard OpenFlow specification for D-CPI, but A-CPI still lacks industry-recognized standards. The key of SDN is to realize the programmable control of the network, and A-CPI is the basis to realize the programmable. Through A-CPI, network service developers can call various network resources in the form of software programming. At the same time, the upper-layer network resource management system can globally control the resource status of the entire SDN network through A-CPI and uniformly schedule it.

2.3 Routing Algorithms

Routing is the process of selecting a delivery path in a network and has been applied in many types of networks including circuit switched networks and computer networks.

In packet switched networks, routing is a higher-level decision-making which uses a specific packet forwarding mechanism to transfer network packets from their source address to their destination address through intermediate network nodes. Packet forwarding refers to the transmission process of logically processed network packets from one network interface to another. The intermediate nodes are usually routers, switches and other network hardware devices. Routing process usually guides forwarding on the basis of routing table, which maintains records of routes to different network destinations. Therefore, it is very important to build routing table in router memory for efficient routing. Most routing algorithms use only one network path at a time.

In traditional IP networks, routing algorithms are mostly implemented step by step. There are widely recognized and used routing protocols based on link state or distance vector algorithm, such as rip (routing information protocol), IGRP (interior gateway route) protocol, OSPF (open shortest path first), etc. These IP based routing protocols calculate the route and forward the packets only according to the destination address of the packets. The result is that all packets with the same destination address will go through the same network path. Some existing multi-path protocols also do not consider the QoS requirements of different traffic flows. From the perspective of path optimization, this has limitations, because it does not consider the traffic status of the whole network.

Each switching device can obtain manually configured routing table through static routing, or obtain forwarding information through dynamic routing and regularly communicate with other devices in the network to obtain network topology changes, so as to update its own routing table. According to the routing table, the switching equipment forming the network forms a forwarding table. When the IP message arrives at the router interface, the chip in the entrance direction extracts the IP message header, obtains the destination address, and then searches the forwarding table according to the destination IP: if the destination address is the same as the network address connected to an interface of the router, the data will be forwarded directly to the corresponding interface; if the destination address is not found to belong to its own direct network segment, then the router It will check its own routing table, find the corresponding interface of the destination network, and forward it from the corresponding interface; if it is found that the network address recorded in the routing table does not match the address of 0 in the message, it will forward to the default interface according to the router configuration, and return the control message protocol (ICMP, Internet Control Message Protocol) with the unreachable destination address to the user without the default interface configured.

Path selection is to select (or predict) the best route through some route metric parameters among multiple routes. In IP network, routing algorithm is usually used to calculate the metric. This metric information can be divided into bandwidth, network delay, hops, path cost, load, MTU (Maximum Transmission Unit), reliability and communication cost.

In the SDN network, considering that the data plane is separated from the control plane, the switching equipment on the SDN data plane is only responsible for data forwarding according to the flow table issued by the controller, and the forwarding path is calculated by the routing module in the controller. At the same time, SDN is a programmable network, so a corresponding routing algorithm can be developed and loaded on the control plane according to the user's needs.

Based on the above analysis, it can be concluded that the SDN network replaces the routing protocol in the traditional IP network through the centralized control of the controller. The routing strategy of SDN is easier to implement and more flexible.

2.4 Reinforcement Learning

The Basic Principles of Reinforcement Learning. Reinforcement learning is a kind of machine learning and it is different from supervised learning and unsupervised learning. Reinforcement learning is the third machine learning mode in the field of machine learning, which continuously explores the surrounding environment through an agent, and gets a corresponding reward value r for every decision. Reinforcement learning consists of: what to do, how to map between States and actions, and how to maximize reward signals. If the current strategy selected by the agent is correct, a positive feedback signal will be obtained, otherwise a negative feedback signal will be obtained. As a kind of unsupervised learning, the biggest difference between reinforcement learning and supervised learning is that there is no need to label data. Its three important features are: basically a closed-loop form; it does not directly indicate which action to choose; a series of actions and reward signals will have an impact in a long time.

The essence of reinforcement learning is actually a decision-making problem which constantly tries to interact with the environment. Each interaction process will get a different feedback value, and then it will choose the one with the largest feedback value as the best result. Reinforcement learning does not need any label processing in advance. The agent constantly interacts with the environment to get feedback, and then adjusts each attempt strategy through the feedback results. Finally, the whole system will converge, and the intelligent experience will get all the information of the whole environment and make the best choice. Reinforcement learning model mainly includes the following components: system environment (Env), agent (agent), reward function (R), selection strategy (action). As shown in Fig. 3, the agent selects a strategy a at the current time, which is that the agent begins to try the surrounding environment. The agent observes the next state and gets a specific reward value, which may be positive or negative. After a certain number of attempts, the agent can finally make the optimal decision in the current environment.

Reinforcement learning also has its own particularity:

(1) Reinforcement learning is a way of learning without strategy.
(2) Reinforcement learning does not need to know the specific surroundings.
(3) Reinforcement learning, as a weak learning method, constantly tries to get feedback from the surrounding environment through agents. Unlike general

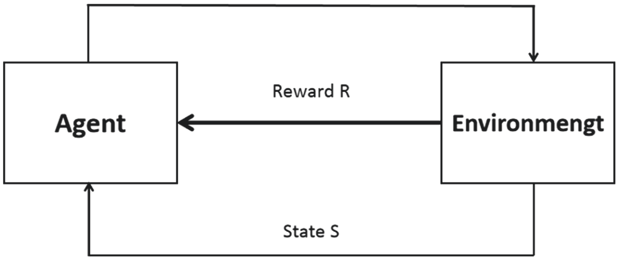

Fig. 3. The interaction of Agent and Environment.

machine learning, it does not even require prior knowledge to train. The feedback information of agent in the process of exploring and learning is presented in the form of reward value, rather than a standard correct answer.
(4) It can be used in combination with other fields such as supervised learning.

Markov Decision Processes. Reinforcement learning is a simple framework from interactive learning to the realization of goals. Learners and decision makers are called agents. What interacts with everything other than agent is called environment and they interact constantly. Agent chooses actions, environment responds to these actions, and communicates new situations to agent. The mathematical framework can be used to summarize reinforcement learning. If the change of state is only related to the previous state and the current action, but not to the previous state and action, then the whole process can be described by Markov decision processes. Most of the research on reinforcement learning methods is based on the theoretical framework of Markov decision-making. The finite state and discrete Markov decision-making process is the basis of the whole reinforcement learning algorithm. When it comes to Markov decision-making process, it is necessary to mention Markov chain and hidden Markov, which are more commonly used in machine learning Hidden Markov model (HMM). In short, once the current state is determined, it is conditionally independent of the historical path of the process, then the process has Markov properties. If a process has Markov properties, then the process is a Markov process. Different from the pure Markov chain, the next state of the Markov decision process system is not only related to the current state, but also to the action to be taken.

A MDP is composed of 4 components $M = (S, A, P, R)$. S represents the state set space, A represents the action set space, P represents the state transition probability matrix, and R represents the expected return value.

The process of simple Markov selection can be represented by Fig. 4. Suppose that the current state of an agent is S_1, and then an action a_1 is selected and executed, and the agent moves to the next state S_2 with probability random P, and then selects an action A, and moves to the next state S_3 with certain probability random, so on.

If action A is selected through state S to obtain MDP state transition process of reward function R, as shown in Fig. 5.

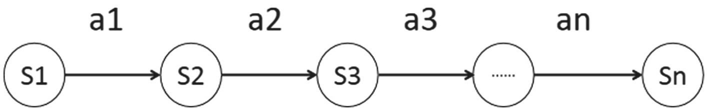

Fig. 4. Simple MDP state transition process.

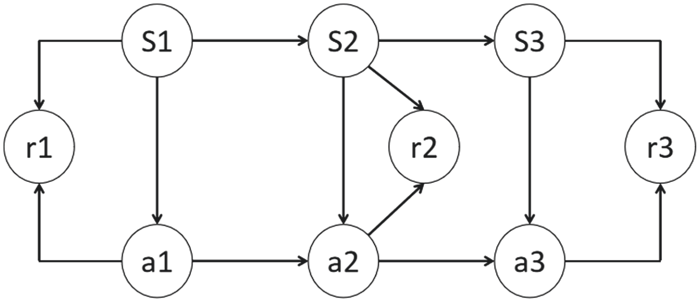

Fig. 5. MDP state transition process with reward function R.

From the MDP process with reward function shown in Fig. 4, it can be seen that the agent will only make decisions based on the reward value under the current decision. In the process of practical application, it is often necessary to consider not only the next step of the current state, but also the next steps of the current state, so the concept of value function is introduced.

Q Learning Algorithm. In real life, the probability of something happening in the environment is usually unknown and Q-learning is just used to estimate an optimal strategy when the system environment is unknown. Q-learning algorithm is a free model Q-value iterative algorithm. The difference is that in sarsa algorithm, the update function of reward value and action value follow different strategies, namely greedy strategy. Q-learning directly uses a Q-valued function that can be calculated iteratively:

$$Q(S_t, a_t) = r_t + \gamma \max_{a_t} \{Q(s(t+1), a_t) | a_t \in A\} \tag{1}$$

The meaning of function is that at time t, the system function is in state s, after action a is taken, the maximum discount reward can be obtained. In Q-Learning algorithm, the update of action value function is different from the policy followed by when selecting action. This update method is also called off-policy.

3 Design and Implementation

3.1 Principle of Routing Planning Algorithm Based on Q-Learning

Q-learning relies on Q-value iteration to achieve the balance between exploration and utilization of strategy selection. In the process of Q-learning algorithm iteration, the direction of Q-value maximum is generally selected. But not every step of iteration will follow this principle. According to the results of many experiments, Q-Learning will distribute with a certain probability, and it will try to go on in the direction of the biggest Q-value. The aim of Q-Learning is to achieve the maximum reward value through continuous learning i.e. Q-learning iteration.

3.2 The Simulation Design of the Experiment

The routing planning system based on reinforcement learning needs to meet the following requirements:

- It has the function of quickly finding a routing path from the starting switch to the ending switch.
- The selection of routing paths can take into account the QoS requirements of the link. The final selection of routing paths should meet the requirements of the link with large available bandwidth, low delay jitter and low packet loss rate.
- The algorithm has good portability and can be compatible with some of the OpenFlow controllers currently on the market, such as Ryu. In order to solve the routing planning problem in SDN environment, this report designs and implements a routing planning system based on reinforcement learning.

The overall architecture design of the routing planning system based on reinforcement learning is shown in Fig. 6. Mainly divided into the following four modules: link discovery module, link state pre-processing module, reinforcement learning training module and Q matrix delivery module.

The link discovery module mainly uses all the information of the whole global network of the controller in the SDN network. So we can get a topology information of the whole network environment through this module. At the same time,

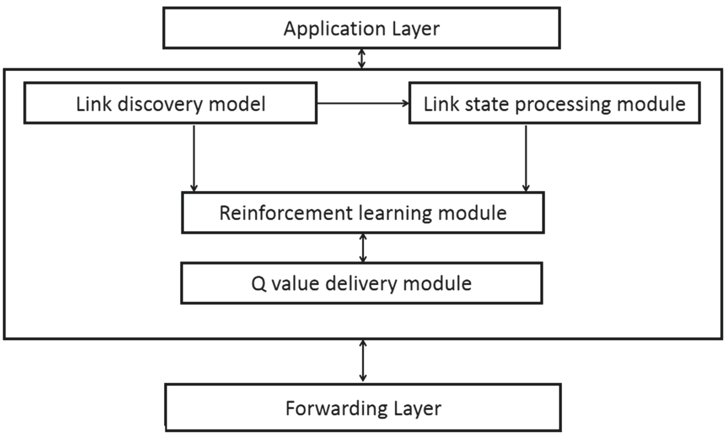

Fig. 6. The architecture of routing plan.

the topology information of the whole link is sent to the reinforcement learning training module.

The link processing module mainly aims at a QoS requirement of the network. The traditional network routing algorithm is basically based on the Dijkstra shortest path algorithm. The selected path only considers the influence of hop number, but ignores other QoS factors such as delay jitter, link available bandwidth and so on. Here, the link state information obtained in advance by the Iperf tool is pre-processed, and then its impact is considered in the reward function of Q learning.

Reinforcement learning training module is the core module of the system. The reinforcement learning algorithm used in this system design is Q-Learning algorithm. Based on the link relationship obtained by the network link discovery module and the link state information processed in advance, the initial Q matrix and the reward function under each link state are established. After training, output a convergent Q value table. The Q-value table distribution module is to distribute the Q-value table obtained after intensive learning training to all OpenFlow switches. SDN controller has two working modes, one is active mode and the other is passive mode. The active mode SDN controller transmits all stream information to the switch at one time; in the passive mode, when the data plane receives a new packet, the control plane will send the relevant information. The first active mode is adopted in this paper. The controller sends the trained Q-value table to all switches. When the switch receives a new packet, as long as you look up the corresponding Q-value table, you can immediately get a route from the start point to the end point.

4 Results and Discussion

This paper introduces the related technologies and principles used in the research process. Firstly, it introduces the basic concept of SDN network topology discovery and the protocol used in SDN network is analyzed. After that, the report introduces the reinforcement learning technology which will be used in the follow-up research. At the same time, most reinforcement learning processes are basically a Markov decision-making process, so this report also introduces several different Markov models and Markov processes in detail to help us understand reinforcement learning algorithm. Then, the current Q-Learning and other reinforcement learning algorithms are described.

In this chapter, we first propose a routing algorithm based on Q-Learning. The implementation principle and algorithm flow of this routing algorithm are introduced in detail. In view of the fact that the route planning algorithm based on Q-Learning needs to design the corresponding features manually, but in the actual application environment, the number of features is often complex, and it is generally unrealistic to extract the required features manually, this paper also puts forward the idea of route planning algorithm based on deep Q-Learning and hopes to use deep-based routing planning algorithm Q-Learning routing algorithm uses neural network model to replace the Q-value table in the conventional Q-Learning algorithm, and learns the design instructions of related functions. Finally, the States and reward functions of reinforcement learning algorithm are described, and a certain network topology is constructed.

5 Conclusion

As a new network architecture, SDN is the focus of the next generation network research, and its biggest feature is the separation of control and forwarding. In SDN network, there is no longer the difference between router and switch. All the work of routing planning is given to the controller, and the switch only undertakes the function of data forwarding. There is no fixed route planning algorithm in the industry at present. The traditional route planning generally adopts Dijkstra shortest path algorithm, which only considers the number of link hops, and does not consider the QoS standard of network links. Therefore, this paper proposes a route planning algorithm based on reinforcement learning, and mainly does the following work around this goal:

(1) The current research status of SDN at home and abroad and the research status of routing planning under SDN network architecture are studied and analyzed. It is concluded that the routing planning in SDN network is actually a very broad research topic, and every aspect of routing aggregation, traffic engineering and so on is worthy of in-depth study. In this paper, only routing algorithm is chosen as the research direction.
(2) In this paper, reinforcement learning is applied to SDN network routing planning. The routing algorithm based on Q-Learning sum is studied. In this process, the network link QoS standard is mainly considered. Obviously, this algorithm can make full use of the network link.
(3) There is a disadvantage in the routing algorithm based on Q-Learning: it needs to design the corresponding characteristics manually. However, in the practical application environment system, the features are often various and complex, so it is very difficult to extract design features by human, and the effect is not very good. Therefore, this paper combines neural network with reinforcement learning and proposes a routing planning algorithm based on deep reinforcement learning. The approximate function trained by neural network is used to replace the Q value table. The experimental results show that the routing algorithm based on deep Q-Learning also has good performance in routing.

References

1. Benzekki, K., Fergougui, A.E., Elalaoui, A.E.: Software-Defined Networking (SDN): a survey. Secur. Commun. Netw. **9**(18), 5803–5833 (2016). https://doi.org/10.1002/sec.1737
2. Bosshart, P., et al.: Forwarding metamorphosis: fast programmable match-action processing in hardware for SDN. In: Chiu, D.M., Wang, J., Barford, P., Seshan, S. (eds.) ACM SIGCOMM 2013 Conference, SIGCOMM 2013, Hong Kong, China, 12–16 August 2013, pp. 99–110. ACM (2013). https://doi.org/10.1145/2486001.2486011

3. Chica, J.C.C., Imbachi, J.C., Botero, J.F.: Security in SDN: a comprehensive survey. J. Netw. Comput. Appl. **159**, 102595 (2020). https://doi.org/10.1016/j.jnca.2020.102595
4. Fu, Q., Sun, E., Kang, M., Li, M., Zhang, Y.: Deep q-learning for routing schemes in SDN-based data center networks. IEEE Access **8**, 103491–103499 (2020). https://doi.org/10.1109/ACCESS.2020.2995511
5. Guo, X., Lin, H., Li, Z., Peng, M.: Deep-reinforcement-learning-based QoS-aware secure routing for SDN-IOT. IEEE Internet Things J. **7**(7), 6242–6251 (2020). https://doi.org/10.1109/JIOT.2019.2960033
6. Henni, D., Ghomari, A., Aoul, Y.H.: A consistent QoS routing strategy for video streaming services in SDN networks. Int. J. Commun. Syst. **33**(10) (2020). https://doi.org/10.1002/dac.4177
7. Liu, Z., Zhu, J., Zhang, J., Liu, Q.: Routing algorithm design of satellite network architecture based on SDN and ICN. Int. J. Satellite Commun. Netw. **38**(1), 1–15 (2020). https://doi.org/10.1002/sat.1304
8. Lu, Y.-H., Leu, F.-Y.: Dynamic routing and bandwidth provision based on reinforcement learning in SDN networks. In: Barolli, L., Amato, F., Moscato, F., Enokido, T., Takizawa, M. (eds.) AINA 2020. AISC, vol. 1151, pp. 1–11. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-44041-1_1
9. Mao, B., et al.: Routing or computing? the paradigm shift towards intelligent computer network packet transmission based on deep learning. IEEE Trans. Comput. **66**(11), 1946–1960 (2017). https://doi.org/10.1109/TC.2017.2709742
10. Tang, F., et al.: On removing routing protocol from future wireless networks: a real-time deep learning approach for intelligent traffic control. IEEE Wirel. Commun. **25**(1), 154–160 (2018). https://doi.org/10.1109/MWC.2017.1700244