Intelligent Network Traffic Control Based on Deep Reinforcement Learning

Fei Wu*, Ting Li, Fucai Luo, Shulin Wu, Chuanqi Xiao, State Grid Fujian Electric Power Company, Fuzhou 350000, China

*Corresponding author: wufeiliting@163.com

Received: June 4, 2021. Revised: December 17, 2021. Accepted: January 13, 2022. Published: January 14, 2022.

Abstract—This paper studies the problems of load balancing and flow control in data center network, and analyzes several common flow control schemes in data center intelligent network and their existing problems. On this basis, the network traffic control problem is modeled with the goal of deep reinforcement learning strategy optimization, and an intelligent network traffic control method based on deep reinforcement learning is proposed. At the same time, for the flow control order problem in deep reinforcement learning algorithm, a flow scheduling priority algorithm is proposed innovatively. According to the decision output, the corresponding flow control and control are carried out, so as to realize the load balance of the network. Finally, experiments show, the network traffic bandwidth loss rate of the proposed intelligent network traffic control method is low. Under the condition of random 60 traffic density, the average bisection bandwidth obtained by the proposed intelligent network traffic control method is 4.0mbps and the control error rate is 2.25%. The intelligent network traffic control method based on deep reinforcement learning has high practicability in the practical application process, and fully meets the research requirements.

Keywords—deep reinforcement learning; flow control; network monitoring.

I INTRODUCTION

With the development of mobile Internet, Internet of things, big data and cloud computing, the number of network users and network scale are growing rapidly. At the same time, the network business is becoming increasingly diversified, and the network structure is becoming more and more complex [1]. In the face of the explosive growth of network traffic, how to realize the load balance of network traffic through reasonable control and scheduling methods, avoid the occurrence of network congestion, so as to ensure the quality of network service and improve the network throughput has become an

increasingly important research topic in the field of computer network [2].

Scholars in related fields have carried out research on this. Reference [3] proposed a network flow control method based on ECMP, using ECMP to forward most of the traffic, and using SDN to selectively reroute some key traffic to balance the link utilization network. Develop and solve simple LP problems to reroute these selected key flows to balance the link utilization of the network and achieve network flow control. However, this method is only suitable for multi-homed terminal hosts, and there are limitations in network flow control. Reference [4] proposed a flow congestion control algorithm based on Hedera and data center TCP. A large number of experiments were carried out on the fat tree data center network, and the probability distribution of its throughput was comprehensively studied. The method will suffer from high load conditions. Influence. However, this method only solves the congestion phenomenon, and there is no solution for places with too little traffic, and there are limitations. Reference [5] proposed a machine learning prediction method to strengthen congestion control in the 5G Internet of Things environment. Using the 5G Internet of Things to work in a high-traffic network with multiple nodes/sensors, a machine learning model based on the decision tree algorithm, under the scalable creation of intelligent applications by Mahout, predict the best enhancement of congestion control in 5G Internet of Things wireless sensors. The model is implemented on the training data set to determine the best parameter settings in the 5G environment. It can predict the optimal solution to improve the performance of the congestion control method, but the network traffic bandwidth loss rate of this method is relatively high. Reference [6] proposed а performance-aware energy-saving method for data center networks. Adopt heuristic algorithms, use switch grouping and link merging to switch traffic to a small number of network devices, Valiant load balancing is used to distribute load on active links and shut down unused switches and links. This method has little impact on the network

performance, but the control error rate of this method is high. Whether in a large number of data center networks arising from cloud computing and Internet, or in the backbone network and LAN communication network constantly updated in the past decade, the traditional flow control and congestion control solutions are becoming more and more inadequate. This is because the traditional solutions are often targeted at some specific network and traffic scenarios. With the continuous emergence of new network structure and traffic mode, the solutions for specific network scenarios will no longer be suitable for the new environment.

To solve the above problems, an intelligent network flow control method based on deep reinforcement learning is proposed. The scheme calculates the global optimal routing strategy through the intelligent network traffic control method based on deep reinforcement learning, provides the corresponding network traffic information for the traffic control decision, and ensures the network traffic control quality control. The proposed intelligent network traffic control method has low network traffic bandwidth loss rate, more average bisection bandwidth and low control error rate.



Figure 1. Schematic diagram of network traffic reinforcement learning

According to Fig. 1, the decision-making body in reinforcement learning is an agent. The agent is in an interactive environment and can perceive the state information of the environment by observing the environment. In different environments, agents may have a variety of actions to choose from [10]. In which state, which action should be performed depends on the strategy adopted by the agent. After the agent makes an action, the environment will give feedback according to the state of the agent and the action of the agent to reflect the quality of the action. According to different feedback, agents can adjust their own strategies. After an agent performs an action, the action may change the state of the agent in the environment, which is called state transition [11]. After the state transition occurs, the agent can continue to make the next round of decisions, execute the next action, and get a series of state action state sequence. The whole process will end when the agent reaches a certain final state or experiences a certain period of time, and the goal of reinforcement learning is to learn a group of optimal behavior strategies, so that the agent can accumulate and maximize the benefits during this period of time [12]. In reinforcement learning, the mathematical model provided by the decision-making process in statistics is usually used to describe the process of reinforcement learning. This is because the essence of reinforcement learning is to solve sequential decision

II OPTIMIZATION OF INTELLIGENT NETWORK TRAFFIC CONTROL METHOD

A. Deep mining method of intelligent network traffic based on deep reinforcement learning

Deep reinforcement learning is a technology combining deep learning method to deal with reinforcement learning problems. The concept and principle of reinforcement learning are introduced, and some typical reinforcement learning algorithms are given [7]. Then the concept of deep learning and deep neural network are briefly introduced [8]. Finally, the deep reinforcement learning and the deep reinforcement learning algorithm used in the subsequent algorithm design are introduced in detail [9]. This paper introduces some basic concepts in reinforcement learning. The essence of reinforcement learning is the process of exploring in the environment and obtaining a behavior strategy that can maximize the expected benefits. Network traffic reinforcement learning is as Fig. 1.

optimization problems, and Markov decision process is a typical tool to solve sequential decision problems in operational research and statistics. Because the Markov decision process has the Markov property, it means that in the process of each state transition, the state of the next moment is only determined by the current state of the agent, and does not depend on the past or future state.

$$P\left[s_{l+1} \mid s_{l}\right] = P\left[S_{l+1} \mid s_{1}, \dots, s_{l}\right]$$
(1)

The whole elements of reinforcement learning can be described by five tuples (S, A, P, R, Y) of Markov decision process: action, state S of executable action set, state transition probability of agent in environment $P[s_{I+1}|s_I]$, probability of action a transferring to states' in states, reward 0.11discount coefficient fed back by environment after state transition, indicates the difference between current and future rewards. Based on the above five tuples, we can define the cumulative reward sum of agents after multi-step state transition in the environment as:

$$G_{t} = P\left[s_{l+1} \mid s_{l}\right]r_{l+1} + \gamma r_{l+2} + \ldots = P\left[s_{l+1} \mid s_{l}\right]\sum_{k=0}^{\infty} \gamma^{k} r_{l+k+1} (2)$$

In formula (2), the discount coefficient γ determines the importance of short-term reward and long-term reward. When γ is close to 0, the agent pays more attention to short-term reward, while when γ is close to 1, the agent pays more attention to short-term reward. After defining the cumulative reward sum G_t , we can get the definition of state value function $V_{\pi}(s)$ the expectation of the cumulative reward sum obtained by an agent starting from a certain state s and taking actions according to strategy is $V_{\pi}(s)$, that is:

$$V_{\pi}(s) = E_{\pi} \left[G_{t} | s_{i} = s \right] = E_{\pi} \left[\sum_{k=0}^{\infty} \gamma^{k} r_{t+k+1} | s_{t} = s \right]$$
(3)

Similarly, the state action value function $Q_x(s,a)$ can be defined. Starting from a certain states, the expectation of the cumulative reward sum obtained by the agent after taking action a according to the strategy is $Q_x(s,a)$:

$$Q_{x}(s,a) = V_{\pi}(s)E_{\pi}\left[G_{t} \mid s_{j} = s, a_{t} = a\right] = V_{\pi}(s)E_{\pi}\left[\sum_{k=0}^{\infty} \gamma^{k}r_{t+k+1} \mid s_{t} = s, a_{t} = a\right] (4)$$

Furthermore, the definition of the optimal value function V(s) is given: for an agent with optimal strategy π , the state value function V(s) in each states is the optimal value function in that state, that is:

$$V(s) = V_{\pi^*}(s), \forall s \in S$$
(5)

The optimal value function satisfies:

$$V_*(s) = \max_{\pi} V_{\pi}(s), \forall s \in S$$
(6)

Similarly, the optimal state action value function under state S can be obtained:

$$Q_*(s,a) = \max_{\pi} Q_{\pi}(s,a), \forall s \in S, \forall a \in A$$
(7)

There is a corresponding relationship between the optimal value function V(s) and the optimal state action value function $Q_{\pi}(s,a)$.

$$V_*(s) = \max_{a \in A} Q_{\pi}(s, a) \tag{8}$$

When the definition of the optimal value function and the optimal state action value function are obtained, the optimal strategy can be found by solving the optimal value function or the optimal state action value function. The goal of reinforcement learning is to find the optimal strategy that agents can find in the environment. According to the different cognitive degree of agents to the environment, the methods to solve the optimal strategy can be divided into two categories: model-based method and model-free method [13]. Before using reinforcement learning to solve problems, it is necessary to model the environment. The process of environment modeling is to obtain the state set, action set, action state transition relationship and reward function in the environment. If all these have been obtained before solving the problem, the method of solving the optimal strategy is called model-based method when the model is known. If some or all of the state set, action set, and action state transition relationship and reward function are unknown, it is necessary to explore and model the environment while solving the optimal strategy [14]. This method is called model free method. When the model is known, the process of solving the optimal strategy can be regarded as an optimization problem under dynamic programming. In this case, dynamic programming method can be used to solve the optimal strategy. Because the Markov decision process has the Markov property, according to the formula:

$$V_{\pi}(s) = E_{\pi} \begin{bmatrix} G_{t} | s_{t} = s \end{bmatrix}$$

= $E_{\pi} \begin{bmatrix} r_{t+1} + \gamma r_{t+2} + \gamma^{2} r_{t+3} + \dots | s_{t} = s \end{bmatrix}$ (9)
= $E_{\pi} \begin{bmatrix} r_{t+1} + \gamma (r_{t+2} + \gamma r_{t+3} + \dots) | s_{t} = s \end{bmatrix}$

Perform the action state full probability expansion for the expectation in the formula:

$$V_{\pi}(s) = E_{\pi} \left[r_{t+1} + \gamma G_{t+1} | s_{t} = s \right]$$

= $\sum_{u \in A} \pi(s, a) \sum_{s' \in S} P_{s \to s'}^{a} \left(r_{s \to s'}^{a} + \gamma E_{\pi} \left[G_{i} | s_{t} = s' \right] \right)$ (10)
= $E_{\pi} \left[r_{t+1} + \gamma G_{t+1} | s_{t} = s \right]$

Replace the formula with:

$$V_{\pi}(s) = \sum_{a \in A} \pi(s, a) \sum_{s' \in S} P_{s \to s'}^{a} \left(r_{s \to s'}^{a} + \gamma V_{z}(s') \right)$$
(11)

For the state action value function Q (s, a), the same form of equation can also be obtained by similar derivation:

$$Q_{\pi}(s,a) = \sum_{s' \in S} P^a_{s \to s'} \left(r^a_{s \to s'} + \gamma V_x(s') \right)$$
(12)

According to the optimal value function, it can be concluded that:

$$V_{*}(s) = \max_{a \in A} \sum_{s'} P^{a}_{s \to s'} \left(r^{a}_{s \to s'} + \gamma V_{*}\left(s'\right) \right)$$
(13)

$$Q^{\cdot}(s,a) = \sum_{s' \in S} P^{a}_{s \to s'} \left(r^{a}_{s \to s'} + \gamma \max_{a' \in A} Q_{*}\left(s', a'\right) \right) \quad (14)$$

The above formula is called optimal equation, which reveals the relationship between the value function of adjacent states and the state action value function in the network structure state space. This relationship is the basis of using dynamic programming to solve the optimal strategy. When the state set, action set, action state transition relationship and reward function in the environment are unknown, the strategy cannot be evaluated as the model-based method, because the model is unknown and the action state full probability expansion cannot be done as the formula. At this time, we can only observe and record the state transition and the reward value by exploring in the network environment, and solve the optimal control strategy of intelligent network traffic through experience.

B. Intelligent network traffic anomaly detection

The purpose of cognitive network traffic control is to share the following functions for subsequent policy routing: one is to filter abnormal traffic; the other is to control the traffic of all kinds of services according to the service perception and classification results of cognitive nodes, so as to prevent network congestion and ensure the expected value of service quality of all kinds of services. For abnormal traffic, zero bandwidth allocation can be used to discard such packets; for normal traffic, bandwidth allocation is made in proportion to the available bandwidth of the link. The setting of bandwidth allocation depends on the quality of service and service parameters corresponding to different traffic types, and the delay, delay jitter and other performance requirements of different traffic types should be considered [15]. Traffic monitoring is the process of collecting data, collecting data and analyzing data from network equipment [16]. The module collects some specific index parameters from the network and feeds them back to the network maintenance personnel. These data can be used to analyze the network performance, understand the network dynamic, predict and diagnose the possible problems. In order to ensure that the estimated value function can converge to the optimal value function, it is necessary to traverse every state in the state space [17]. Therefore, the strategy used in Monte Carlo sampling needs to be exploratory for different states. A common exploratory strategy is. The grey strategy will select the action with the largest state action value function in all actions with a certain probability, and the other probabilities will select the action randomly:

$$\pi(s) = \begin{cases} \arg\max_{a \in A} Q(s, a), & p = 1 - \varepsilon \\ \operatorname{random}, & p = \varepsilon \end{cases}$$
(15)

Each sampling and updating of the method requires a complete process from the start state to the end state. When the process experiences too many state transitions, the efficiency of the Monte Carlo method will be reduced [18]. The time series difference method does not need to update after reaching the end state, but updates the estimated value of the state action value function after each action. The updating method of time series difference method makes use of the idea of dynamic programming, and uses the state action value function of subsequent states to approximately estimate the current state action value function.

$$Q_{\pi}(s,a) = \sum_{s' \in S} P_{s \to s'}^{a} \left(r_{s \to s'}^{a} + \gamma V_{\pi}(s') \right)$$
$$= \sum_{s' \in S} P_{s \to s'}^{a} \left(r_{s \to s'}^{a} + \gamma \sum_{a' \in A} \pi(s',a') Q_{\pi}(s',a') \right)$$
(16)

The updating formula of state action value function can be obtained by incremental summation.

DQN is a deep reinforcement learning algorithm based on value function. Construct a deep reinforcement learning model based on the DQN algorithm. In reinforcement learning, besides solving the optimal strategy based on value iteration, it can also be solved based on strategy iteration. Similarly, in deep reinforcement learning, the model can be optimized based on strategy, which is called strategy gradient. The basic idea of strategy gradient is to continuously calculate the expected cumulative reward of the current strategy and the gradient of the model strategy



Figure 2. Schematic diagram of DQN deep neural network

It should be emphasized that the number of layers in the full connection layer and the number of neurons in the full connection layer are not fixed, but need to be determined according to the size of the network. It cannot give the exact relationship between the network size and the number of full connection layers and neurons. However, according to a large number of experimental results, the neural network structure with two full connection layers and 120 neurons in each full connection layer performs better in the small-scale network with less than 15 hosts and less than 20 switches. For the larger network, it needs to be adaptive when increasing the number of the whole connective layer

parameters, and then update the model strategy parameters based on the gradient until it converges to the optimal strategy [19]. Compared with the deep reinforcement learning algorithm based on value function and its improved model, the deep reinforcement learning algorithm based on strategy gradient has a wider range of application. In traditional networks, network control and data forwarding architecture are usually closely coupled, which leads to the complexity of network control. With the continuous development of network technology, the demand for network flexibility and scalability is also higher and higher. SDN is different from the traditional network architecture [20]. It separates the network into control plane and data plane, which can realize the centralized management of the network. At the same time, it can realize more efficient and fine-grained network control by programming, which makes the network more flexible and intelligent [21]. Traffic control technology is an important part of traffic engineering. Traffic engineering refers to a series of related technologies and methods to optimize the network performance, including network measurement, traffic control, load balancing, congestion control, fault recovery and so on. The core of the traffic control algorithm based on deep reinforcement learning is the deep reinforcement learning algorithm in deep reinforcement learning [22]. This paper gives a detailed introduction of the deep reinforcement learning algorithm. The following will specifically explain how to use the deep reinforcement learning algorithm to deal with the optimization problem of network traffic control. The DQN deep neural network is as Fig. 2.

or neurons [23]. Traffic control can control and manage the traffic in the network in a centralized or distributed way to realize the load balance of the network, so as to improve the bandwidth utilization and throughput of the network and reduce the network delay. In the network of deep reinforcement learning architecture, the SDN controller with centralized network control function is introduced, and the global network view can be obtained from the data plane, which makes the network traffic control based on deep reinforcement learning more effective and flexible [24]. Efficient and reasonable network flow control scheme can improve the network transmission performance, avoid network congestion, and provide bandwidth support for the upper application layer, so as to meet the user needs of the application layer. In the traditional IP network, the routing method based on the shortest path algorithm is often used to distribute the traffic to each physical link. According to the static link weight, the optimal path of data forwarding is calculated between switches in the network. However, although the routing strategy based on the shortest path algorithm can give the optimal path of a single traffic, it cannot avoid the congestion problem caused by multiple traffic concentrated on some links. The overall structure of deep learning is as Fig. 3.



Figure 3. Overall structure of deep learning

According to Fig. 3, the overall architecture of deep learning consists of five parts, from top to bottom: application plane, north interface, control plane, South interface and data plane. The application plane consists of a number of deep reinforcement learning business applications, which are developed by network managers and can call one or more northbound interface APIs provided by the control plane. Network managers can use the API of these northbound interfaces to obtain control plane information, schedule network resources, or flexibly implement business functions in other programmable ways. Through the application plane of deep reinforcement learning architecture, network managers can make full use of the openness of SDN to realize various intelligent services [25]. The interface between the application plane and the control plane, which provides the corresponding interface API for the application plane, enables the deep reinforcement learning business application to issue instructions to the control plane or obtain relevant network information. Different from the southbound interface, there are many kinds of SDN controllers, so there is no unified northbound interface standard. Control plane: control plane is the center of deep reinforcement learning architecture, which is SDN controller and can realize the control function of traditional network devices. The control plane can be regarded as a centralized abstraction of control functions of distributed network devices in traditional networks. After separating the control plane from the data plane, the control plane can only focus on network strategy planning and flow control, while the data plane is responsible for data forwarding. On the one hand, the SDN controller of the control plane is responsible for providing the north interface for the business applications in the upper application plane, on the other hand, it is responsible for transmitting the network control information of the application layer to the data plane through the South interface [26]. The interface between the control plane and the data plane provides a transmission medium for the information interaction between the SDN controller and the underlying forwarding device. With the help of the southbound interface, the SDN controller can collect the topology information and traffic information of the

underlying forwarding device in the network, and can also send the network control information such as the flow table to the underlying forwarding device, so as to control the packet forwarding behavior. The current southbound interface protocols include OpenFlow protocol, NETCONF protocol and PCEP, among which OpenFlow protocol is the mainstream southbound protocol. The data plane is mainly composed of the underlying physical forwarding device, and the forwarding operation of the forwarding device depends on the flow label issued by the SDN controller. In order to use the southbound interface, these forwarding devices need to support the southbound interface protocol for communication with the upper layer. The data plane is the executor of the upper layer network policy. It does not have the decision-making ability. It needs to accept the upper layer application policy and implement the corresponding configuration in the data plane to realize the user data forwarding.

C. Realization of intelligent network flow control

For the problem of network load balancing, if we want to find a feasible load balancing scheme, the first thing to do is to determine the network scenario of the load balancing scheme. For different uses of the network, it often has a different network architecture and topology, and the network routing and forwarding protocols deployed and running on it are also designed for the network architecture. Therefore, for a specific network, we need to design specific load balancing and congestion control mechanisms according to its special network architecture, topology and communication protocol. In this paper, the load balancing problem in data center network is discussed. Compared with campus network, intranet and other common networks, the network structure of data center network is significantly different [27]. The communication goals and traffic patterns of campus network and intranet are usually regular and predictable. The path between the source node and the destination node in the network structure is relatively single, which makes the network manager only set a small number of redundant paths for fault tolerance when designing the network structure. In the data center network, in order to realize the flattening of the network and ensure the equal status of each host in the network, multiple paths are often provided between hosts [28]. Before the implementation of the algorithm, we first need to determine the basic information in the network: the link information between switches in the network, the information of all possible flows in the network, the bandwidth requirement range of all flows, and the optional forwarding path of each flow in the network. Based on the above information, we use the proposed flow scheduling priority algorithm to calculate the order of selecting paths for flows. When the algorithm is running, the basic information in the network must be consistent with the network information in the deep reinforcement learning model training. The flow chart of intelligent network traffic control method based on deep reinforcement learning is as Fig. 4.



Figure 4. Flow chart of intelligent network traffic control method based on deep reinforcement learning

According to Fig. 4, the algorithm will be executed under two conditions to schedule the global traffic. The first case is to detect a new elephant flow in the network, and the second case is to execute periodically according to the timer. The purpose of periodic execution algorithm is to reserve new available bandwidth for the link when some streams in the network are finished. At this time, the execution algorithm can reuse the available bandwidth to increase the network throughput. In the deep reinforcement learning architecture, the global traffic demand information can be provided by the SDN controller. After inputting the global traffic demand information, the algorithm will preprocess the global traffic demand information, and the preprocessed state information is used as the initial state input of deep neural network. The deep neural network outputs the action Q value to determine the forwarding path selection of the first flow, and gets the next state at the same time, and then repeats the process of state input and action output until the forwarding paths of all flows in the network are obtained, that is, the new traffic routing strategy. The intelligent network traffic control scheme based on deep reinforcement learning architecture is as Fig. 5.



Figure 5. Intelligent network flow control scheme based on deep reinforcement learning architecture

It can be seen from the system design scheme in the

figure that the routing decision-making module is highly decoupled from the whole SDN architecture network, and the intelligent control of the network is realized only through simple data interaction with the SDN controller, which makes the flow control algorithm have certain flexibility and can be adjusted and modified based on the actual situation of the network [29]. While calculating the new routing strategy, the algorithm also calculates the total traffic bandwidth loss in the current network based on the current routing strategy. After obtaining the new strategy, the algorithm calculates the total traffic bandwidth loss of the new strategy and calculates the difference between the new strategy and the current strategy, which is the traffic bandwidth gain of the new routing strategy. If the gain is greater than the cost of flow control, the algorithm will output a new routing strategy to update the forwarding path of each flow in the network. Otherwise, it will keep the current routing strategy and end the algorithm waiting for the next execution. An important feature of intelligent network traffic control method is that the output result of the algorithm is not the forwarding path of a certain flow in the network, but the forwarding path of all flows in the network, which means that each output result of the algorithm can be regarded as the rerouting of all flows in the network. Due to the need to use control signaling to change the flow path, this global rerouting will bring some overhead to the network, and may also lead to the increase of flow delay and packet disorder. Therefore, after obtaining the output of the algorithm, we should determine whether the bandwidth gain obtained by using the new routing strategy is greater than the cost of rerouting. The cost of rerouting is related to the difference between the new and old routing strategies, that is, the number of forwarding paths changed by the new routing strategy compared with the current routing strategy in the network. When the number of flows that need to change the forwarding path is very large, the cost of rerouting will also be very large. Some quantitative methods can be used to calculate the cost of rerouting, which can be compared with the bandwidth gain of the new routing strategy. The specific calculation method of the cost is related to the network type, scale, transmission protocol, routing method, control signaling overhead, etc., so it is impossible to give a unified solution in this paper In practical application, the calculation method is set according to the specific situation of the network, so as to ensure the effect of network flow control.

III ANALYSIS OF EXPERIMENTAL RESULTS

In order to evaluate the proposed intelligent network traffic control method, a network environment simulation model based on flow level is written in Python. The simulation model can simulate the generation of network traffic and the load generated on the link in the form of numerical simulation. In the experiment, NS2 network simulation software is used to simulate and analyze different algorithms. The network topology used in the experiment is standard dumbbell structure. The figure shows the throughput comparison of the DQN based traffic control algorithm and the other four algorithms under different traffic densities: andom20 and random40, and the comparison of the network traffic bandwidth loss rate of the five algorithms under different traffic densities. The comparison results of throughput and network traffic bandwidth loss rate of the five algorithms under different traffic densities are as Figs. 6-9.



Figure 6. Throughput comparison of five control methods under random 20 traffic density



Figure 7 Throughput comparison of five control methods under random 40 traffic density



Figure 8. Comparison of network traffic bandwidth loss rate of five

control methods under random 20 traffic density



Figure 9. Comparison of network traffic bandwidth loss rate of five control methods under random 40 traffic density

According to Figs. 6-9, the x-axis in the above figures is the average traffic bandwidth demand in the network. With the increase of average traffic bandwidth demand, the changes of throughput and network traffic bandwidth loss rate are recorded. The demand of traffic bandwidth in the network will gradually increase from 0.1 Gbps to 5.5 Gbps, which means that the demand range of traffic bandwidth generated in the network is from 0.5 Gbps. With the increase of the average bandwidth requirement of traffic in the network, the throughput of all algorithms in the network increases, but finally tends to be stable. This is because the upper limit of link bandwidth capacity in the network determines the upper limit of the transmission capacity of the network itself, that is, the maximum throughput of the network. Although the throughput of the network will gradually increase with the increase of traffic bandwidth demand, it will not exceed the maximum throughput of the network. The traffic control algorithm used in the network is also an important factor affecting the maximum throughput of the network. Under three different traffic densities, the throughput obtained by using DQN based traffic control algorithm and mahout algorithm is significantly higher than that obtained by ECMP, VLB and Hedera. The throughput of DQN based traffic control algorithm is close to mahout when the network traffic density is random20, slightly higher than mahout when the network traffic density is random40, and slightly higher than mahout when the network traffic density is random40. The reason for this phenomenon is that when the traffic density in the network is low, the throughput gain obtained by reasonably scheduling the traffic is also small. According to the offline increasing first fit algorithm in mahout, the traffic forwarding path selection is approximate to the optimal global network routing strategy. At this time, the traffic control algorithm based on DON can get better performance than ECMP, VLB and Hedera have better routing strategies, but they can only get approximate results compared with mahout. When the traffic density is high, it is more likely to happen in the network. For example, the scheduling algorithm based on DQN can get more gain, so the throughput of the algorithm is higher than mahout and the other three algorithms. The INTERNATIONAL JOURNAL OF CIRCUITS, SYSTEMS AND SIGNAL PROCESSING DOI: 10.46300/9106.2022.16.73



Figure 10. Average bisection bandwidth comparison of algorithms under three traffic modes

Because the simulation network of fat tree structure used in the experiment is small, the effective transmission bandwidth between two pods in the network is taken as the index of average bisection bandwidth. It can be seen that the average bisection bandwidth obtained by DQN based intelligent network traffic control method under random 20 traffic density is similar to mahout, which is higher than other algorithms, while the average bisection bandwidth obtained by DQN based intelligent network traffic control method under random 40 traffic density is similar to mahout the average bisection bandwidth obtained by the proposed intelligent network traffic control method is 3.2% and 4.8% higher than that obtained by mahout, which performs best among other algorithms. From the above experimental results, it can be seen that in the typical data center network with fat tree structure, the intelligent network traffic control method based on deep reinforcement learning can perform the same performance as offline increasing first in mahout under the condition of low traffic density The performance of fit algorithm is close to that of other four traffic control algorithms, and it achieves better performance in high traffic density network environment. This means that the proposed intelligent network traffic control method based on deep reinforcement learning is more suitable for those data center networks with high traffic density and heavy network load. Therefore, a more intelligent network traffic control scheme is needed to optimize the network routing strategy based on the global traffic bandwidth demand. Experiments show that the designed intelligent network traffic control method is more effective the control scheme can realize this requirement.

In order to prove that the method in this paper can accurately control the flow, the method in this paper is compared with the other four methods. When comparing the five methods to control the flow output, the comparison result of the flow control error rate is shown in Fig. 11.



According to the analysis results of Fig. 11, as the random flow density increases, the control error rates of the five methods all become larger. In the case of random 20 flow density, random 40 flow density, and random 60 flow density, the intelligent network based on DQN the control error rates of the flow control methods are 1.6%, 1.9%, and 2.25% respectively, and the control error rates are all lower than the other four methods, indicating that the method in this paper can better control the network flow, because the method in this paper uses the deep learning reinforcement method. The status information of the environment can be perceived by observing the environment, thereby reducing the control error rate.

IV CONCLUSION

This paper studies the problem of load balancing and flow control in data center network, and innovatively proposes an intelligent network flow control scheme based on deep reinforcement learning. The experimental results show that compared with the other four methods, this scheme can effectively optimize the network routing strategy and realize the network load balancing, so as to improve the overall transmission performance of the network. The network traffic bandwidth loss rate of the proposed intelligent network traffic control method is low. Under random 40 traffic density or random 60 traffic density, the average bisection bandwidth obtained by the proposed intelligent network traffic control method is 3.65mbps and 4.0mbps respectively. Under random 20 traffic density, random 40 traffic density and random 60 traffic density, the control error rates of intelligent network flow control method based on dqn are 1.6%, 1.9% and 2.25% respectively. The proposed intelligent network traffic control scheme based on deep reinforcement learning is based on deep reinforcement learning structure. The SDN controller provides the required global network traffic information for the algorithm. At the same time, the global network routing strategy output by the DQN based traffic control algorithm will be distributed to the network data plane in the form of flow table through the SDN controller, so as to give full play to the deep reinforcement learning the flexibility of the network traffic control method is improved. The environment built in this paper cannot be measured in the real city-wide area, and the application and implementation of real cities need to be further studied. The research on intelligent network flow control method provides a reliable basis for the optimization, integration, planning, construction and operation and maintenance of intelligent network flow control, and reduces the repeated investment in intelligent network flow control. The research results can promote the steady and rapid development of data communication network, and have good backward compatibility.

REFERENCES

- Manchella K, Umrawal A K, Aggarwal V. FlexPool: A distributed model-free deep reinforcement learning algorithm for joint passengers and goods transportation. IEEE Transactions on Intelligent Transportation Systems, 2021, 22(4):2035-2047.
- [2] Wu W, Huang Z, Zeng J, et al. A fast decision-making method for process planning with dynamic machining resources via deep reinforcement learning. Journal of Manufacturing Systems, 2021, 58(10):392-411.
- [3] Zhang J, Ye M, Guo Z, et al. CFR-RL: Traffic engineering with reinforcement learning in SDN. IEEE Journal on Selected Areas in Communications, 2020, 38(10):2249-2259.
- [4] Alawadi A H, Zaher M, S Molnár. Methods for predicting behavior of elephant flows in data center networks. Infocommunications Journal, 2019, 15(11):arXiv:1911.11673.
- [5] Najm I A, Hamoud A K, Lloret J, et al. Machine learning prediction approach to enhance congestion control in 5G IoT environment. Electronics, 2019, 8(6):607-612.
- [6] Al-Tarazi M, Chang J M. Performance-aware energy saving for data center networks. IEEE Transactions on Network and Service Management, 2019, 16(1):206-219.
- [7] Lei Y, Li W. Interactive recommendation with user-specific deep reinforcement learning. ACM Transactions on Knowledge Discovery from Data, 2019, 13(6):1-15.
- [8] Gochhait S, Rimal Y, Pageni S. The comparison of forward and backward neural network model – A study on the prediction of student grade. WSEAS Transactions on Systems and Control, 2021, 16: 422-429.
- [9] Xu H, Zhang W, Deng J, et al. Active flow control with rotating cylinders by an artificial neural network trained by deep reinforcement learning. Journal of Hydrodynamics, 2020, 32(2):254-258.
- [10] Chen J, Xue Z X, Fan D Q. Deep reinforcement learning based left-turn connected and automated vehicle control at signalized intersection in vehicle-to-infrastructure environment. Information (Switzerland), 2020, 11(2):77.
- [11] He B, Wang J, Qi Q, et al. Towards intelligent provisioning of virtualized network functions in cloud of things: A deep reinforcement learning based approach. IEEE Transactions on Cloud Computing, 2020, 1-1.
- [12] Al-Najjar A, Khan F H, Portmann M. Network traffic control for multi-homed end-hosts via SDN. IET

Communications, 2020, 14(19):3312-3323.

- [13] Wu Q, Shen J, Yong B, et al. Smart fog based workflow for traffic control networks. Future Generation Computer Systems, 2019, 97:825-835.
- [14] Baldi S, Michailidis I, Ntampasi V, et al. A simulation-based traffic signal control for congested urban traffic networks. Transportation Science, 2019, 53(1):6-20.
- [15] Luo J, Huang Y S, Weng Y S. Design of variable traffic light control systems for preventing two-way grid network traffic jams using timed petri nets. IEEE Transactions on Intelligent Transportation Systems, 2019, 21(07):3117-3127.
- [16] Smith Q J, Valverde R. A perceptron based neural network data analytics architecture for the detection of fraud in credit card transactions in financial legacy systems. WSEAS Transactions on Systems and Control, 2021, 16:358-374.
- [17] Younes M B. Secure traffic efficiency control protocol for downtown vehicular networks. International Journal of Network Security, 2019, 21(3):511-521.
- [18] Xu H, Zhang W, Deng J, et al. Active flow control with rotating cylinders by an artificial neural network trained by deep reinforcement learning. Journal of Hydrodynamics, 2020, 32(2):254-258.
- [19] Lu J, Feng L, Yang J, et al. Artificial agent: The fusion of artificial intelligence and a mobile agent for energy-efficient traffic control in wireless sensor networks. Future Generation Computer Systems, 2019, 95:45-51.
- [20] Hao S X, Yang L C. Traffic network modeling and extended max-pressure traffic control strategy based on granular computing theory. Mathematical Problems in Engineering, 2019, 2019(10):1-11.
- [21] Leon J P A, Rico-Novella F J, Llopis L J D L C. Predictive traffic control and differentiation on smart grid neighborhood area networks. IEEE Access, 2020, 8(10):216805-216821.
- [22] Chow A H F, Sha R, Li Y. Adaptive control strategies for urban network traffic via a decentralized approach with user-optimal routing. IEEE Transactions on Intelligent Transportation Systems, 2020, 21(4):1697-1704.
- [23] Sun P, Samaan N. A novel VANET-assisted traffic control for supporting vehicular cloud computing. IEEE Transactions on Intelligent Transportation Systems, 2020, 1-11.
- [24] Dong H, Lin X, Gao S, et al. Neural networks-based sliding mode fault-tolerant control for high-speed trains with bounded parameters and actuator faults. IEEE Transactions on Vehicular Technology, 2020, 69(2):1353-1362.
- [25] Mansour A M. Internal marketing and service quality in jordanian hotels. WSEAS Transactions on Environment and Development, 2020, 16:831-843.
- [26] Weg G S V D, Keyvan-Ekbatani M, Hegyi A, et al. Linear MPC-based urban traffic control using the link transmission model. IEEE Transactions on Intelligent Transportation Systems, 2020, 21(10):4133-4148.
- [27] Bahtiti N, Rayyan A A, Sasa T, Alahmad W. Survey of

Volume 16, 2022

Jordanian awareness about hazardous symbols of chemicals. WSEAS Transactions on Environment and Development, 2021, 17:1030-1038.

- [28] Bahtiti N H., Abdel-Rahaman I. Anti-Corrosive effect of Jordanian-bay- leaves aqueous extract on mild steel in 1.0 M hydrochloric acid solution. WSEAS Transactions on Environment and Development, 2021, 17: 614-618.
- [29] Aljarrah M. The Impact of Enterprise resource planning system of human resources on the employees' performance appraisal in Jordan. WSEAS Transactions on Environment and Development, 2021, 17: 351-359.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0

https://creativecommons.org/licenses/by/4.0/deed.en US