Wireless Networks (2024) 30:4507–4525
https://doi.org/10.1007/s11276-022-03066-x

Intelligent routing method based on Dueling DQN reinforcement learning and network traffic state prediction in SDN

Linqiang Huang[1] · Miao Ye[2,3] · Xingsi Xue[3] · Yong Wang[1] · Hongbing Qiu[2] · Xiaofang Deng[2]

Accepted: 27 June 2022 / Published online: 9 July 2022

Abstract
The traditional routing method makes use of limited information on the network links to make routing decisions, which makes it difficult to adapt to the dynamic and complex network and adjust the router's forward strategy. To address these issues, this paper proposes an intelligent routing method based on the Software Defined Network (SDN), Dueling DQN (a Deep Reinforcement Learning algorithm) and network traffic state prediction. First, the global network awareness information is obtained with the SDN network measurement mechanism, which is converted into a traffic matrix consisting of multiple network link status information such as bandwidth and delay, etc. Then, the optimal forwarding route under the current network state is generated by predicting the network traffic matrix and the Dueling DQN. The experimental results show that: (1) compared with the traditional Dijkstra and OSPF routing methods, the proposed method significantly improves the network throughput and effectively reduces the network delay and packet loss rate; (2) comparing with the reinforcement learning algorithms DDPG and PPO, the proposed approach achieves a faster convergence state, which improves the efficiency of network routing.

Keywords Software defined network · Intelligent routing method · Deep reinforcement learning · Network traffic state prediction

1 Introduction

The network scale has increased in recent years, and various novel network devices have emerged one after another. Due to the characteristics of tightly coupled control with forwarding, and decentralized management, the traditional network architecture is unfavorable for network updating and maintenance and increases the difficulty of providing routing optimization services for network data traffic, making it difficult to meet current network requirements. The emergence of software defined network (SDN) [1, 2] represents the direction for solving the above problems. In this novel SDN architecture, global network information is obtained through the SDN southbound interface, and SDN northbound interface provides application services for upper layers, thereby realizing centralized and unified control of the network, simplifying network management, reducing network operation costs, and facilitating network updating and deployment. Compared with the traditional network architecture, the SDN architecture has the advantages of an open and programmable network, decoupling of the control plane and data plane, and centralized logical control. Therefore, SDN technology is superior to the traditional TCP/IP network architecture in many respects, and can better meet the needs of the current networks.

Routing optimization is always a crucial research optic in data center network (DCN), traffic engineering (TE) and

✉ Miao Ye
yemiao@guet.edu.cn

Linqiang Huang
huanglinqiang_2020@foxmail.com

[1] School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China

[2] Guangxi Key Laboratory of Wireless Wideband Communication and Signal Processing, Guilin University of Electronic Technology, Guilin 541004, China

[3] Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou 350118, Fujian, China

quality of service (QoS). In the development of traditional network architecture, open shortest path first (OSPF) [3], routing information protocol (RIP) [4], Valiant load balancing (VLB) [5] and other traditional routing methods have been successfully applied to many fields in the past decades. However, as the network scale continues to increase, network traffic is presenting exponential growth. In this case, traditional routing methods only use inadequate network links information to make routing decisions, and cannot generate optimal route-forwarding strategies based on the global status of the SDN network. Furthermore, the routing method of traditional network architecture has slow convergence speed, long response time, and poor adaptability, which is more obvious as the SDN network scale is larger, and is no longer applicable to the SDN network architecture. Therefore, proposing an effective solution to optimize the routing strategy in the SDN network under the basic premise of ensuring network service quality is important.

Routing optimization is a crucial technology to improve network performance, reduce network load and strengthen network service quality. Compared to the traditional methods of measuring network link status information, the SDN architecture allows global link status information to be more conveniently obtained for the flexible deployment of routing strategies, and accordingly many researchers have investigated routing optimization based on the SDN architecture. At present, the heuristic routing algorithm is still the main basis of routing optimization problems. However, with the rapid development of artificial intelligence technology, some intelligent algorithms show significant advantages in solving complex and high-dimensional problems, and have achieved good results in high-volume data processing, complex strategy decision-making and other aspects. Many researchers have begun to apply these intelligent algorithms to the routing optimization field of SDN network architecture, and some work and achievements of intelligent routing optimization methods have also been produced. Based on the SDN environment, an SDN intelligent routing method (DRL-TP) based on Dueling DQN deep reinforcement learning (DRL) [6] and network traffic state prediction proposed in this paper, it can obtain the global network topology information and link status information through SDN technology, and utilizes the end-to-end awareness ability of deep learning (DL) [7], the ability of decision making for exploration and utilization in reinforcement learning (RL) [8], and the capability of time-series prediction in recurrent neural network (RNN) [9]. After proper training, the optimal route-forwarding path between all source-destination switches can be obtained in real-time.

The main innovations of this paper are as follows.

1. In contrast to traditional method of measuring network link status information, this paper designs a network measurement mechanism based on SDN technology, which can simultaneously monitor port changes in network link switches and obtain link status information such as bandwidth and delay so as to obtain real-time global link status information of the network.
2. An SDN intelligent routing method based on DRL is designed and implemented. Compared with the existing methods which consider the single network link metric of bandwidth to construct the traffic matrix as the basis of DRL state space, this paper comprehensively considers multiple network link indicators, such as bandwidth, delay, packet loss rate and used bandwidth. The Dueling DQN algorithm can thus rely on a traffic matrix composed of multiple indicators, to guide the agent to better explore the state space in accordance with the designed reward function, and select the optimal route forwarding path from the action space.
3. In the proposed DRL-based SDN intelligent routing method, a gated recurrent unit (GRU) based network traffic state prediction model is designed. This model is more concise and can more easily converge than common long short-term memory (LSTM) prediction models, allowing it to more quickly discover unknown and hidden network traffic states.

The remainder of this paper is organized as follows. Section 2 introduces the current research status and existing problems of mainstream routing methods. Section 3 describes the overall architecture and modeling of the SDN intelligent routing method proposed in this paper. In Section 4, the proposed intelligent routing method proposed in this paper and the specific algorithm design details are introduced. Section 5 introduces the specific experimental environment and tests the effectiveness of the SDN intelligent routing method through relevant experiments. Section 6 gives the final conclusions and proposes the direction for further research.

2 Related work

In the research of routing optimization methods under the SDN network architecture, in addition to traditional static routing methods and dynamic routing methods, the existing mainstream methods can be divided into heuristic methods and intelligent methods.

2.1 Heuristics methods

Ahn et al. [10] proposed a genetic algorithm to solve the shortest path routing problem. Derbel et al. [11] considered

that the genetic algorithm easily falls into local optimal solution, and designed an iterative local search genetic algorithm to enhance the search space to find a better route-forwarding strategy. Zhang et al. [12] studied that dynamic source path protocol (DSR) only based on the minimum number of hops in the network to select the route-forwarding path without considering the network node capacity, they proposed a genetic-bacterial foraging optimization algorithm to select the optimal network route. First, a algorithm was used to quickly find the optimal routing forwarding path in the current network, and then bacterial foraging optimization algorithm was used to optimize the optimal solution to prevent the genetic algorithm from falling into a local optimum, and then find the optimal route-forwarding path in the network. Parsaei et al. [13] took advantage of the numerical control separation of SDN to model QoS as a constrained shortest path linear programming problem. Aiming at the completely NP property of this problem, a solution based on the ant colony algorithm was proposed. Jing et al. [14] proposed a genetic ant colony routing algorithm to adapt the increasing data processing and forwarding requirements of power communication networks, which can obtain multiple candidate optimal solutions through fast search, and the positive feedback method was used to shorten the maximum search time to quickly find the optimal forwarding paths. Lin et al. [15] proposed a simulated annealing based QoS-aware routing (SAQR) algorithm based on an SDN hybrid network architecture, that can dynamically obtain the route-forwarding paths according to the current network status and predefined QoS requirements. Truong et al. [16] proposed a heuristic traffic engineering method based on SDN. First, the path with the lowest cost was selected from K candidate paths, and then the heuristic method was used to evaluate whether the current path with the lowest cost was optimal according to the current network load, thereby realizing the search for the optimal forwarding path in multipath forwarding. Ke et al. [17] designed an artificial bee colony algorithm to dynamically make optimal routing decisions based on the status information of sensor nodes monitored and obtained by SDN controller to improve the utilization rate of sensor nodes and packet route-forwarding problems in wireless sensor networks. Shokouhifar et al. [18] proposed an ant colony optimization algorithm based on fuzzy heuristic to solve the NP-hard problem of virtual functional routing layout in a virtual network, which effectively solved the problem of virtual network functional routing layout. Zhang et al. [19] considered dividing large scale networks into multiple domains to improve the routing efficiency of information centric networking (ICN) by reducing route-forwarding among similar data domains. Therefore, based on the idea of region division, a particle swarm optimization algorithm was proposed to provide an advanced implementation method for intradomain routing operations.

The above methods mainly adopt heuristic methods to iteratively continuously to converge the network state. However, such methods require strict scenarios, and the changes of network topology and link state may cause large fluctuations and errors in heuristic methods, leading to potential scalability issues and affecting network performance.

2.2 Intelligent methods

2.2.1 Machine learning algorithm

Valadarsky et al. [20] used supervised learning to predict future network traffic changes, and designed a supervised learning algorithm to obtain the optimal routing strategy according to the changing demand of network traffic. Sharma et al. [21] designed a decision tree and neural network model, which took into account switch node power consumption, node location and port flow rate and other factors to train the model, and finally realized the transmission of packets in the network to the destination switch node with minimum delay. Li et al. [22] introduced a predesign route algorithm based on multimachine learning, using the clustering algorithm to extract the characteristics of the flow in the network as the input. Then, supervised learning was used to predict the future network traffic demand, so as to achieve the adaptive strategy of multipath routing. Zhou et al. [23] proposed a path planning algorithm based on Naive Bayes. After learning a large amount of training data, established a naive Bayes classifier model by learning a large number of training data, and designed and implemented a path planning algorithm that can intelligently generate the optimal propagation tree under the closed-loop structure. Yanjun et al. [24] presented a meta-layer framework based on supervised learning, Multiple machine learning modules were built in the meta-layer, and heuristic algorithms were used to create a reliable training set as the input of the meta-layer. Each meta-layer obtains the routing strategy of the current meta-layer, and the optimal routing strategy is obtained by converging the meta-layer, which effectively improves the network performance. Tang et al. [25] proposed a network intelligent traffic control method. A deep convolutional neural network was used as the network backbone, utilizing the end-to-end awareness ability of deep learning to reduce the degree of congestion in the wireless backbone network. Mao et al. [26] designed a deep convolutional network algorithm to intelligently calculate the network routing path and realize the adaptive network routing strategy considering that the lack of adaptive capability in the routing strategy of traditional

communication systems and the inability to efficiently use the resources of SDN controller. Kato et al. [27] proposed a supervised deep neural network system, that resolves the difficulty of deep learning in establishing appropriate input and output models under large-scale heterogeneous networks and improves the performance of heterogeneous network flow control.

The above algorithms use machine learning technology to improve the performance of the network. However, machine learning requires a large amount of labelled data for training, which leads to extremely high computational complexity. Machine learning also must establish an underlying network model, but for complex and dynamic networks, it is difficult to design a model that meets all network state requirements.

2.2.2 Reinforcement learning algorithm

Hendriks et al. [28] proposed a Q^2-routing algorithm for routing optimization of AdHoc wireless networks, which realized the trade-off between the communication cost and routing quality of the network. Chen et al. [29] designed a distributed high-efficiency buffer scheme, through the distributed cache protocol and management mechanism based on TCP, the server queries the cache table and the Pache false positive information table to determine the sent data packets, eliminating redundant traffic in the data center network. Casas-Velasco et al. [30] considered the limitations of traditional routing protocols in adapting to changes in network traffic, and proposed a Q-Learning routing algorithm based on SDN, which uses link state information to make routing decisions. Jin et al. [31] designed a routing mechanism based on Q-learning algorithm with an SDN environment to meet the diverse needs of users and effectively reduce the packet loss rate of business data flows; Yin et al. [32] proposed an accelerated Q-learning routing algorithm for existing satellite routing algorithms that ignores the state of the satellite network, which improves the convergence speed of the model and achieves the goal of selecting the optimal route according to the dynamic characteristics of the satellite network.

The above methods store experience and reward in a table and find the optimal routing policy based on a lookup table. However, as the network scale increases, storage space and query time become a major challenge and cannot adapt to dynamic network requirements. To solve this problem, many researchers have begun to combine deep learning and reinforcement learning to optimize routing.

Zhao et al. [33] proposed an intelligent routing algorithm based on DRL, which effectively alleviates network congestion, achieves network load balance and meets the needs of urban intelligent network services. Chen et al. [34] developed a method based on the DDPG framework to solve complex and difficult dynamic network modelling problems. The network is divided into upstream and downstream networks, multiple new features are introduced as state space, and the action space is the intersection of paths between source and destination switches in the upstream and downstream networks. The reward function can adjust and optimize the delay and throughput of the upstream and downstream networks, effectively solving the traffic engineering problems in SDN. Zhang et al. [35] considered that in traditional traffic engineering, the optimal network performance needs to reroute as many data flows as possible, and frequent rerouting will lead to network interference. A critical flow rerouting-reinforcement learning algorithm (CFR-RL) based on the actor-critic framework was proposed. Some critical flows were selected for rerouting through the CFR-RL algorithm, and equal cost multi-path (ECMP) was used to forward most flows, effectively solving the problem of network service quality degradation and network interference caused by frequent rerouting. Fu et al. [36] proposed a deep Q-Learning reinforcement learning method to meet the requirements of low delay and low packet loss rate for mouse flow and high throughput and low packet loss rate for elephant flow in a data center network. Liu et al. [37] took the cache as a factor affecting the routing strategy, reorganized the cache and bandwidth of multiple network resources in the quantitative fraction of reducing the delay, expressed the state as a multidimensional space, and proposed a routing algorithm based on deep reinforcement learning, which improved the network throughput rate and robustness. Hossain et al. [38] designed an intelligent situational awareness routing algorithm to reduce the impact on application-driven quality of service through intelligent awareness and network management when the network is attacked. Yu et al. [39] proposed a DRL mechanism that utilizes black box technology to realize customizable and adaptive routing optimization, which simplified network management and maintenance.

The above methods effectively solve the defects of Q-table and accelerate model convergence by combining deep learning with reinforcement learning, which can process and adapt to complex and high-dimensional dynamic network environments and improve network performance. However, these methods do not consider the trend of the future state of network traffic, and some methods take the link weight value as the output of the action, which requires further calculation to obtain the route-forwarding path.

This paper proposes a routing method based on Dueling DQN deep reinforcement learning and network traffic state prediction. The SDN measurement mechanism is used to obtain the network traffic matrix in real-time, and the GRU [40] (Gate Recurrent Unit) prediction algorithms that is a

variant of the long short-term memory network (LSTM) [41], which discover the hidden and unknown network traffic states to enhance the awareness ability of the intelligent routing method. Then, the action strategy is output by the deep reinforcement learning agent, which directly outputs the optimal route-forwarding path under the current network state, and realizes the adaptive intelligent routing forwarding operation.

3 SDN intelligent routing optimization architecture and modelling

The overall structure of the intelligent routing method model under the SDN architecture designed in this paper is shown in Fig. 1, which mainly includes the data plane, control plane, management plane and knowledge plane. The functions of the four designed planes are introduced below.

3.1 Data plane

The data plane is composed of forwarding equipment based on the OpenFlow protocol in the SDN architecture, which is the entity that performs network data packet processing and is responsible only for forwarding or discarding the corresponding data packets according to the corresponding instructions of the control plane. The programmability of the data plane brings excellent convenience to the deployment and maintenance of the SDN. In the proposed intelligent routing method, the primary functions of the data plane are: (1) Providing global awareness information of the network by responding to the requests periodically sent in the control plane, such as the flow rate of the switch port, the number of bytes sent and received. (2) The control plane generates a route-forwarding strategy according to the optimal route-forwarding path generated by the agent, and the data plane forwards the data packets according to the route-forwarding strategy.

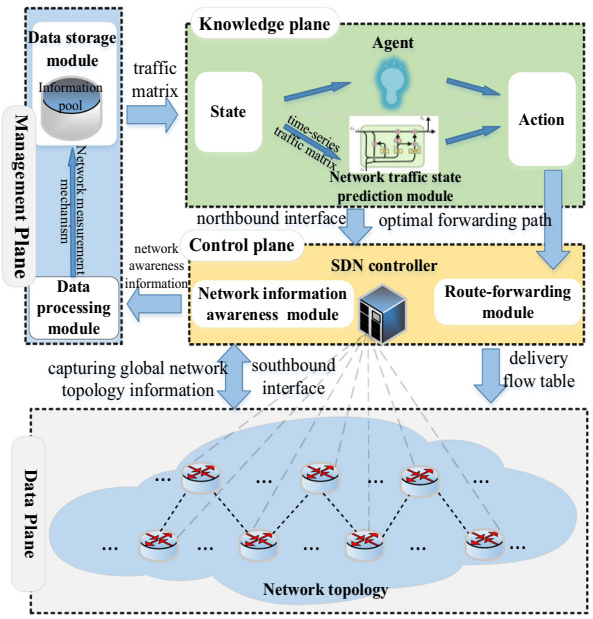

Fig. 1 The overall architecture of the intelligent routing method

3.2 Control plane

The SDN control plane interacts with the data plane through the southbound interface, periodically sends the corresponding request instructions to obtain the global network topology and link information, processes the requests of the upper application through the northwards interface, and provides services. The control plane is the central system of the SDN network architecture, which deploys two modules: the network information awareness module and the route-forwarding module. The functions of the network information awareness module are as follows: (1) Capturing the network topology information by means of the link layer discovery protocol (LLDP); (2) Periodically sending corresponding Request instructions to gain the port status information of each switch on the network to construct network awareness information. The functions of the route-forwarding module are as follows: (1) Finding the corresponding host node according to the optimal route-forwarding path obtained by the knowledge plane; (2) Based on the optimal forwarding path and the corresponding host node, the optimal routing forwarding strategy is generated and delivered to the data plane.

3.3 Management plane

In particular, the model of the intelligent routing method proposed in this paper appends a management plane to the SDN architecture, including a data processing module and a data storage module. The management plane converts the sensing information obtained in the control plane into bandwidth, delay, packet loss rate, used bandwidth and other network link status information through its internal data processing module and saves it in the information pool of the data storage module. Meanwhile, the management plane contains a large number of processing operations such as data monitoring and calculation, conversion and storage, which are important to ensure the performance of the proposed method and the stable operation of the network. To solve the above-mentioned problem, a measurement scheme based on the SDN network is deployed on the management plane as shown in Fig. 2. This measurement mechanism uses a multi-threading method. On the one hand, the fact that the delivery flow table and the network

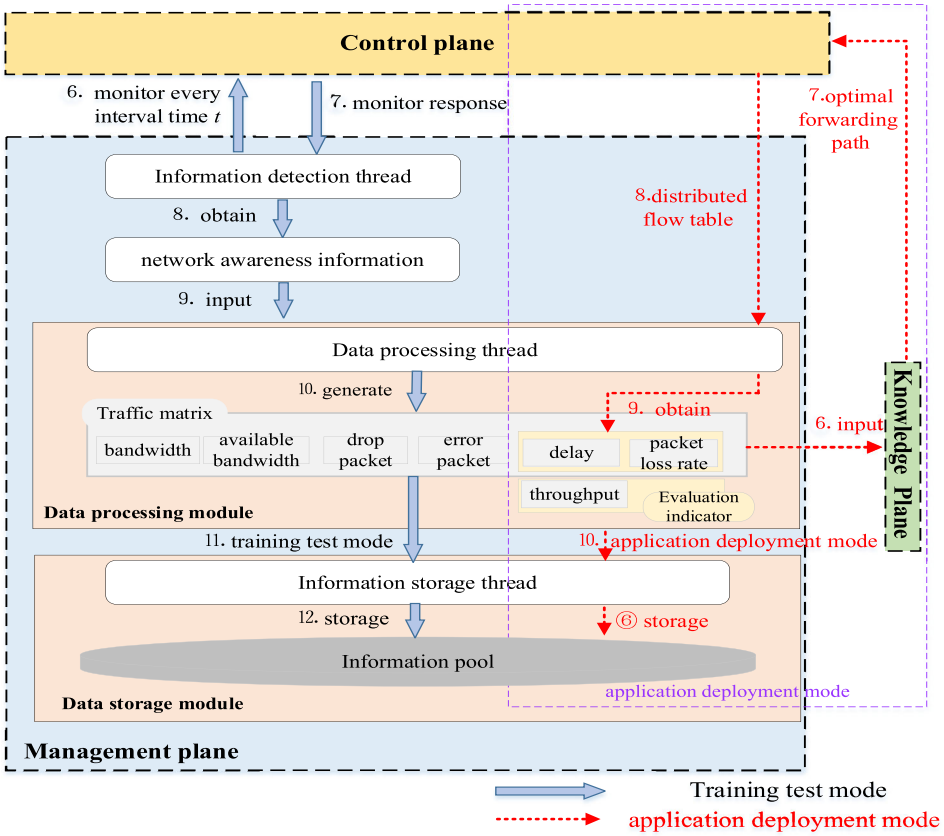

Fig. 2 Network measurement mechanism based on SDN

state measurement mechanism in the SDN environment are in the same thread, will cause the network to run incorrectly for a period of time. Furthermore, the knowledge plane needs to determine the optimal route-forwarding strategy in real-time according to the current network status, which requires frequent interaction with the management plane to obtain the network state information at the current moment. However, the single-threading measurement mechanism has difficulty successfully completing this interaction; On the other hand, the two operations of data processing and information storage are performed by different threads, which can not only accurately and timely process and store network link information, but also make full use of hardware resources such as processors during the algorithm model offline training. The subsequent experimental results prove that the network measurement mechanism based on multithreading can effectively obtain the network link status information while improving the performance of the intelligent routing method.

The main process of the SDN network measurement mechanism is as follows: An information detection thread is created that sends global network sensing information requests to the control plane every t seconds to monitor the current network status, and the global network sensing information obtained in response to these requests is returned via the control plane. Then, the obtained network sensing information is transmitted to a data processing thread for the conversion of network link information such as bandwidth, delay, and packet loss rate, which is used to construct the traffic matrix at the current moment. This data processing thread, which is the core of the SDN network measurement mechanism, is divided into two modes: training test mode and application deployment mode in intelligent routing method. As shown by the blue arrows in Fig. 2, in the training test mode, the traffic matrix constructed at each time point is saved to the information pool of the data storage module by an information storage thread for offline training model on the knowledge plane. As shown by the red lines in Fig. 2, in the application deployment mode, the knowledge plane uses the trained model to generate the optimal route-forwarding path in accordance with the current traffic matrix, and the control plane generates the optimal route-forwarding strategy and delivers the flow table. Thus, the SDN network measurement mechanism is used to obtain the link information in the current network state, and the three network evaluation indicators of throughput, delay and packet loss rate are calculated by means of a correlation equation and are saved in the information pool of the storage module for subsequent comparative experiments.

3.4 Knowledge plane

In the proposed SDN intelligent routing method, a knowledge plane is attached to the SDN architecture. The addition of knowledge plane to the SDN architecture, was first proposed by Clark et al. [42]; then, Mestres et al. [43] presented the concept of knowledge defined network and noted that the knowledge plane can integrate decision-making behaviour and reasoning processes into the SDN architecture and provide functions such as description, learning, and intelligence to support the routing decision-making process. Fundamentally, the knowledge plane [44] converts the global network link state information acquired from the management plane into knowledge through RL, and then intelligently develops corresponding strategies for the network based on this knowledge. In the proposed method, the knowledge plane includes DRL module and network traffic state prediction module. The DRL module, namely the agent, whose primary function is to build the DRL environment, converts the traffic matrix obtained by the management plane into the state space, and makes use of end-to-end, model-free DRL to continuously interact with the network environment and make the agent learn in the direction of higher reward value. When the algorithm model training tends to converge, the action is obtained according to the network state at the current moment, which is the optimal route-forwarding path.

The knowledge plane needs to obtain the traffic matrix in real-time in order to make routing decisions, which requires the SDN controller to respond frequently and continuously to process the data flows in the network. However, as the scales of the network and the user demands increase, the network will exhibit many different types of data flows, placing a high load on the controller and causing some data flows to be queued in the waiting state for a long time without being handled. This scenario may even cause the phenomenon of loop circulation and lead to network fluctuations, which will affect the correct operation of the network. Therefore, the SDN measurement mechanism designed in this article adopts a multithreaded method to periodically query the global network link states to effectively solve the high load problems caused by the continuous processing of a large number of different types of data flows by the controller, while competently satisfying the real-time requirements of the knowledge plane in obtaining the traffic matrix. However, some omissions in the network traffic matrix will still occur, which will affect the performance of the intelligent routing method. To address the problem that the proposed interval SDN measurement mechanism will miss some traffic monitoring information in the network, a network traffic state prediction module is also added to the knowledge plane to predict the missing traffic information and discover the hidden and unknown network traffic states.

4 Designed DRL-TP intelligent routing algorithm

The network topology in the data plane is an undirected graph, which can be represented by $= \{V, E, W\}$, where V represents the SDN switch nodes in the network topology, E represents the links between switch nodes in the topology, and W represents the weight of the links, which are generally set to the same constant. The DRL-TP intelligent routing algorithm is composed of a DRL algorithm and a network traffic state prediction algorithm, and the corresponding flowchart is shown in Fig. 3. First, the traffic matrix is obtained through the SDN measurement mechanism, and the predicted traffic matrix is obtained via the network traffic state prediction algorithm to form the state space of the deep reinforcement learning algorithm. Then, the DRL algorithm and the network traffic state prediction algorithm are trained offline, and the two trained algorithm models are combined into the DRL-TP algorithm model. Finally, the traffic matrix under the current network state is obtained in real-time according to the size of the sending flow, and the optimal route-forwarding path is generated through the DRL-TP intelligent routing algorithm. Subsequently, the designed DRL algorithm, network traffic state prediction algorithm, and DRL-TP intelligent routing algorithm are introduced.

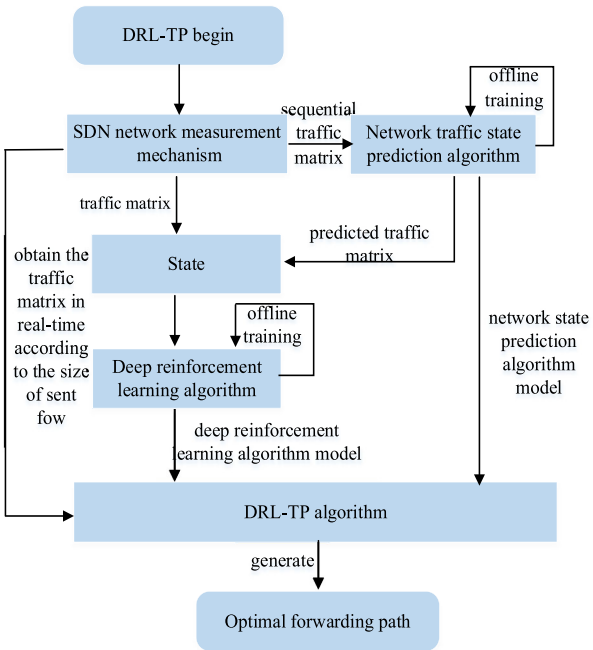

Fig. 3 DRL-TP algorithm flowchart

4.1 Deep reinforcement learning algorithm

Deep reinforcement learning is a combination of DL technology and RL technology, which is the most popular learning method in artificial intelligence. RL is based on the standard Markov decision process (MDP), using the "exploration and exploitation" method for learning, interacting with the environment, and taking corresponding actions to obtain the maximum cumulative reward, to achieve the goal of continuously strengthening one's decision-making ability. Fig. 4 shows a typical RL mechanism, consisting of the state space, action space and reward function. First, a state s_t is initialized based on the generated RL environment. After the agent performs the corresponding action a_t according to the state s_t , it returns a certain reward r_t and the next state s_{t+1} to the agent. the above learning process is repeated, and a greater reward is continuously obtained through gradual training to achieve optimal decision-making.

Deep reinforcement learning algorithm is a framework algorithm, that needs to design different state spaces, action spaces and reward functions for different problems and application fields. The design of the state space, action space and reward function in the DRL-TP intelligent routing algorithm based on the deep reinforcement learning framework is introduced below.

The state space(S): The state space can be expressed as S = TM, where *TM* refers to the traffic matrix within interval *t*. The traffic matrix is a multidimensional matrix composed of multiple two-dimensional matrix $M_{|V||*|V|}$ in interval *t*, $M_{|V||*|V|}$ is designed according to the Eq. (1):

$$\begin{aligned} m_{ij} &= w_1 \frac{1}{L_{bwij}} + w_2 L_{delayij} + w_3 L_{lossij} + w_4 L_{used_{bwij}} \\ &\quad + w_5 L_{dropsij} + w_6 L_{errorsij}\ i \\ &= 1, 2, ..., |V|; j = 1, 2, ..., |V| \end{aligned} \tag{1}$$

m_{ij} is the element that constructs $M_{|V||*|V|}$, which is composed of L_{bw}, L_{delay},L_{loss}, L_{used_bw}, L_{drops} and L_{errors}, namely the information matrix elements of the network link residual bandwidth, delay, packet loss rate, used bandwidth, discarded packets and error packets by additive mapping. Each network link information matrix contains the link information between all switch nodes at the current time; $w_l \in [0, 1], l = 1, 2, ..., 6$ is the weight factor of constructing traffic matrix elements; *i* and *j* are the names of switch nodes in the network topology; and $|V|$ represents the number of switch nodes in the network topology. The traffic matrix structure diagram is shown in Fig. 5.

Considering that the element values of each network link information matrix are quite different, the contribution of each network link information matrix to the formed traffic matrix cannot be objectively reflected, and the traffic matrix formed by additive mapping may be interfered with a larger link information matrix, resulting in excessive fluctuation in the training process of the DRL-TP intelligent routing algorithm that makes convergence difficult. Therefore, the Min–Max [45] method is used to normalize the traffic matrix as shown in Eq. (2) and the elements in the matrix are normalized to the specified range $[\mu_1, \mu_2]$.

$$\overline{m}_{ij} = \mu_1 + \frac{\left(m_{ij} - \min(TM)\right) \cdot (\mu_2 - \mu_1)}{\max(TM) - \min(TM)} \tag{2}$$

where $\overline{m_{ij}}$ is the element after normalization of the flow matrix; and *min(TM)* and *max(TM)* are the smallest and the largest elements in the traffic matrix, respectively.

The action space(A): the action space is the behaviour generated by the agent according to the current network state in RL, which is used to guide the agent to learn in a higher reward direction. The actions in the action space are composed of the forwarding link weight value and the forwarding path. The former does not need to store a colossal action space, but must use relevant methods to further transform into a forwarding path, while the latter directly outputs the forwarding path, and must store a colossal action space. An effective solution is to select the candidate path set as the action space, Refs. [30, 35, 37, 46] proved the effectiveness of this method. The action space designed in this paper is modified from the latter output of the forwarding path directly. Each action $a_t \in [0, 1, ..., k]$ corresponds to the forwarding path selection in the state of $s_t \in S$, which is composed of candidate path matrices $C_{|V|\times|V|}$. Each candidate path matrix contains the paths

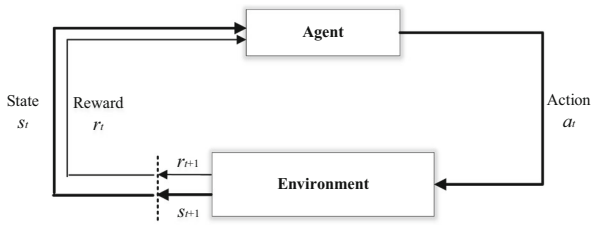

Fig. 4 Reinforcement learning mechanism

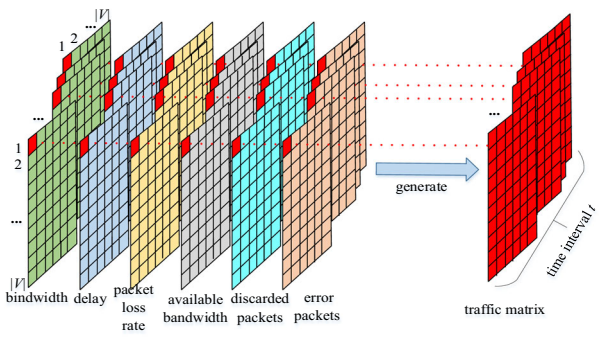

Fig. 5 Traffic matrix structure diagram

from source switch nodes to destination switch nodes, and the element in each candidate path matrix is the $path_{ij} = [i, ..., j]$ from switch node i to switch node j.

The reward function (R): The reward is the profit indicator generated by the RL agent after executing the corresponding action, which is used to evaluate the performance of the current action. The reward function of the DRL-TP intelligent routing algorithm model is shown in Eq. (3). The objective of optimization is to maximize the residual bandwidth and minimize delay, packet loss rate, used bandwidth, number of discarded packets and number of error packets. Since RL aims to continuously obtain the maximum reward, the positive correlation coefficient 1 is given to the maximization indicator, and the negative correlation coefficient -1;$\varphi_l \in [0,1], l = 1, 2, \ldots, 6$ is the weight factor to construct the reward function.

$$R = \varphi_1 L_{bw} - \varphi_2 L_{delay} - \varphi_3 L_{loss} - \varphi_4 L_{used_{bw}} - \varphi_5 L_{drops} - \varphi_6 L_{errors} \tag{3}$$

The improved DQN form Dueling DQN [47] based on value strategy is used as the framework of deep reinforcement learning algorithm of this paper. This method can solve the problem of overestimation of the value function caused by DQN, improve the learning performance and accelerate the convergence of the algorithm. In contrast to the DQN algorithm, Dueling DQN divides the value function of the deep Q network into two parts. The first part is related to the state s, which is called the value function and denoted as$VF(s, w, \beta)$; The second part which is related to both state s and specific action a, is called advantage function and denoted as $AF(s, a, w, \beta)$, as shown in Eq. (4):

$$Q(s, a, w, \beta, \lambda) = VF(s, w, \beta) + AF(s, a, w, \lambda) \tag{4}$$

where w is a common parameter of the two parts, β and λ are the unique parameters of the value function and advantage function respectively. Dueling DQN adopts the advantages of DQN experience replay and a greedy exploration mechanism. The experience replay mechanism refers to the fact that the agent stores the historical learning experience (s_t, a_t, r_t, s_{t+1}) in the experience pool, and then randomly samples the data in the experience pool for offline training to update the network parameters. The experience replay mechanism, in addition to reducing the correlation between data, and can reuse the experience samples, thereby substantially reducing the time cost of the model used to obtain experience samples, and increasing the learning efficiency of the model. The DRL-TP routing algorithm adopts the-*greedy* decay detection mechanism [48], as shown in Eq. (5):

$$a_t = \begin{cases} argmax_a\, Q_{policy}(\phi(s_t), a; \theta), & if \quad x > \varepsilon \\ random.rondom(\ ), & otherwise \end{cases} \tag{5}$$

for each training step, a variable $x \in [0, 1]$ is randomly generated. When $x > \varepsilon$, the agent performs the exploitation operation, otherwise, it performs the exploration operation; $\varepsilon \in [0, 1]$ is an adjustable parameter, that is set to a value close to 1 at the beginning of model training. As training proceeds, ε continuously decreases linearly, as shown in Eq. (6):

$$\varepsilon = e_{max} + u \cdot (e_{min} - e_{max}) \tag{6}$$

where $u = \min(1, step/totalStep)$ is the exploitation factor; *step* is the current training step; *totalStep* is the total step of the ε -*greedy* decay detection mechanism; and e_{max} and e_{min} are the maximum exploration factor and the minimum exploration factor, respectively. As the training step increases, ε decreases linearly to e_{min}.

A detailed implementation of Dueling DQN deep reinforcement learning algorithm is shown in Algorithm 1. *TM* includes the traffic matrix obtained by the SDN network measurement mechanism and the traffic matrix generated by the network traffic state prediction module. Line 1 initializes the policy network Q_{policy} and target network Q_{target} with the same structure, and an experience pool M for storing experience samples. Lines 3 to 8 store the experience samples learned by the agent for to train and update the model. Lines 9 to 12 perform the Dueling DQN network update operation. When the data in M exceed the batch size, the estimated value *p_value* is obtained by using the Q_{policy} network, and the target value *t_value* is obtained by using the Q_{target} network.Then the weight and deviation of the Q_{policy} network are adjusted via gradient descent and backpropagation, and the loss value of the network model is calculated using the mean square loss function. Line 14 updates the weights and biases of the Q_{policy} network to the Q_{target} network. Line 16 moves the agent to the next state for the next model learning. At the end of each training process, the route-forwarding paths from all source switches to destination switches in the current network state are obtained.

Algorithm 1 Dueling DQN Deep Reinforcement learning Algorithm

Input:
learning rate: *lr*, sample size: *batch*, discount factor: γ, weight factor: $wl\ ,\varphi l\ \ l = 1,2,\ldots,6$, attenuation parameter: ε, attenuation rate: *decay*, target network update frequency: *freq*, total number of training: *episodes*, traffic matrix: TM.

Output:
Forwarding paths from all source switch nodes to destination switch nodes in the network.

1: Initialize Q_{policy} and Q_{target} network weight θ, experience pool M
2: **For** *episode* ← 1 to *episodes* do:
3: The agent obtains the initialization state s_t
4: **While** next_state s_{t+1} is not final state do:
5: Update attenuation parameter $\varepsilon = \varepsilon - (steps * decay)$
6: According to equation (5), the agent obtains the action a_t of the current state s_t
7: According to equation (3), the agent obtains the current reward $r_t \leftarrow R(s_t, a_t)$
8: Store experience (s_t, a_t, r_t, s_{t+1}) into M
9: **If** len(M) >= *batch* then:
10: Sample *batch* data randomly from M
11: According to the value function equation (4), obtain *p_value*
$$t\_value = \begin{cases} r_t & \text{, if next state is final state} \\ r_t + \gamma max_{a'} Q_{target}(s_{t+1}, a'; \theta) & \text{, otherwise} \end{cases}$$
12: Execute gradient descent with loss = $(t\_value - p\_value)^2$ to to update the Qpolicy weight parameter θ
13: **End if**
14: **If** *step* % *freq* == 0 then:
Update Q_{target}, $\theta^{Q\,target} \leftarrow \tau\theta^{Q\,policy} + (1\text{-}\tau)\theta^{Q\,target}$
15: **End if**
16: $s_t \leftarrow s_{t+1}$
17: **End while**
18: **End for**

Algorithm 2 GRU Network Traffic State Prediction Algorithm

Input:
learning rate: *lr*, sample size: *batch*, input layer dimension: *input_dim*, hidden layer dimension: *hidden_dim*, output layer dimension: *output_dim*, serialization step: *seq*, network update frequency: *freq*, total number of training: *episodes*, traffic matrix: *TM*.

Output:
Prediction traffic matrix $TM_{predict}$.

1: **Initialize** GRU network weight θ
2: Execute the time-series operation on the traffic matrix to obtain two
time-series traffic matrices
time-series input matrix, $TM_{input} \leftarrow TM$
time-series input matrix, $TM_{target} \leftarrow TM$
3: **For** *episode* ← 1 to *episodes* do:
4: Initialize hidden
5: **For** t ← 0 to len($TM_{predict}$) do:
6: $TM_{output}^{t+seq+1}$, $hidden_{output} \leftarrow GRU(TM_{input}^{t,t+seq}, hidden)$
7: Execute gradient descent with loss = $(TM_{output}^{t+seq+1} - TM_{target}^{t+seq+1})^2$ to update the GRU network weight parameter θ
8: $hidden \leftarrow hidden_{outpu}$
9: **End for**
10: **End for**

Algorithm 3 DRL-TP Intelligent Routing Algorithm

Input:
Dueling DQN Algorithm hyperparameters, GRU algorithm hyperparameters, send flow size: *bw_list* .

Output:
The optimal forwarding path from all source switch nodes to destination switch.

1: Load the Dueling DQN and GRU models to build the DRL-TP algorithm model
2: **For** *bw* ← 0 to *bw_list* do:
3: Obtain the traffic matrix in the current network state, *TM*
4: all_paths ← DRL-TP (*TM*)
5: **End for**

4.2 Network traffic state prediction algorithm

With the development of the network business, the linear network traffic state prediction model has been difficult to adapt to the complex and nonlinear characteristics of the current network traffic state, resulting in the prediction effect of the network traffic state not being ideal. The network traffic state prediction model based on neural networks has obvious advantages, especially RNNs, which have excellent performance in addressing dynamic time-series prediction problems, and are widely used to predict the network traffic state. However, an RNN is not suitable for time-series prediction problems with long-term dependence, and the new RNN network LSTM and GRU with gating a mechanism can solve the above problems. Compared with LSTM, GRU has fewer parameters, which can reduce the risk of over-fitting the prediction model, make the prediction model reach the convergence state more quickly, and better meet the requirements of the knowledge plane designed in this paper for real-time traffic matrix acquisition. Therefore, GRU is adopted as the network traffic state prediction model.

As Algorithm 2 shows, the input layer, hidden layer and output layer dimensions are the basic parameters of the GRU network traffic prediction algorithm. Line 2 generates two time-series traffic matrices after the time-series operation, that are used as the input matrix and target matrix of the GRU algorithm model respectively. Line 4 initializes the hidden as part of the input parameters of the GRU algorithm model. Line 7 adjusts the weight and deviation of the GRU model according to the gradient descent and backpropagation algorithm, and uses the mean square loss function to calculate the loss value. Line 8 reassigns hidden as the next input parameter of GRU. When the training is completed, the GRU mode is used to generate the predicted traffic matrix as a component of the traffic matrix in Algorithm 1.

4.3 DRL-TP intelligent routing algorithm

The DRL-TP intelligent routing algorithm is composed of Dueling DQN DRL algorithm and GRU network traffic state prediction algorithm, as shown in Algorithm 3. First, the hyperparameters of Dueling DQN and GRU algorithms, including size of each sending flow, are input. Line 1 loads the offline trained Dueling DQN and GRU algorithm models to form the DRL-TP intelligent routing algorithm. Line 3 uses the designed SDN measurement mechanism to obtain the traffic matrix *TM* in the current network state according to the size of the current sending flow. Then, the DRL-TP algorithm is used to generate the optimal route-forwarding path *all_paths* from all the source switch nodes to the destination switch nodes under the current network state. The control plane generates the route-forwarding strategy according to *all_paths* and delivers the flow table. Finally, the data plane transmits the data packets according to the forwarding strategy.

5 Experimental analysis

5.1 Experimental environment

For the experiments reported in this paper, Mininet 2.3.0 [49] and Ryu 4.34 [50] were installed on an Ubuntu 16.04 system with 2GB of memory and a 4-core processor to construct an SDN environment, for which Mininet was used to build the SDN topology, Ryu was used as the SDN controller, and Iperf [51] was used to simulate the sending of data flows, as shown in Fig. 6. The data flows size of uniformly distributed and equal probability was sent each time to measure the network traffic matrix, and a total of 1616 traffic matrices were obtained. The equal probability function is shown in Eq. (7):

$$\text{prob}(f)=\frac{1}{(\lambda_2-\lambda_1)},\lambda_1<f<\lambda_2 \tag{7}$$

where λ_1 is 5, λ_2 is 100, f is the data flow size, and $\text{prob}(f)$ is the equal probability function used to select the current sent data flow size. The network topology used in the experiments is a modified New York City Center network [52]. As shown in Fig. 7, the network topology includes 14 nodes, and each node represents a switch supporting the OpenFlow protocol, and a host is mounted under each switch. To bulid a heterogeneous network environment, the bandwidth of the transmission link between switches is randomly set in the range of [15, 100] Mbit.

5.2 Prediction algorithm performance and experimental parameter analysis

First, the influence of the GRU network traffic state prediction algorithm on the performance of SDN intelligent routing method is analysed. Figure 8 shows that the reward obtained with using GRU is significantly higher than that obtained without using GRU because the GRU prediction algorithm can find hidden and unknown net-work traffic status in the SDN, and can predict the trend of network traffic status changes in the future, while these hidden and unknown network traffic status attributes are difficult to obtain with the network measurement mechanism based on SDN technology. Therefore, based on the GRU network traffic state prediction algorithm, the DRL-TP intelligent routing algorithm can explore a larger state space, learn better behaviour, and obtain a higher reward, which confirms that the GRU network traffic state prediction algorithm can improve the performance of the DRL-TP algorithm.

Both Eqs. (1) and (3) use weight factors to design the traffic matrix and reward function. These weight factors reflect the importance of each parameter indicator and determine the convergence speed of the algorithm model. Therefore, determining appropriate weight factors is foremost for the performance of the DRL-TP intelligent routing algorithm. First, the comparison results of the weight factors that construct the traffic matrix are shown in Fig. 9. In Fig. 9a, b, c, and d, the values in the upper left label represent L_{bw}, L_{delay},L_{loss}, L_{used_bw}, L_{drops} and L_{drops} respectively. As seen from the four figures in Fig. 9, compared to the traffic matrix composed of other weight factors, the traffic matrix composed of weight factors [0.6, 0.3, 0.1, 0.1, 0.1, 0.1] will obtain higher reward. Moreover, the performance of DRL-TP intelligent routing algorithm is better when the weight factors of L_{bw} and L_{delay} are larger, so we choose [0.6, 0.3, 0.1, 0.1, 0.1, 0.1] as $\varphi_l \in [0,1], l = 1,2,\ldots,6$ to construct the traffic matrix. Next, the weight

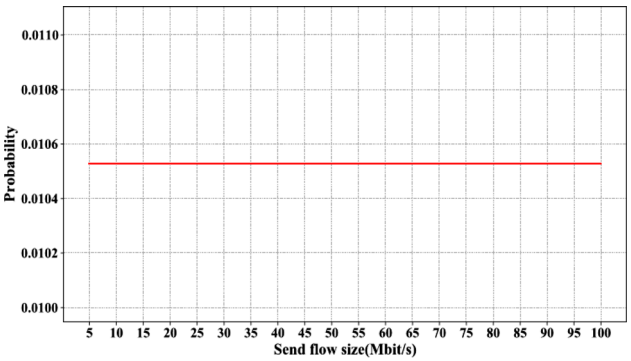

Fig. 6 Uniform distribution of sent data flow sizes with equal probability

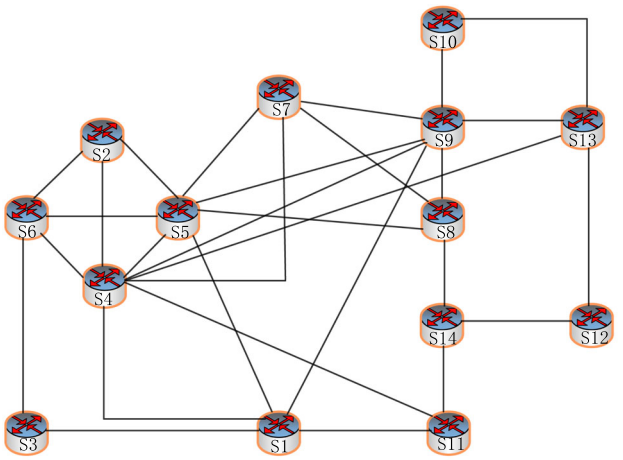

Fig. 7 The network topology diagram

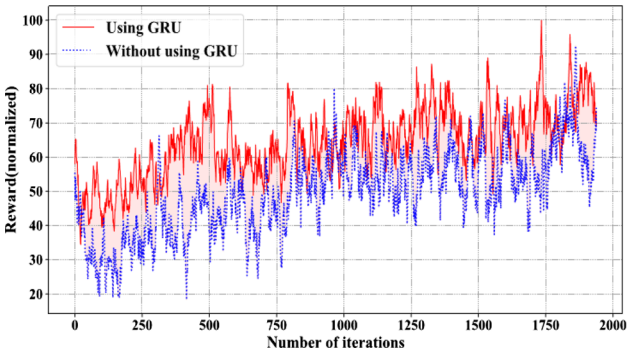

Fig. 8 Comparison of using GRU and without using GRU

factors that constitute the reward function are determined. As shown in Fig. 10a, b and c, the values in the upper left label represent L_{bw}, L_{delay},L_{loss}, L_{used_bw}, L_{drops} and L_{drops} respectively. Since the goal of the DRL-TP intelligent routing algorithm is to obtain the maximum reward, the positive correlation parameters of the reward function are set to be positive and the negative correlation parameters are set to be negative. Figure 10 shows that when the weight factor is [0.5, − 0.4, − 0.3, − 0.3, − 0.3, − 0.3], the corresponding reward function achieves the highest reward. Therefore, we choose [0.5, − 0.4, − 0.3, − 0.3, − 0.3, − 0.3] as $\varphi_l \in [0, 1]$, $l = 1, 2, ..., 6$ to construct the reward function. On the basis of the comparison analysis of the above weight factors, this experiment uses the weight factor [0.6, 0.3, 0.1, 0.1, 0.1] to construct the traffic matrix, and the weight factor [0.5, − 0.4, − 0.3, − 0.3, − 0.3, − 0.3] to construct the reward function.

In the SDN intelligent routing optimization problem, most of the research work constructs the state space using a traffic matrix composed of a single network link indicator such as bandwidth. The DRL-TP intelligent routing

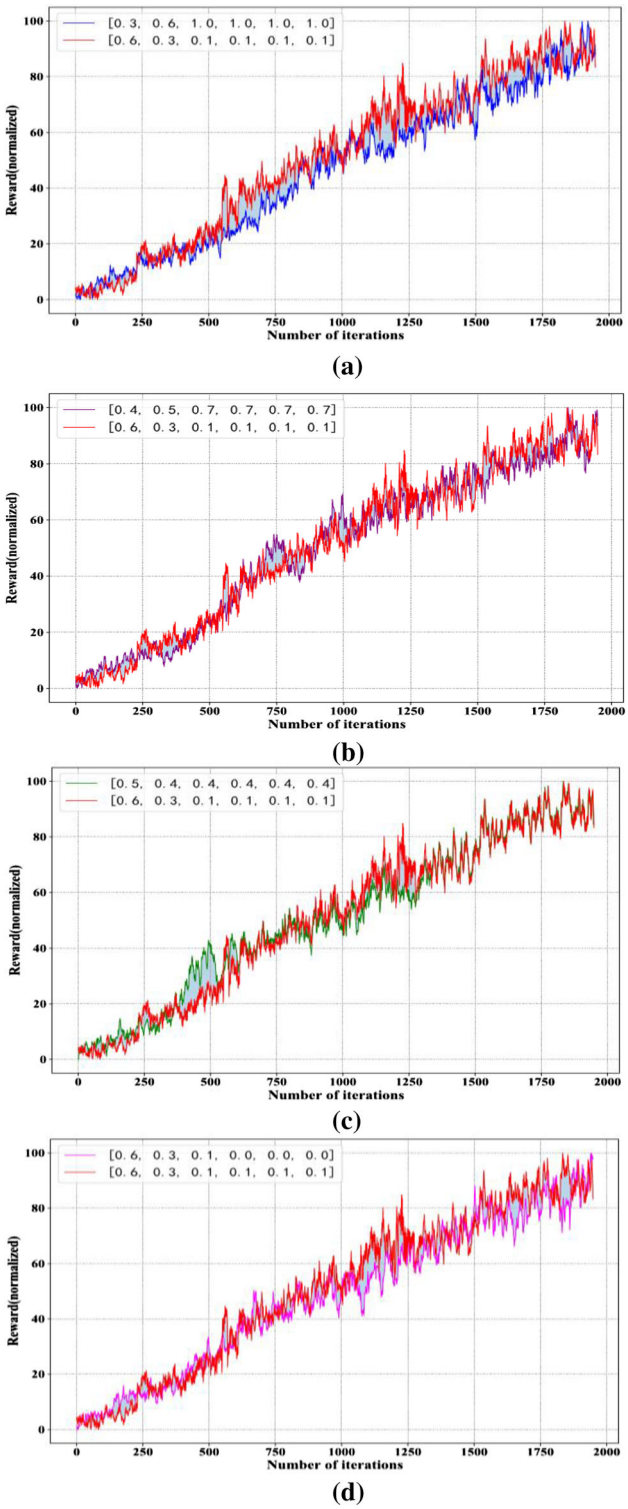

Fig. 9 Comparison of weight factors of traffic matrix

algorithm proposed in this paper uses a traffic matrix composed of six network link indicators, including bandwidth, delay, and packet loss rate to construct the state space. To verify the traffic matrix constructed by multiple network link indicators and improve the performance of the

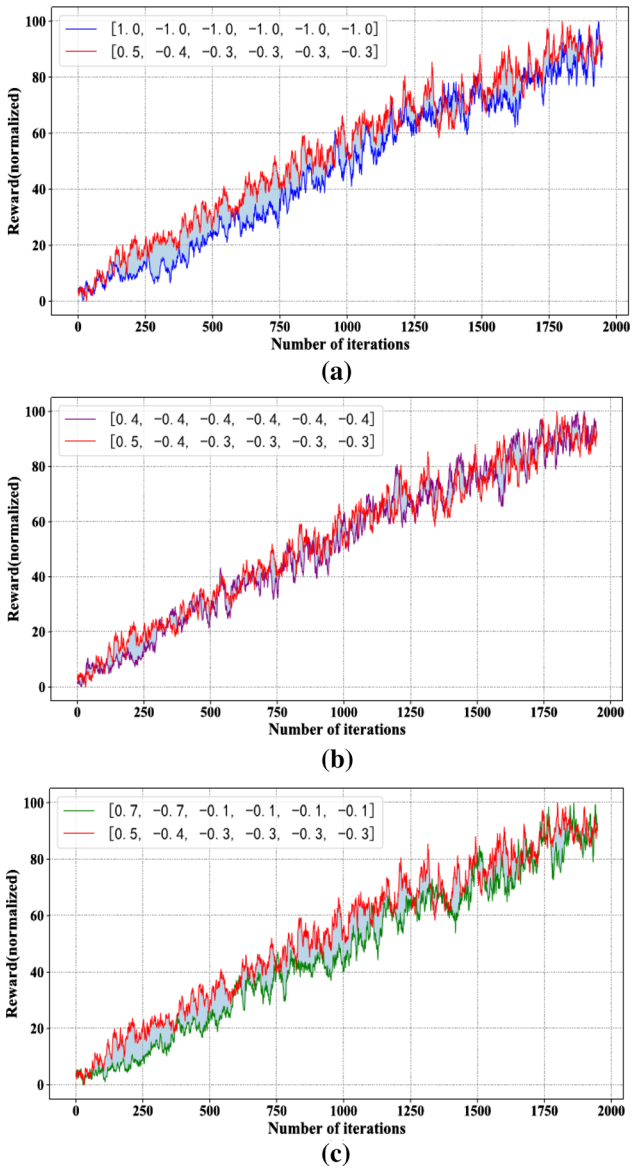

Fig. 10 Comparison of weight factors of reward function

SDN intelligent routing method, this paper conducts experiments, as shown in Fig. 11. In Fig. 11a, the red line representing the traffic matrix is composed of six link information matrices, namely,L_{bw},L_{delay},L_{loss},L_{used_bw}, L_{drops} andL_{drops}, which is obviously more than that of the blue line representing the traffic matrix composed of a single link information matrix ofL_{bw}. Figure 11 shows that the traffic matrix represented by the green line is composed of two link information matrices, L_{bw} andL_{delay}. The reward is higher than that of the traffic matrix represented by the blue line in Fig. 11a, but the overall reward is still inferior to that of the traffic matrix represented by the red line. In Fig. 11c, the purple line represents the traffic matrix, composed of three link information matrices L_{bw}, L_{delay} and L_{loss}.The corresponding reward is similar to that of the flow matrix represented by the red line. Some conclusions can be drawn from comparison of the three figures. (1) Compared with the traffic matrix composed of a single network link indicator using bandwidth, the traffic matrix composed of multiple network link indicators can obtain a higher reward and improve the DRL-TP intelligent routing algorithm; (2) Fig. 11c indicates that the improvement in the performance of the DRL-TP intelligent routing algorithm is related mainly to the three network link information matrices,L_{bw}, L_{delay} andL_{loss}, and the other three link information matrices have little impact. However, the fluctuation of the traffic matrix represented by the red line is better than that represented by the purple line because the weight factor of multiple indicators can avoid fluctuations caused by the excessive influence of one indicator on the algorithm model and accelerate the convergence of the

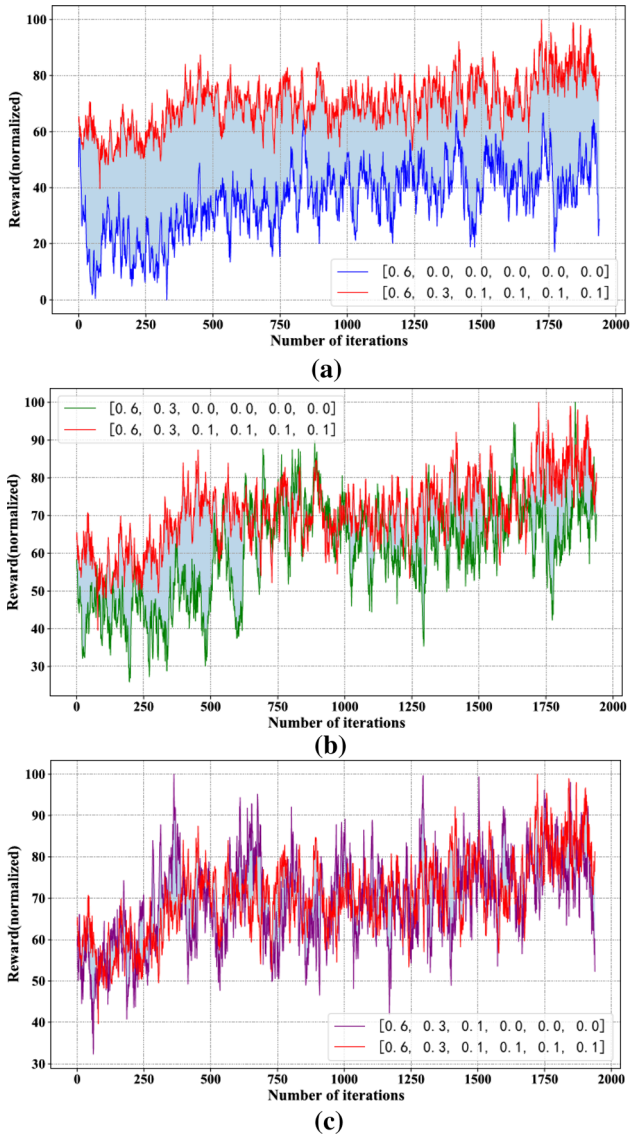

Fig. 11 Comparison of link indicators composing traffic matrix

algorithm model. Therefore, this paper uses six link information matrices to construct the traffic matrix.

5.3 Comparative experiments and results

RL algorithms can be divided into two categories: value-based and policy-based algorithm. Policy-based RL algorithms, which select action policies probabilistically, perform better for high-dimensional and continuous action spaces but have disadvantages such as a tendency to converge to local optima and lower efficiency of the policy evaluation process. In contrast, value-based RL algorithms simply select action strategies based on the highest value, and the action strategies are adjusted over time as the state value changes; consequently, these algorithms can converge to the global optimum faster and perform well for discrete action spaces. The action space in this paper is a discrete space consisting of candidate path matrices. Considering the discrete nature of the action space, coupled with the fact that the SDN intelligent routing method needs to be able to make optimal routing decisions in real time, the value-based Dueling DQN algorithm is used as the basis of the DRL module in the proposed DRL-TP intelligent routing algorithm. To verify the performance of this algorithm experiments were conducted to compare it with two policy-based RL algorithms: Proximal Policy Optimization (PPO) and Deep Deterministic Policy Gradient (DDPG). As shown in Figs. 12 and 13, in the early stage of training, compared with DDPG and PPO, the reinforcement learning algorithm based on Dueling DQN obtains relatively low reward and poor performance. However, as training proceeds, the performance of Dueling DQN algorithm gradually improves and can make the SDN intelligent routing algorithm reach algorithm reach convergence state in a short time.

This paper uses the Dijkstra and OSPF routing algorithm to compare with the DRL-TP intelligent routing algorithm. The following describes the design of the Dijkstra and OSPF algorithms.

Dijkstra routing algorithm: In the construction of the SDN topology, the link weight value W is set to 1 for the links between all switches, to obtain the shortest hop path from each source switch node to the destination switch node.

OSPF routing algorithm: The delay of each link in the network is obtained in real-time through the designed SDN measurement mechanism. The path between all switch nodes is obtained by the network link delay at the current time, and the shortest hop is selected as the route-forwarding path.

This paper designs three indicators to evaluate the impact of three routing algorithms on network performance: network throughput, network delay, and network

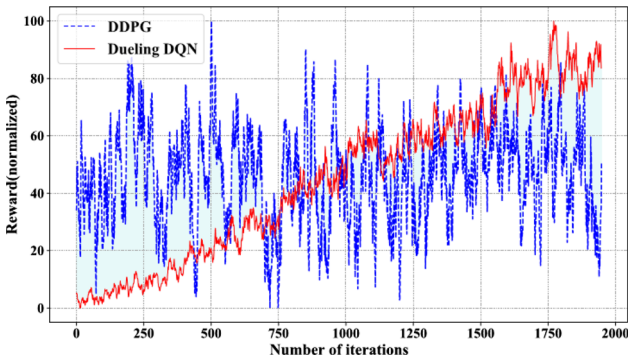

Fig. 12 Comparison of Dueling DQN and DDPG

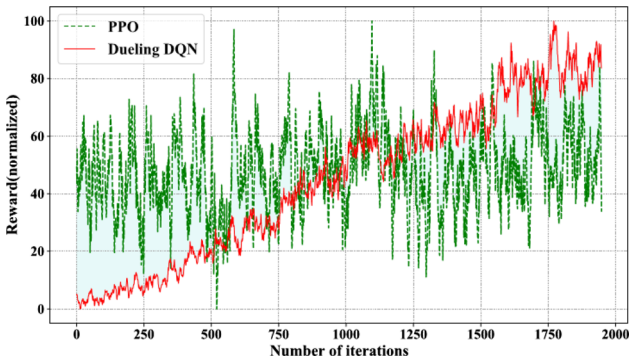

Fig. 13 Comparison of Dueling DQN and PPO

packet loss rate. First, the equations of network link throughput, network delay and network packet loss rate are given. The network link throughput is shown in Eq. (8):

$$throughput_{ij} = \frac{tx\_bytes(e_{i,j})}{bw_t(e_{i,j}) \cdot |\Delta t|} \tag{8}$$

where i and j are the names of switch nodes in the network topology; $throughput_{ij}$ is the throughput of link $e_{i,j}$ within time interval Δt; $tx\_bytes(e_{i,j})$ is the number of bytes sent from switch node i to switch node j within time interval Δt and $bw_t(e_{i,j})$ is the residual bandwidth of a link within time interval Δt.

The network link delay can be obtained by the measurement mechanism of the SDN delay [53]. As shown in Fig. 14, the SDN controller obtains T_1 and T_2 through the link discovery protocol (LLDP); then, the SDN controller periodically sends echo messages to switch A and switch B to obtain T_a and T_b respectively. The delay of the SDN network link was calculated according to the Eq. (9).

$$delay_{ij} = \frac{T_1 + T_2 - T_a + T_b}{2} \tag{9}$$

The network link packet loss rate is shown in Eq. (10):

$$loss_{ij} = \frac{tx_{pkts(e_{i,j})} - rx_{pkts(e_{i,j})}}{tx_{pkts(e_{i,j})}} \tag{10}$$

where i and j are the names of switch nodes in the network topology; $loss_{ij}$ is the packet loss rate of link $e_{i,j}$ within time interval Δt; $tx\_pkts(e_{i,j})$ is the number of data packets sent from switch node i to switch node j within time interval Δt and $rx\_pkts(e_{i,j})$ is the number of data packets received by switch node i from switch node j within time interval Δt. SDN network long-term operation may exhibit some fluctuations, resulting in certain deviations in the three performance indicators of network throughput, delay, and packet loss rate in a single measurement, which cannot accurately reflect the network performance at the current moment, thereby affecting the evaluation criteria. To mitigate the impact of network fluctuation, this paper takes the average value of multiple measurements to represent the network performance indicators, as shown in Eq. (11):

$$\begin{aligned} avg\_throughput &= \frac{\sum_n \sum_i \sum_j throughput_{ij}}{n \cdot |V| \cdot |V|} \\ avg\_delay &= \frac{\sum_n \sum_i \sum_j delay_{ij}}{n \cdot |V| \cdot |V|} \\ avg\_throughput &= \frac{\sum_n \sum_i \sum_j throughput_{ij}}{n \cdot |V| \cdot |V|} \\ i &= 1, 2, ..., |V|; j = 1, 2, ...|V| \end{aligned} \tag{11}$$

where *avg_throughput*, *avg_delay* and *avg_loss* represent the average throughput, average delay and average packet loss rate of the network measured every five times; i and j are the names of switch nodes in the network topology and $|V|$ represents the number of switch nodes. For convenience, network throughput, network delay, and network packet loss rate are used to refer to *avg_throughput*, *avg_delay* and *avg_loss* respectively. To compare the DRL-TP, Dijkstra and OSPF routing algorithms and verify the performance of the SDN intelligent routing algorithm designed in this paper, the network throughput, network delay and network packet loss rate were measured under multiple conditions by incrementing the sent flow size by 10Mbit/s each time.

Figure 15 compares the network throughput under the three routing algorithms. As the sending flow size increases, the network throughput of the three routing algorithms shows an increasing trend, but the growth trend of the DRL-TP intelligent routing algorithm is significantly greater than that of the Dijkstra and OSPF routing algorithms.

Figure 16 compares the network delay under the three routing algorithms. The overall network delay of the Dijkstra routing algorithm increases exponentially as the sending flow size increases because the Dijkstra routing algorithm only considers the shortest hop of the forwarding path between switches, when the sending flow size increases, the corresponding forwarding path in the network will appear congestion. However, the Dijkstra routing algorithm cannot adjust the forwarding path adaptively according to the network state, resulting in severe congestion in the network, thereby exponentially increasing the network delay. When the sending flow size is 10 – 40 Mbit/s, the network delay of the OSPF routing algorithm is similar to that of the DRL- TP intelligent routing algorithm because the route-forwarding path selected by the OSPF routing algorithm is based on the shortest hop path in consideration of the link delay, which can dynamically adjust the corresponding forwarding path according to the changes in the delay state in the network link. Therefore, the network delay under the OSPF routing algorithm is lower than that under the Dijkstra routing algorithm, and the network delay is similar to that under the DRL-TP intelligent routing algorithm when the sending flow size is not large. As the sending flow size increases, the network delay under the OSPF routing algorithm gradually exceeds that under the DRL-TP intelligent routing algorithm because the OSPF routing algorithm considers the delay to select the routing forwarding path and the routing forwarding strategy cannot be adjusted according to the overall indicator of the network, Therefore, network

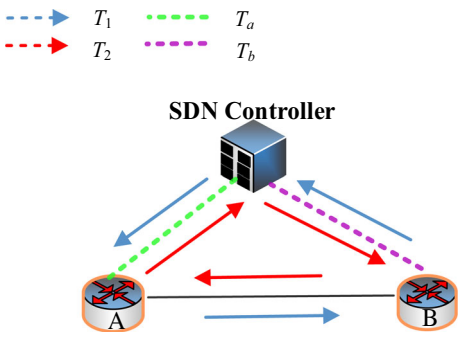

Fig. 14 Measurement of SDN link delay

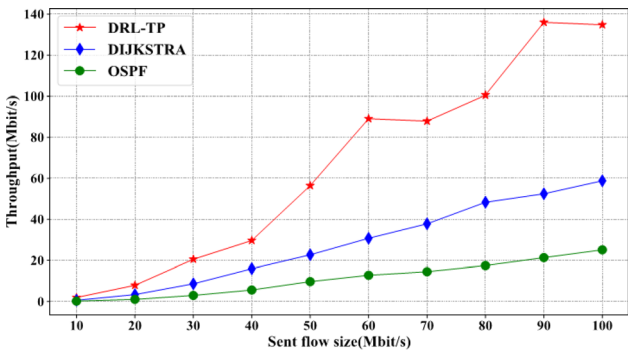

Fig. 15 Comparison of the network throughput

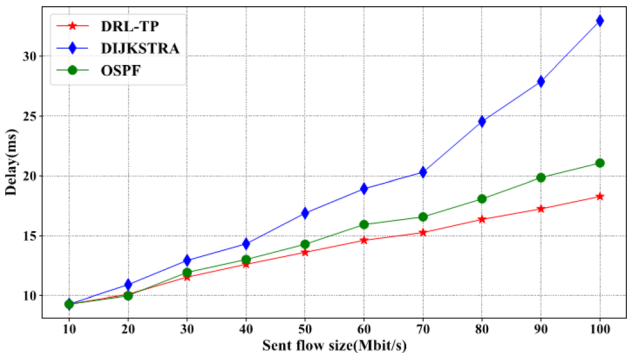

Fig. 16 Comparison of the network delay

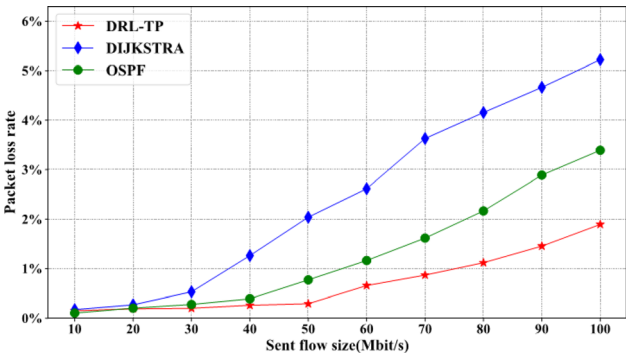

Fig. 17 Comparison of the network packet loss rate

congestion will still occur. The DRL-TP intelligent routing algorithm considers six network link indicators, including residual bandwidth, delay, and packet loss rate. As the sending flow continues to increase, the route-forwarding strategy can be adaptively and intelligently adjusted through the corresponding weight factors and the overall indicators of the network, and the congested route-forwarding paths can be transferred to the equivalent non-congested route-forwarding paths, which effectively alleviates network congestion.

Figure 17 compares network packet loss rate under the three routing algorithms. The link bandwidth of the proposed SDN network topology is set between 15 and 100 Mbit. When the sending flow size is between 10 and 20 Mbit/s, most links in the SDN network can forward data packets normally without network congestion. Therefore, when the sending flow size is 10–20 Mbit / s, the network packet loss rates of the three routing algorithms are similar. Nevertheless, as the sending flow size increases, the forwarding paths selected based on the Dijkstra and OSPF routing algorithms have different network congestion levels, resulting in a rapidly increasing packet loss rate on the network. The DRL-TP intelligent routing algorithm can dynamically adjust the route-forwarding strategy according to the current network state and select the optimal route-forwarding path. Therefore, compared with that of the Dijkstra and OSPF routing algorithms, the network packet loss rate is effectively reduced.

From Figs. 15, 16, and 17, it can be found that after the sending flow size reaches 40Mbit/s, the three indicators of network throughput, network delay, and network packet loss rate all show a more significant increasing trend, so the following conclusions can be drawn: (1) For the proposed SDN network topology, when the sending flow size increases to 40 Mbit/s, obvious congestion phenomenon will incur in the network; (2) When the network is congested, the Dijkstra and OSPF routing algorithms cannot effectively adjust the route-forwarding strategy, thereby reducing the performance of the network. In contrast, the proposed DRL-TP intelligent routing algorithm can monitor the network state in real-time and dynamically adjust the optimal route-forwarding strategy based on multiple network link indicators, which can guarantee the performance of the network even in the case of severe congestion and verifies the effectiveness of the DRL-TP intelligent routing algorithm.

6 Conclusion

With the continuous expansion of the scale of the SDN network, a variety of new network applications have appeared, and network traffic presents diversified and exponential growth characteristics. Finding a real-time adaptive intelligent route-forwarding strategy based on the SDN network state and demand is crucial to improve network performance and service quality. For this purpose, this paper proposes an SDN intelligent routing method based on Dueling DQN DRL and network traffic state prediction. Through the design of the SDN measurement mechanism, the current network state can be obtained in real time, and the DRL-TP intelligent routing algorithm is used to generate the optimal route-forwarding path in real-time. Compared with the Dijkstra and OSPF routing algorithms, the DRL-TP intelligent routing algorithm can adapt to complex and dynamic networks and significantly improve network performance, and has certain practical value for solving SDN routing optimization problems. Furthermore, the proposed method is based on the routing optimization problem under the an SDN with a single controller. When the scale of the SDN network gradually increases, it will bring high load to the single controller of the SDN. Therefore, the intelligent routing optimization problem based on an SDN multi-controller will be addressed in future work.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grant No. 62161006, No. 61861013 and No. 61662018, in part by the Science and Technology

Major Project of Guangxi No. AA18118031, in part by Guangxi Natural Science Foundation of China under Grant No. 2018GXNSFAA050028, in part by Director Fund project of Key Laboratory of Cognitive Radio and Information Processing of Ministry of Education under Grant No. CRKL190102, and in part by Guangxi Key Laboratory of Wireless Wide band Communication and Signal Processing No. GXKL06220110.

Source code The experimental code can be accessed at https://github.com/GuetYe/experiment-code

Declarations

Conflict of interest The authors declare no conflict of interest.

Data availability The dataset generated during this study by the SDN multi-threaded measurement mechanism designed in this paper through the flow measurement, which includes 1616 flow matrices, can be obtained from the author or accessed at https://github.com/GuetYe/experiment-data.

References

1. Nunes, B. A. A., Mendonca, M., Nguyen, X. N., Obraczka, K., & Turletti, T. (2014). A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys and Tutorials, 16*(3), 1617–1634. https://doi.org/10.1109/surv.2014.012214.00180
2. Sun, P., Yu, M., Freedman, M. J., Rexford, J., & Walker, D. (2015). Hone: Joint host-network traffic management in software-defined networks. *Journal of Network and Systems Management, 23*(2), 374–399. https://doi.org/10.1007/s10922-014-9321-9
3. Guerin, R. A., Orda, A., & Williams, D. (1997). QoS routing mechanisms and OSPF extensions. In *GLOBECOM 97. IEEE Global Telecommunications Conference*, pp. 1903–1908. IEEE. https://doi.org/10.17487/rfc2676
4. Verma, A., & Bhardwaj, N. (2016). A review on routing information protocol (RIP) and open shortest path first (OSPF) routing protocol. *International Journal of Future Generation Communication and Networking, 9*(4), 161–170. https://doi.org/10.14257/ijfgcn.2016.9.4.13
5. Ni, W., Huang, C., Wu, J., & Savoie, M. (2013). Availability of survivable Valiant load balancing (VLB) networks over optical networks. *Optical Switching and Networking, 10*(3), 274–289. https://doi.org/10.1016/j.osn.2013.02.002
6. Ibarz, J., Tan, J., Finn, C., Kalakrishnan, M., Pastor, P., & Levine, S. (2021). How to train your robot with deep reinforcement learning: Lessons we have learned. *The International Journal of Robotics Research, 40*(4–5), 698–721. https://doi.org/10.1177/0278364920987859
7. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444. https://doi.org/10.1038/nature14539
8. Botvinick, M., Ritter, S., Wang, J. X., Kurth-Nelson, Z., Blundell, C., & Hassabis, D. (2019). Reinforcement learning, fast and slow. *Trends in Cognitive Sciences, 23*(5), 408–422. https://doi.org/10.1016/j.tics.2019.02.006
9. Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena, 404*, 132306. https://doi.org/10.1016/j.physd.2019.132306
10. Ahn, C. W., & Ramakrishna, R. S. (2002). A genetic algorithm for shortest path routing problem and the sizing of populations. *IEEE Transactions on Evolutionary Computation, 6*(6), 566–579. https://doi.org/10.1109/tevc.2002.804323
11. Derbel, H., Jarboui, B., Hanafi, S., & Chabchoub, H. (2012). Genetic algorithm with iterated local search for solving a location-routing problem. *Expert Systems with Applications, 39*(3), 2865–2871. https://doi.org/10.1016/j.eswa.2011.08.146
12. Zhang, D. G., Liu, S., Liu, X. H., Zhang, T., & Cui, Y. Y. (2018). Novel dynamic source routing protocol (DSR) based on genetic algorithm-bacterial foraging optimization (GA-BFO). *International Journal of Communication Systems, 31*(18), 1–20. https://doi.org/10.1002/dac.3824
13. Parsaei, M. R., Mohammadi, R., & Javidan, R. (2017). A new adaptive traffic engineering method for telesurgery using ACO algorithm over software defined networks. *European Research in Telemedicine/La Recherche Europeenne en Telemedecine, 6*(3–4), 173–180. https://doi.org/10.1016/j.eurtel.2017.10.003
14. Jing, S., Muqing, W., Yong, B., & Min, Z. (2017). An improved GAC routing algorithm based on SDN. *IEEE International Conference on Computer and Communications (ICCC)*, pp. 173–176. https://doi.org/10.1109/compcomm.2017.8322535
15. Lin, C., Wang, K., & Deng, G. (2017). A QoS-aware routing in SDN hybrid networks. *Procedia Computer Science, 110*, 242–249. https://doi.org/10.1016/j.procs.2017.06.091
16. Truong Dinh, K., Kukliński, S., Osiński, T., & Wytrębowicz, J. (2020). Heuristic traffic engineering for SDN. *Journal of Information and Telecommunication, 4*(3), 251–266. https://doi.org/10.1080/24751839.2020.1755528
17. Ke, C. K., Wu, M. Y., Hsu, W. H., & Chen, C. Y. (2019). Discover the optimal IoT packets routing path of software-defined network via artificial bee colony algorithm. In *International Wireless Internet Conference*, pp. 147–162. Springer, Cham. https://doi.org/10.1007/978-3-030-52988-8_13
18. Shokouhifar, M. (2021). FH-ACO: Fuzzy heuristic-based ant colony optimization for joint virtual network function placement and routing. *Applied Soft Computing, 107*, 107401. https://doi.org/10.1016/j.asoc.2021.107401
19. Zhang, L., & Lei, Y. (2021). Particle swarm optimization-based information-centric networking intra-domain routing strategy. *Internet Technology Letters, 4*(1), e196. https://doi.org/10.1002/itl2.196
20. Valadarsky, A., Schapira, M., Shahaf, D., & Tamar, A. (2017). Learning to route. In *Proceedings of the 16th ACM workshop on hot topics in networks*, pp. 185–191. https://doi.org/10.1145/3152434.3152441
21. Sharma, D. K., Dhurandher, S. K., Woungang, I., Srivastava, R. K., Mohananey, A., & Rodrigues, J. J. (2016). A machine learning-based protocol for efficient routing in opportunistic networks. *IEEE Systems Journal, 12*(3), 2207–2213. https://doi.org/10.1109/jsyst.2016.2630923
22. Li, W., Li, G., & Yu, X. (2015). A fast traffic classification method based on SDN network. In *The 4th International Conference on Electronics, Communications and Networks*, pp. 223–229. Beijing, China. https://doi.org/10.1201/b18592-42
23. Zhou, X., Su, M., Liu, Z., Hu, Y., Sun, B., & Feng, G. (2020). Smart tour route planning algorithm based on naïve Bayes interest data mining machine learning. *ISPRS International Journal of Geo-Information, 9*(2), 112. https://doi.org/10.3390/ijgi9020112
24. Yanjun, L., Xiaobo, L., & Osamu, Y. (2014). Traffic engineering framework with machine learning based meta-layer in software-defined networks. In *2014 4th IEEE International Conference on Network Infrastructure and Digital Content*, pp. 121–125. IEEE. https://doi.org/10.1109/icnidc.2014.7000278
25. Tang, F., Mao, B., Fadlullah, Z. M., Kato, N., Akashi, O., Inoue, T., & Mizutani, K. (2017). On removing routing protocol from future wireless networks: A real-time deep learning approach for

intelligent traffic control. *IEEE Wireless Communications, 25*(1), 154–160. https://doi.org/10.1109/mwc.2017.1700244
26. Mao, B., Tang, F., Fadlullah, Z. M., & Kato, N. (2019). An intelligent route computation approach based on real-time deep learning strategy for software defined communication systems. *IEEE Transactions on Emerging Topics in Computing, 9*(3), 1554–1565. https://doi.org/10.1109/tetc.2019.2899407
27. Kato, N., Fadlullah, Z. M., Mao, B., Tang, F., Akashi, O., Inoue, T., & Mizutani, K. (2016). The deep learning vision for heterogeneous network traffic control: Proposal, challenges, and future perspective. *IEEE Wireless Communications, 24*(3), 146–153. https://doi.org/10.1109/mwc.2016.1600317wc
28. Hendriks, T., Camelo, M., & Latré, S. (2018). Q 2-routing: A Qos-aware Q-routing algorithm for wireless ad hoc networks. In *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 108–115. IEEE. https://doi.org/10.1109/wimob.2018.8589161
29. Chen, T., Gao, X., Liao, T., & Chen, G. (2019). Pache: A packet management scheme of cache in data center networks. *IEEE Transactions on Parallel and Distributed Systems, 31*(2), 253–265. https://doi.org/10.1109/tpds.2019.2931905
30. Casas-Velasco, D. M., Rendon, O. M. C., & da Fonseca, N. L. (2020). Intelligent routing based on reinforcement learning for software-defined networking. *IEEE Transactions on Network and Service Management, 18*(1), 870–881. https://doi.org/10.1109/tnsm.2020.3036911
31. Jin, Z., Zang, W., Jiang, Y., & Lan, J. (2019). A QLearning based business differentiating routing mechanism in SDN architecture. *Journal of Physics: Conference Series, 1168*(2), 022025. https://doi.org/10.1088/1742-6596/1168/2/022025
32. Yin, Y., Huang, C., Wu, D. F., Huang, S., Ashraf, M., & Guo, Q. (2021). Reinforcement learning-based routing algorithm in satellite-terrestrial integrated networks. *Wireless Communications and Mobile Computing*. https://doi.org/10.1155/2021/3759631
33. Zhao, L., Wang, J., Liu, J., & Kato, N. (2019). Routing for crowd management in smart cities: A deep reinforcement learning perspective. *IEEE Communications Magazine, 57*(4), 88–93. https://doi.org/10.1109/mcom.2019.1800603
34. Chen, Y. R., Rezapour, A., Tzeng, W. G., & Tsai, S. C. (2020). Rl-routing: An sdn routing algorithm based on deep reinforcement learning. *IEEE Transactions on Network Science and Engineering, 7*(4), 3185–3199. https://doi.org/10.1109/tnse.2020.3017751
35. Zhang, J., Ye, M., Guo, Z., Yen, C. Y., & Chao, H. J. (2020). CFR-RL: Traffic engineering with reinforcement learning in SDN. *IEEE Journal on Selected Areas in Communications, 38*(10), 2249–2259. https://doi.org/10.1109/jsac.2020.3000371
36. Fu, Q., Sun, E., Meng, K., Li, M., & Zhang, Y. (2020). Deep Q-learning for routing schemes in SDN-based data center networks. *IEEE Access, 8*, 103491–103499. https://doi.org/10.1109/access.2020.2995511
37. Liu, W. X., Cai, J., Chen, Q. C., & Wang, Y. (2021). DRL-R: Deep reinforcement learning approach for intelligent routing in software-defined data-center networks. *Journal of Network and Computer Applications, 177*, 102865. https://doi.org/10.1016/j.jnca.2020.102865
38. Hossain, M. B., & Wei, J. (2019). Reinforcement learning-driven QoS-aware intelligent routing for software-defined networks. In *2019 IEEE global conference on signal and information processing (GlobalSIP)* , pp. 1–5. IEEE. https://doi.org/10.1109/globalsip45357.2019.8969320
39. Yu, C., Lan, J., Guo, Z., & Hu, Y. (2018). DROM: Optimizing the routing in software-defined networks with deep reinforcement learning. *IEEE Access, 6*, 64533–64539. https://doi.org/10.1109/access.2018.2877686
40. Zhang, D., & Kabuka, M. R. (2018). Combining weather condition data to predict traffic flow: A GRU-based deep learning approach. *IET Intelligent Transport Systems, 12*(7), 578–585. https://doi.org/10.1109/mwscas.2017.8053243
41. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
42. Clark, D. D., Partridge, C., Ramming, J. C., & Wroclawski, J. T. (2003). A knowledge plane for the internet. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 3–10. https://doi.org/10.1145/863955.863957
43. Mestres, A., Rodriguez-Natal, A., Carner, J., Barlet-Ros, P., Alarcón, E., Solé, M., Muntés-Mulero, V., Meyer, D., Barkai, S., Hibbett, M. J., & Estrada, G. (2017). Knowledge-defined networking. *ACM SIGCOMM Computer Communication Review., 47*(3), 2–10. https://doi.org/10.1145/3138808.3138810
44. Xue, X., & Huang, Q. (2022). Generative adversarial learning for optimizing ontology alignment. *Expert Systems*. https://doi.org/10.1111/exsy.12936
45. Al Shalabi, L., & Shaaban, Z. (2006). Normalization as a preprocessing engine for data mining and the approach of preference matrix. In *2006 International conference on dependability of computer systems*, pp. 207–214. IEEE. https://doi.org/10.1109/depcos-relcomex.2006.38
46. Casas-Velasco, D. M., Rendon, O. M. C., & da Fonseca, N. L. (2021). DRSIR: A deep reinforcement learning approach for routing in software-defined networking. *IEEE Transactions on Network and Service Management*. https://doi.org/10.1109/tnsm.2021.3132491
47. Ban, T. W. (2020). An autonomous transmission scheme using dueling DQN for D2D communication networks. *IEEE Transactions on Vehicular Technology, 69*(12), 16348–16352. https://doi.org/10.1109/tvt.2020.3041458
48. White, S. R., Hanson, J. E., Whalley, I., Chess, D. M., & Kephart, J. O. (2004). An architectural approach to autonomic computing. In *International Conference on Autonomic Computing*, 2004. Proceedings, pp. 2–9. IEEE. https://doi.org/10.1109/icac.2004.1301340
49. Mininet. Accessed: Jan. 5, 2021. [Online]. Available: http://mininet.org/
50. Ryu. Accessed: Dec. 31, 2020. [Online]. Available: https://github.com/faucetsdn/ryu
51. IPerf. Accessed: Jan. 5, 2021. [Online]. Available: https://iperf.fr/
52. New York Metro IBX data center data sheet. Accessed: Dec. 31, 2020[Online]Available:https://www.equinix.com/resources/data-sheets/nyc-metro-data-sheet/
53. Li, Y., Cai, Z. P., & Xu, H. (2018). LLMP: Exploiting LLDP for latency measurement in software-defined data center networks. *Journal of Computer Science and Technology, 33*(2), 277–285. https://doi.org/10.1007/s11390-018-1819-2

Linqiang Huang was born in 1998. He is currently pursuing the master's degree with the School of Computer Science and Information Security, Guilin University of Electronic Technology. His main research interests include reinforcement learning and software defined networking.

Miao Ye received his B. S. degree in theory physics from Beijing Normal University in 2000 and his Ph.D. degree from School of Computer Science and Technology from Xidian University in 2006. He is currently a full professor and Ph.D. supervisor at Guilin University of Electronic Technology. His research interests include software defined networks, edge computing and edge storage, wireless sensor networks, deep learning.

Xingsi Xue received the B. S. degree in Software Engineering from Fuzhou University, China in 2004, the M.S. degree in Computer Application Technology from Renmin University of China, China in 2009, and the Ph.D. degree in Computer Application Technology from Xidian University, China in 2014. He is an associate professor at Center for Information Development and Management, Fujian University of Technology, and the director of Intelligent Information Processing Research Center, Fujian University of Technique. His research interests include intelligent computation, data mining and large-scale ontology matching technology. He is the member of IEEE and ACM

Yong Wang was born in 1964. He received his Ph.D. degree from East China University of Science and Technology, Shanghai, China, in 2005. He is currently a full professor and Ph.D. supervisor at Guilin University of Electronic Technology. His main research interests are cloud computing, distributed storage systems, software defined networks and information security.

Hongbing Qiu was born in 1963, Ph.D., professor of Guilin University of Electronic Science and Technology, doctoral supervisor of Xidian University and Guilin University of Electronic Science and Technology, visiting researcher at University of Minnesota, Twin Cities in 2012, communication theory of China Academy of Communications Member of the Signal Processing Committee, Director of the Key Laboratory of Cognitive Radio and Information Processing of the Ministry of Education, member of the Broadband Wireless IP Standard Working Group, etc. His main research directions are wireless communication, ultra-wideband communication, wireless sensor networks and software defined networks.

Xiaofang Deng received the B.Eng. and M.Eng. degrees in communication engineering from the Guilin University of Electronic Technology (GUET), China, in 1998 and 2005, and the Ph.D. degree from the South China University of Technology (SCUT), Guangzhou, China, in 2016. She was a Visiting Scholar with Coventry University, in 2017. She is currently an Associate Professor with the School of Information and Communication Engineering, GUET. Her research interests include cognitive networks, network economy, and information sharing.