Load Balancing Algorithm of Controller Based on SDN Architecture Under Machine Learning

Siyuan LIANG

Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of Posts & Telecommunications, Xi'an 710121, China

Wenli JIANG*

Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of
Posts & Telecommunications, Xi'an 710121, China
E-mail: wenlijiang0524@163.com

Fangli ZHAO

Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of Posts & Telecommunications, Xi'an 710121, China

Feng ZHAO

Shaanxi Key Laboratory of Information Communication Network and Security, Xi'an University of Posts & Telecommunications, Xi'an 710121, China

Abstract With the rapid development of cloud computing and other related services, higher requirements are put forward for network transmission and delay. Due to the inherent distributed characteristics of traditional networks, machine learning technology is difficult to be applied and deployed in network control. The emergence of SDN technology provides new opportunities and challenges for the application of machine learning technology in network management. A load balancing algorithm of Internet of things controller based on data center SDN architecture is proposed. The Bayesian network is used to predict the degree of load congestion, combining reinforcement learning algorithm to make optimal action decision, self-adjusting parameter weight to adjust the controller load congestion, to achieve load balance, improve network security and stability.

Keywords software defined networks (SDN); Bayesian network; reinforcement learning

1 Introduction

With the rapid development of Internet technology and mobile communication technology, the infrastructure, resources and structure of network system become more and more complex. In order to manage, maintain and optimize the network system more effectively, more intelligent

Supported by the National Natural Science Foundation of China (61875164), Shaanxi Provincial Key Research and Development Project (2020GY-059) and Special Scientific Research Program of Shaanxi Education Department in 2017 (17JK0702)

Received February 11, 2020, accepted June 5, 2020

^{*} Corresponding author

devices need to be deployed. As an important carrier of data, network structure becomes more and more complex with the exponential growth of Internet data. In a network, in addition to the information contained in the node itself, the relationship between nodes also has great value of data mining. As a new network technology, SDN provides powerful functions for network control and new opportunities for network security research. In multi-controller condition, load balancing of controller is serious problem which impacts the controller resource utilization rate^[1-3], it is one of the important methods to ensure the quality of network service, it is mainly to expand the data processing ability of the network by reasonably scheduling the network and computing resources, so as to avoid network congestion and improve the performance and robustness of the network, based on the network security, realize the network security optimization and load balance of the controller.

The emergence of SDN, from a global perspective, getsrid of the limitation of hardware on network architecture, reduces the cost of network management, and reduces the difficulty of management and operation. SDN introduces a controller as the brain of the whole network, namely the central controller. It separates the transmission layer and control layer of the network from each other, that is, decouples the control plane and data plane of traditional network equipment. The central controller is responsible for the entire network control, and the control logic is supervised by the control plane, making network management more convenient. The efficiency of network operation, maintenance and management is improved, the central control of the network is more convenient, and the unified external interface is provided through the network resources managed by the data plane, making the network open and programmable[4-6]. In the field of artificial intelligence, reinforcement learning has become an extremely important core technology, and people's attention is also focused on research and application of reinforcement learning. How to make a decision is the essential problem of reinforcement learning. The agent is in the surrounding environment and gains experience through continuous exploration to make the best action choice. Reinforcement learning algorithms can explore the surrounding environment through agents, and without prior knowledge, continue to train for many cycles to make the most correct action decision^[7-12]. Many researchers at home and abroad have a lot of interest in the study of Bayesian networks. The research directions mainly include Bayesian network structure learning, parameter learning and practical applications. The premise of parameter learning is how to construct a reasonable network structure. Therefore, Bayesian network structure learning, this research direction is the core research direction of the three^[13-17]. Through continuous development and progress of SDN, in order to achieve SDN load balancing, and to prevent and solve the occurrence of SDN link congestion, many domestic and foreign scholars in the field of SDN continue to study SDN load balancing algorithms. The transmission of information in the network needs to ensure its transmission speed and quality. The SDN load balancing optimization strategy mainly includes the following two aspects, namely SDN controller and link load balancing^[18-22]. Therefore, the load balancing problem of SDN controller has become the focus and hotspot of SDN research in recent years. Recently, there are many studies focus on the reinforcement learning method in load balancing of SDN controllers, but it has not been combined with Bayesian method [23-25]. This paper proposes a self-learning load balancing algorithm for SDN controller based on Bayesian

network (BN) which predict the load congestion. And then this algorithm selects the optimal action strategy according to the prediction result by the reinforcement learning. The simulation results indicate this proposed algorithm can improve the performance of controller load balancing in SDN, to improve network security and stability.

2 Data Center Block Diagram Based on SDN Architecture

SDN technology is applied to the data center network, which enables the network resources to schedule together with computing resources and storage resources, and provides the open interface to the upper application, and further provides the ability of new business development^[26-29]. The data center system architecture built with SDN is shown in Figure 1. The system is divided into three parts.

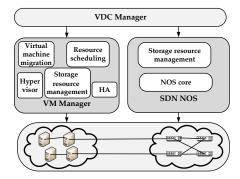


Figure 1 Data center block diagram based on SDN architecture

1) VDC Manager

The virtual data center manager is responsible for the realization of the business logic of the whole multi tenant data center. It controls the calculation and storage of network resources in a unified way with the interface of VM Manager and SDN NOS to ensure the simultaneous scheduling of calculation resources and network resources. VDC manager stores the corresponding relationship between virtual machine and network configuration information of each tenant, provide users with a logical view, so that each user can see their own virtual network, and according to their own policies for management, while not interfering with the network of other tenants. User operations on virtual machines and virtual networks are translated into calls to sdn NOS and VM Manager^[30,31].

2) SDNNOS

That is to say, SDN controls the entities, establishes the whole network view, and is responsible for the overall management of network resources, control SDN forwarding device of data center through openflow protocol. The software part of SDN NOS includes the core of NOS, that is, the core of network operating system, provide running environment and system services for upper application. Various modules can be run on top of NOS, and SDN NOS provides open API to VDC manager^[32,33].

3) VM Manager

That is, virtual machine manager, which is responsible for managing computing resources and storage resources, and reporting events such as virtual machine creation and migration, The internal modules include virtual machine migration, high availability (NA), resource scheduling and so on. It provides an interface to the VDC manager.

3 Bayesian Network Design

The entire network is treated as a black box when the Q-learning algorithm is employed solely and it is easy to get the Q value to a local maximum. Therefore, this paper uses BN to solve the problem of Q value local maximum^[34]. The original BN structure is based on reasoning and domain knowledge, as shown in Figure 2.

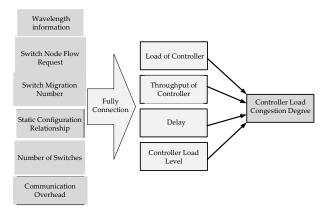


Figure 2 Initial BN architecture

The BN needs to define the controller's load level before predicting the load congestion level of the SDN controller, and controller load level C is defined as follows:

$$C = \frac{N_{Cj}}{N_S} \cdot \frac{L_{Cj}}{L_S}. (1)$$

In Formula (1), N_{Cj} is defined as the number of switches connected by controller C_j , N_S is defined as the total number of switches in the topology, L_{Cj} is defined as the load of all switches connected by controller C_j , L_S is defined as the total load of all switches in the topology network. The congestion level of SDN controller is predicted by BN. The load congestion level of SDN controller is predicted by using controller parameters such as controller load, delay D, number of switches migration, throughput T and controller load level C. The calculation of controller load congestion degree is shown in Equation (2):

$$P(C|L_{Cj}, D, T) = \frac{P(L_{Cj})P(D|L_{Cj}, C)P(T|L_{Cj}, C)}{P(L_{Cj}, D, T)} \sum_{N} \left(P(C|N) \frac{P(L_{Cj}|N)P(N)}{P(L_{Cj})} \right)$$

$$= \frac{P(D|L_{Cj}, C)P(T|L_{Cj}, C)}{P(L_{Cj}, D, T)} \sum_{N} \left(P(C|N)P(L_{Cj}|N)P(N) \right). \tag{2}$$

In Formula (2), N denotes the number of switch migration. Each congestion level corresponds to a predefined weight set. Therefore, after predicting the load congestion degree of SDN controller, the return function is calculated by Formula (3) using the corresponding weight set $w = \{w1, w2, w3\}$. The basic idea is to use a smaller weight for a higher level of congestion,

adjusting the parameter weight size to adjust the load congestion of the SDN controller, and thus achieving load balancing for the SDN controller.

4 Reinforcement Learning Model for SDN Controller Under Load Balancing

In general, Q-learning algorithm requires each node to record a table, that is, a Q table storing Q values. This algorithm requires each node to record two tables, one is Q table for storing Q value, and the other is BN table for storing prior probability and conditional probability.

4.1 State Space and Action Space

The Q-learning algorithm of SDN controller self-learning load based on Bayesian network needs to divide the state space first. Its state space is defined as a set of all controllers in SDN system, namely $\{State\} = \{C_1, C_2, C_j, \dots, C_{N_c}\}$. The Q-learning algorithm of SDN controller self-learning load based on Bayesian network defines the action space as the connection of all switches to the core controller, namely $A = \{(S_1, C_1), \dots, (S_i, C_j), \dots, (S_{N_s}, C_{N_c})\}$. Each element is defined as the corresponding action of switch S_i connected to controller C_j .

4.2 Return Function

The self-learning load balancing algorithm of SDN controller based on Bayesian network records the relevant parameters of SDN controller after each migration, such as controller load L_{Cj} , delay D during migration, throughput T, and so on. These parameters are important indicators to measure the load balancing algorithm of the controller, the return function can be obtained by using the load parameters of the SDN controller.

$$r(s,a) = w_1 \cdot L_{Cj} + w_2 \cdot T - w_3 \cdot D. \tag{3}$$

In Formula (3), w_1 is the weight of the controller's load L_{Cj} ; w_2 is the weight of the throughput of the switch during migration, and throughput T is the number of services that the controller can handle per second. w_3 is the weight of the average delay D from the switch to the controller in the migration of switches. When action a is executed under states, the larger the return function is, the larger the corresponding Q value is.

4.3 Value Function

In the learning process of Q-learning algorithm, iterative calculation is used to approximate the value function. A Q-value matrix is preset, the matrix is initialized and the initial value is set to 0. Q-value iteration updates are as in (4):

$$Q_{t+1}(s,a) = (1-a)Q_t(s,a) + \alpha(r(s,a) + \gamma \max Q_t(s,a_n)).$$
(4)

In Formula (4), s and a represent the current state and action of the controller respectively, and a_n represents the next action taken by the current state. At t+1 iterations, State s chooses action a, α is the learning rate, which is used to control the difference between the old Q value and the new Q value. If $\alpha = 0$, the function will not iterate; if $\alpha = 1$, it is a Bellman equation. γ is the discount factor and the value is between [0,1]. If $\gamma \to 0$, it means that the reward value

of current behavior is more considered in controller decision-making; if $\gamma \to 1$, it means that the reward value of future behavior is more considered in controller decision-making. $\max Q_t(s, a_n)$ is defined as the maximum reward value for selecting the next action a_n .

4.4 Action Decision

When migrating an switch, the action decision is made according to the relevant Q value of the current SDN controller, and the migration decision is made based on Boltzmann probability distribution. The action with the highest Q value is not always selected in the next action selection. The action selection strategy expression of self-learning load balancing algorithm for SDN controller based on Bayesian network is shown in (5):

$$P(a|s) = \frac{e^{\beta Q_t^j(s,a)}}{\sum_{l \in A(s)} e^{\beta Q_t^j(s,l)}}.$$
 (5)

In Formula (5), at the t-th iteration, state s chooses action a and $Q_t^j(s, a)$ is defined as the Q value of action a selected by current controller j under state s. A(s) is defined as a set of optional actions selected under state s, and β is a parameter.

5 Self-Learning Load Balancing Design of SDN Controller Based on Bayesian Network

SDN controller self-learning system based on Bayesian network, The core controller node is defined as C_j ; the common switch node is defined as S_i ; the connection between the switch and the controller is defined as b_{ij} , with a value of 0 or 1; the total load of all switches in the topology is defined as L_S ; and the number of all controllers in the topology is defined as N_C .

$$N_{Cj} = \sum_{S_i \in , C_j \in C} b_{ij}, \tag{6}$$

$$L_{cj} = \sum_{s} P(s_i), \tag{7}$$

Object: min
$$\sqrt{\sum_{j=1}^{N_C} \left(L_{Cj} - \frac{1}{N_C} \sum_{j=1}^{N_C} L_{Cj} \right)^2}$$
. (8)

Formula (6) defines the number of switches connected to controller j, that is N_{Cj} ; Formula (7) defines the load of the controller, and the load of controller L_{Cj} defines the amount of data transmitted by signals from all switches connected to it. The optimization objective of this algorithm is to minimize the standard deviation of controller load, it can be presented as load balance degree, as shown in Formula (8). To obtain the solution, the inherent constraints of SDN controller load balancing problem are delay constraints, cost constraints, binary constraints and algebraic constraints.

s.t.
$$\min(\max(d_{ij})) \le T_D,$$
 (9)

$$e < T_E, (10)$$

$$\forall i, \ \exists j, \ b_{ij'} = 0, \ \text{if} \ j \neq j', \tag{11}$$

$$\sum_{i=1}^{N_c} = N_S. {12}$$

Delay constraints: The delay between the controller and the switch is defined as d_{ij} . The minimum maximum delay between the controller and the witch must be less than the threshold T_D , as shown in Formula (9). Cost constraints: The number of migrated switches is defined as the migration cost, marked as e, and its maximum value must be less than the threshold T_E , as shown in Formula (10). Binary constraints: Switches are only connected to dedicated controllers, as shown in Formula (11). Algebraic constraints: The total number of switches connected to the controller in SDN is equal to N_S , as shown in Equation (12).

Bayesian network is used to predict the degree of load congestion. Based on the prediction results, the optimal action decision is made by reinforcement learning. The return function and value function are calculated by using the parameter weight set. In the learning process, the self-regulation of the parameter weight is realized by using the feedback mechanism, and the load congestion degree of the controller is adjusted so as to realize the load balance and improve the load balance efficiency of the controller.

6 Simulation Analysis and Results

6.1 Algorithmic Simulation Environment

The simulation language of this paper is Python 3.6, the CPU is Intel (R) core (TM) i5-2450 M CPU, 2.50 GHz, the installed memory is 4 GB PC, 64 bit win7 operating system. NSFMET network topology with high recognition is adopted as the experimental topology, as shown in Figure 3 (the data in the link in the figure is (cost, delay)). The network topology consists of 3 controllers and 11 switches. According to the principle of proximity, 11 switches are divided into three core controllers. In the process of simulation, the congestion degree of the controller is predicted by parameter delay D, controller load L_{Cj} and throughput T, and then the weight of the parameters is adjusted to minimize the congestion degree w_1, w_2, w_3 of the controller, so as to realize the load balance of the controller. The load of each controller in the topology is limited. When the load changes beyond its range, the optimal strategy should be selected to transfer the switch connected to other core controllers to the core controller for SDN controller load balancing. Similarly, when other core controllers are overloaded, the switch connected to the core controller can also be transferred to other core controllers for SDN controller load balancing. In order to ensure the reliability of the simulation data, we repeated the simulation algorithm many times to eliminate the interference of unrelated factors and record the relevant simulation data.

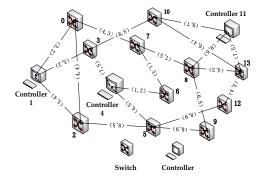


Figure 3 NSFMET network topology

6.2 Analysis of Load Balance Average Delay (AD) During Migration

Firstly, the influence of migration times of switch on the average delay between switch and controller is simulated. We will compare the self-learning load balancing algorithm of SDN controller based on Bayesian network proposed with the average delay change in the process of load balancing algorithm of SDN controller based on Q-Learning in [25], and the simulation results are shown in Figure 4. It can be seen that with the increase of migration times of switch, the average delay from switch to controller is reduced, and the self-learning load balancing algorithm of SDN controller based on Bayesian network is obviously better than Q-learning algorithm. With the increase of migration times of switch, the average delay of Q-learning algorithm from switch to controller is 1.9825, while the average delay of the proposed algorithm from switch to controller is 0.4976, and the average delay used is reduced by 74.9%. It is shown that the self-learning load balancing algorithm based on Bayesian network designed is much better than Q-learning algorithm in time delay, which improves the load balancing efficiency of the SDN controller.

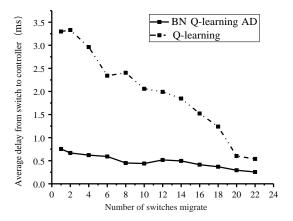


Figure 4 Analysis Chart of AD during migration

6.3 Analysis of Load Balance Degree (LBD) During Migration

Then the influence of the change of the migration times of the switch on the load balance degree of the controller is simulated, and the change of the load balance degree of the Q-learning algorithm and the self-learning load balance algorithm of the SDN controller based on the Bayesian network when the migration times of the switch change is compared and analyzed. The simulation results are shown in Figure 5. It can be seen that with the increase of migration times of switch, the load balancing degree of Q-learning algorithm and the proposed algorithm will decrease. In the whole simulation process, with the increase of migration times, the curve convergence speed of Q-learning algorithm is slow, the curve convergence value is 0.672, while the curve convergence value of the proposed algorithm is 0.545, and its convergence value is lower than Q-learning algorithm. In the process of simulation, the average value of load balance degree of Q-learning algorithm is 1.918, and Bayesian network is introduced to predict the load congestion degree of controller, furthermore, combined with Q-learning algorithm, the controller load balancing is faster. The average load balancing degree of the proposed algorithm

is 1.391, the average value of the controller load balancing degree is reduced by 27.5%, and the convergence value of the controller load balancing degree is reduced by 19%. It can be seen that the proposed algorithm has faster curve convergence rate, smaller convergence value, better load balancing degree and better load balancing function of the controller.

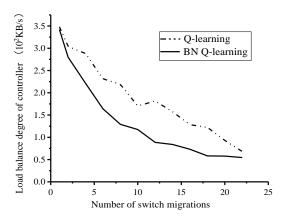


Figure 5 Analysis Chart of LBD during migration

6.4 Analysis of the Change of AD with LBD During Migration

In the simulation process, In Figure 6, the influence of the average time delay between the switch and the controller on the load balance of the controller is analyzed. The average time delay from the switch to the controller of Q-learning algorithm is 1.471. The average time delay of Bayesian Q-learning algorithm is 0.296, and the average time delay of Bayesian Q-learning algorithm is reduced by 73%. It is easy to see that the Bayesian Q-learning algorithm is much better than the Q-learning algorithm, which can improve the load balancing speed of the controller more quickly on the basis of good load balancing.

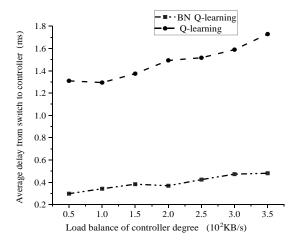


Figure 6 Analysis of the change of AD with LBD during migration

7 Concluding Remarks

With the rapid development of network scale and cloud computing, data center, as the core infrastructure of large-scale server collection, is highlighted, which is mainly reflected in the explosive growth of network traffic. The traditional network architecture has a single static processing method, which is more unsuitable for the data center network traffic scheduling, and seriously hinders the development of cloud computing. SDN network architecture, with its characteristics of decoupling control plane and forwarding plane, centralized network control, standardization of control and forwarding interface, providing programmable interface, provides a good solution for the uniform distribution of network traffic, improving the bandwidth utilization of links, flexibility of traffic scheduling, and becomes an important development direction of future traffic load balancing scheduling.

This paper focuses on load balancing of IOT controller based on SDN architecture of data center. Bayesian network is used to predict the load congestion degree of SDN controller. Based on the prediction results and reinforcement learning, the optimal action strategy is made to balance the load of SDN controller. The simulation results show that the proposed algorithm effectively reduces the average time delay from the switch to the controller and the load balance of the controller, on the basis of well realizing the load balancing function of the controller, the load balancing speed of the controller is improved more quickly, and improves the performance of controller load balancing, improve the security and stability of the network.

References

- Zhou Y, Zheng K, Ni W, et al. Elastic Switch Migration for Control Plane Load Balancing in SDN. IEEE Access, 2018, 6(99): 3909–3919.
- [2] Zhang J, Hu T, Zhao W, et al. DDS: Distributed decision strategy based on switch migration towards sdn control plane// International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, 2017: 486–493.
- [3] Cheng G, Chen H, Hu H, et al. Toward a scalable SDN control mechanism via switch migration. 2017, 14(1): 111–123.
- [4] Lin Y H, Kuo J J, Yang D N, et al. A cost-effective shuffling-based defense against HTTP DDoS attacks with SDN/NFV// IEEE ICC 2017 — International Conference on Communications. IEEE, 2017.
- [5] Yang Z, Yeung K L. Flow monitoring scheme design in SDN. Computer Networks, 2020, 167(53): 129–146.
- [6] Shah N, Giaccone P, Rawat D B, et al. Solutions for adopting software defined network in practice. International Journal of Communication Systems, 2019, 32(17): 101–113.
- [7] Yang M C, Samani H, Zhu K N. Emergency-response locomotion of hexapod robot with heuristic reinforcement learning using Q-learning// Interactive Collaborative Robotics. 2019.
- [8] Min Z, Hua Q, Zhao J. Dynamic switch migration algorithm with Q-learning towards scalable SDN control plane// International Conference on Wireless Communications and Signal Processing (WCSP), October 11–13, 2017, Nanjing, China. Piscataway: IEEE, 2017: 1–4.
- [9] Mu T Y, Ala A F, Khaled S, et al. SDN flow entry management using reinforcement learning. ACM Transactions on Autonomous and Adaptive Systems, 2018, 13(2): 11.
- [10] Da Silva F L, Glatt R, Costa A H R. MOO-MDP: An object-oriented representation for cooperative multiagent reinforcement learning. IEEE Transactions on Cybernetics, 2019(13): 59–73.
- [11] Atsushi Y, Enzo K, Chisato K, et al. Strategies for design of molecular structures with a desired pharmacophore using deep reinforcement learning. Chemical & Pharmaceutical Bulletin, 2020, 68(3): 128–142.
- [12] Rego A, Sendra S, Garcia L, et al. Adapting reinforcement learning for multimedia transmission on SDN. Transactions on Emerging Telecommunications Technologies, 2019, 30(9): 73–91.
- [13] Pan J H, Rao H X, Zhang X L, et al. Application of a Tabu search-based Bayesian network in identifying

- factors related to hypertension. Medicine, 2019, 98(25): 392-403.
- [14] Fenton N, Noguchi T, Neil M. An extension to the noisy-OR function to resolve the 'explaining away' deficiency for practical Bayesian network problems. IEEE Transactions on Knowledge and Data Engineering, 2019, 31(12): 2441–2445.
- [15] Huang H, Guo S, Liang W, et al. Technique report: Near-optimal routing protection for SDN networks using distributed Markov approximation, 2016(10): 103–115.
- [16] Ali S R. Software defined networking (SDN): Using markov models and software reliability engineering// Next Generation and Advanced Network Reliability Analysis, 2019.
- [17] Baz A. Bayesian machine learning algorithm for flow prediction in SDN switches. 2018 1st International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, 2018: 1–7. DOI: 10.1109/CAIS.2018.8441969.
- [18] Wang C, Hu B, Chen S, et al. A switch migration-based decision-making scheme for balancing load in SDN. IEEE Access, 2017, 5: 4537–4544.
- [19] Zhang C C, Wang X W, Dong A W, et al. Dynamic network service deployment across multiple SDN domains. Transactions on Emerging Telecommunications Technologies, 2020, 31(2): 37–51.
- [20] Singh M P, Bhandari A. New-flow based DDoS attacks in SDN: Taxonomy, rationales, and research challenges. Computer Communications, 2020, 154: 509–527.
- [21] Shah N, Giaccone P, Rawat D B, et al. Solutions for adopting software defined network in practice. International Journal of Communication Systems, 2019, 32(17): 52–67.
- [22] Shahryari S, Hosseini-Seno S A, Tashtarian F. An SDN based framework for maximizing throughput and balanced load distribution in a Cloudlet network. Future Generation Computer Systems, 2020, 110: 18–32.
- [23] Kim S, Son J, Talukder A, et al. Congestion prevention mechanism based on Q-leaning for efficient routing in SDN// International Conference on Information NETWORKING. IEEE Computer Society, 2016: 124– 128.
- [24] Min Z, Hua Q, Zhao J. Dynamic switch migration algorithm Q-learning towards scalable SDN control plane. 2017 9th International Conference on Wierless Communications and Signal Processing, 2017: 1–4.
- [25] Omnes N, Bouillon M, Fromentoux G, et al. A programmable and virtualized network & IT infrastructure for the internet of things: How can NFV & SDN help for facing the upcoming challenges// International Conference on Intelligence in Next Generation Networks. IEEE, 2015.
- [26] Szymanski T H. Security and privacy for a green internet of things. IT Professional, 2017, 19(5): 34-41.
- [27] Bizanis N, Kuipers F. SDN and virtualization solutions for the internet of things: A survey. IEEE Access, 2016, 4: 5591–5606. DOI: 10.1109/ACCESS.2016.2607786.
- [28] Lu Y, Ling Z, Zhu S, et al. SDTCP: Towards Datacenter TCP congestion control with SDN for IoT applications. Sensors, 2017, 17(1): 109. DOI: 10.3390/s17010109.
- [29] Kalkan K, Zeadally S. Securing Internet of Things (IoT) with Software Defined Networking (SDN). IEEE Communications Magazine, 2017: 1–7. DOI: 10.1109/MCOM.2017.1700714.
- [30] Wang B, Su J. FlexMonitor: A flexible monitoring framework in SDN. Symmetry, 2018, 10(12): 713–713.
- [31] Chakrabarty S, Engels D W, Thathapudi S. Black SDN for the internet of things// 2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems (MASS). IEEE, 2015.
- [32] Farshin A, Sharifian SA chaotic grey wolf controller allocator for Software Defined Mobile Network (SDMN) for 5th generation of cloud-based cellular systems (5G). Computer Communications, 2017, 108: 94–109.
- [33] Shabbir G, Akram A, Iqbal M M, et al. Network Performance Enhancement of Multi-sink Enabled Low Power Lossy Networks in SDN Based Internet of Things. International Journal of Parallel Programming, 2020, 48: 367–398. https://doi.org/10.1007/s10766-018-0620-8.
- [34] Wang K, Wong W C, Chai T Y. A MANET routing protocol using Q-learning method integrated with Bayesian network. IEEE International Conference on Communication Systems, November 21–23, 2012, Singapore, Singapore. Piscataway: IEEE, 2012: 270–274.