



A reinforcement learning optimization for future smart cities using software defined networking

Kulandaivel Rajkumar¹ · Manikandan Ramachandran¹ · Fadi Al-Turjman² · Rizwan Patan³

Received: 21 July 2020 / Accepted: 19 November 2020 / Published online: 2 January 2021
© Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract

Nowadays smart cities towards software defined network (SDN) approach will become better flexibility and manageability. A stronger, more dynamic network is an SDN network, which is precisely what a smart city network must be if it wants to be viable on a real-world scale. SDN architecture is developed to implement a learning framework for network optimization. The proposed method is called mixed-integer and reinforcement learned network optimization (MI-RLNO) for SDN monitoring. In the first phase, mixed-integer programming formulation is used as an optimization formulation for latency and convergence time. In the second phase, a reinforced Q Learning model is designed that uses communication and computation time as input state vector. Optimization formulation is used as the actions and strategies to be followed during the design and operation of communication networks, therefore contributing fairness and throughput. Simulation results improved the efficiency of the MI-RLNO method.

Keywords Software defined network · Mixed integer · Reinforcement · Q learning · Network optimization

1 Introduction

With the inception of the 5G wireless communication system, a fully connected and well-established society is said to persist, provide different types of business structures, domain expertise, and services, connected with different types of traffic patterns and intense prerequisites. Besides, several network models and different types of user equipment (UE) in turn coordinate with several potentialities

comprising ultra-dense network (UDN) disposals, thus promising the comprehensive future of wireless communications even more demanding.

With the era of the inceptive stage of 5th generation (5G), this type of heterogeneous framework forces the users or devices in an abstract resource pool, shifting the move from the conventional cell notion. Also, exhibit the network administrator with the cooperative resources in terms of time, frequency, and space. A crucial requirement is thus produced, to acquire an outline of the network to acknowledge optimally.

The abstractions of SDN and software defined radio (SDR), also called cross-layer controller (CLC) [1], were an exercise in a harmonized and coordinated and enlightened pattern, therefore contributing to optimize bandwidth utilization. However, fairness during resource allocation was not concentrated. In this work, throughput optimization is addressed by proposing a reinforcement machine learning that stops learning only when the convergence is attained.

SDN is distinguished by the network programmability and 5G control centralization in the controller with which fine-grained network management (NM) is said to be ensured. Intelligent probing (IPro) was utilized in [2] knowledge defined networking (KDN) standard and reinforcement learning. With the KDN-based architecture,

✉ Manikandan Ramachandran
srmanimt75@gmail.com
Kulandaivel Rajkumar
rajkumar@cse.sastra.edu
Fadi Al-Turjman
alturjman@outlook.com
Rizwan Patan
prizwan5@gmail.com

¹ School of Computing, SASTRA Deemed University, Thanjavur, Tamil Nadu, India

² Artificial Intelligence Department, Research Center for AI and IoT, Near East University, Nicosia, Mersin 10, Turkey

³ Department of Computing Science and Engineering, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada 520007, India

probing interval tuning was said to be optimized by also maintaining an admissible monitoring accuracy (MA) and optimal bandwidth. Despite the accuracy and optimal bandwidth achieved, the network optimization factors were not concentrated. In this work, network optimization is arrived at by proposing a mixed integer programming formulation. It uses latency and convergence time as optimization factors.

Motivated by the above works, the proposed MI-RLNO is introduced for SDN monitoring with network optimization. By combining SDN with machine learning technique, a network optimized method is introduced in 5G-SDN. The major idea of the method is to divide the network into the data plane, control plane, and application plane.

The data plane is for data transmission and the control plane controlling the input user equipment. With the introduction of machine learning techniques in the control plane, decision making is optimal via global knowledge of network states. The method achieves efficient network management and optimizes network utilization, thus results in lesser convergence time and higher throughput.

Our contributions are summarized as follows:

- To achieve network-wide optimization, the MI-RLNO method is introduced by monitoring the real dataset of sensors.
- We integrate Mixed Integer Programming and Q Learning to optimize the network in terms of latency, throughput, and convergence time.
- To verify the feasibility of the MI-RLNO method, communication time, and computation time is calculated based on transmission power and bandwidth. Simulations are performed to illustrate convergence time, latency, and throughput.

The remainder of the article is ordered as follows. In Sect. 2, related works are reviewed. The MI-RLNO method is described in Sect. 3 with the aid of diagrams and algorithms. In Sect. 4 simulation setup is provided and a detailed analysis of the discussion is included. The article is concluded in Sect. 5.

2 Literature review

With rising developments in wireless networks, a large volume of data is generated at a faster rate than could be handled by the server. A three-parameter Weibull Cumulative Distribution was presented in [3] through SDN to scrutinize latency in a cyber-physical system. However, with the rapid spectrum shortage faced due to the voluminous increase during wireless communication, optimal spectrum utilization was not ensured.

To address the issue related to volume, Artificial Intelligence-based data analytics was presented in [4] for both feature extraction and dimensionality reduction, therefore contributing to optimal spectrum allocation. But, it failed to consider the spectrum shortage problems. Besides, to ensure secure and optimal resource allocation, a blockchain empowered AI model was proposed in [5] to minimize cumulative average system utility. However, the energy communication cost was higher.

In the current era, with the global vehicles reaching over 1 billion, to guarantee ubiquitous and reliable communications, (5G) enabling technologies have been used in recent years. A 5G enabled SDN was proposed in [6] to ensure scalable networking. But, the convergence time was not minimized. Yet another software-defined space-air ground integrated moving cells contributing integrity and security was presented in [7]. Despite the improvement in scalability and security, transmission interference was not analyzed. A Euclidean planar graph along with an interference relationship graph was designed in [8] therefore improving the average throughput capacity. However, the end-to-end delay was not reduced.

In 5G communication systems, energy and spectrum resources play a significant role in continuous evolution. In [9], a Software-Defined Energy Harvesting Networking (SD-EHN) was designed to contribute to the energy scheduling process. Besides, a stochastic inventory theory was also designed using the Nash bargaining gaming theory to optimize energy utilization and energy saving also. However, for latency-sensitive businesses like the internet of things it is difficult to converge at an optimal rate and time. To address this issue, and Experience Weighted Attraction (EWA) algorithm was presented in [10] resulting in optimal convergence time. But, the transmission delay was not reduced.

With the overall performance of the network being hindered by its lifetime, the internet of things is highly susceptible to energy harvesting. A software-defined energy harvesting internet of things (SEANET) was designed in [11] to enhance the communication speed and lessen energy consumption. Though energy harvesting was attained the power consumption was not found to be optimal. An energy efficiency metric called Ratio for Energy Saving in SDN was proposed in [12] to both improve the link saving and traffic proportionality. But, the supervised and reinforcement machine learning techniques were not considered.

A load-balancing scheme called, genetic programming based load balancing (GPLB) was presented in [13] and significantly big improvement was observed in latency and throughput. However, the performance of the network was not improved. A self adaptive load balancing (SALB) scheme was investigated in [14] that performed the task of load balancing between multiple controllers with the minimum number of packet drops. Though, flexibility is offered

in SDB. However, security remains one of the most complicated metrics to be achieved. In [15] a recurrent neural network based on a new regularization technique was proposed to enhance the network security. But, the throughput rate was not increased.

A Q-learning algorithm based on host weight and vulnerability success rate to reduce both complexity and improve security was presented in [16]. Though propagation latency was concentrated, however, the controller capacity and load switches were overlooked.

A Varna based optimization (VBO) technique was presented in [17] to address issues related to the controller placement problems. But, it failed to analyze the large-sized network's capacity. An up-to-date review of security concerns in SDN was proposed in [18]. However, the method was not discussing the DDoS attack on SDN. Yet another method called, Adaptively Adjusting and Mapping controllers was investigated in [19] to minimize the delay and ensure robustness. However, the communication cost for obtaining flow information was not addressed. An integer linear programming model along with a heuristic technique was formulated in [20] to minimize the cost involved in communication between flows. But, the computational complexity was not minimized.

To cope with the problem, in this paper, we propose MI-RLNO for SDN monitoring that addresses the limitations with low latency, convergence time while ensuring a high throughput.

3 Materials and methods

From the above review of related literature, it is inferred that though the robust model of bringing cognitive structure into networks has been recommended in the past few years. With the shift into the 5G wireless communication systems, the idea has not yet found its way to a large-scale application. With the ever-increasing demand for high data rates and mobility, 5G communications technologies have started revolutionizing the current network.

In this work, a 5G-enabled communication technology that possesses dual advantages, mixed integer programming for latency and converging time optimization, and reinforcement learning for fair resource allocation or throughput improvement is presented.

The integrated framework enhances the overall network by optimizing network-wide objectives like network throughput maximization, reducing convergence time, fairness resource allocation, minimizing latency, or hop count. According to these premises, the MI-RLNO method concentrates on programming models, and in particular MI-RLNO as key enabling factors to network optimization. In the

remainder of this section, the MI-RLNO method is detailed along with a system model and block diagram.

3.1 System model

Consider an SDN system with a number of switches fixed in a designated area. Assume that switches have diverse traffic transmission demands with SDN connected with user equipment (i.e. IoT devices, cloud users, nodes, and so on) and are authorized to select controllers from the distinct set. In this work, the controller placement problem of the network is studied. The number of switches in the SDN system is indicated as ' N ' and ' $S = S_1, S_2, \dots, S_N$ ' is set of switches, where ' S_i ' is the ' i th' switch. To differentiate link status among switches, a duplex identifier ' DID ' is provided, where ' $DID_{ij} = 1$ ', if ' S_i ' is directly linked with ' S_j ' in SDN, otherwise ' $DID_{ij} = 0$ '.

Let ' C_i ' represent the ' i th' controller, then it is assumed that ' C_i ' is combined with ' S_i ' with a maximum of ' M ' controllers. Let ' $\alpha = \alpha_1, \alpha_2, \dots, \alpha_M$ ' represent the controller set capacity, where ' α_M ' denotes the ' M th' capacity that is said to be allocated to one controller. Let us further assume the User Equipment to be ' UE ' (i.e., IoT device, Cloud computing users, mobile devices). In Fig. 1, the proposed 5G-enabled SDN is presented.

As shown in the figure, data plane forward flows in observed SDN by network devices from user equipment ' UE '. The user equipment here includes the IoT devices, cloud users, mobile devices, and so on. Next, the control plane includes various controllers ' C ' that translates the requirements from user equipment ' UE ' to the application plane. Figure 2 shows the block diagram of the MI-RLNO method for 5G-enabled SDN optimization.

The diagram of the MI-RLNO is presented in Fig. 2. The agent is located within a control plane in SDN. It obtains the Key Performance Indicators (KPIs), i.e., the raw and aggregated spectrum data from a list of all sensors that define the

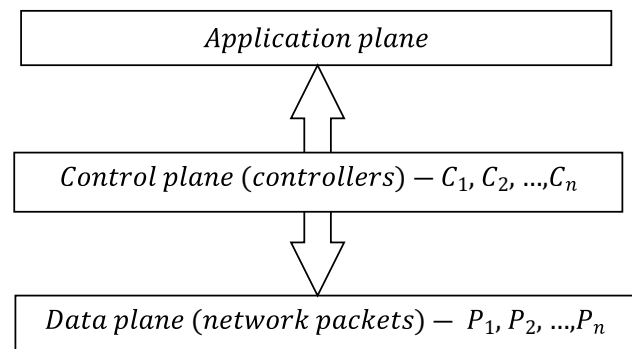
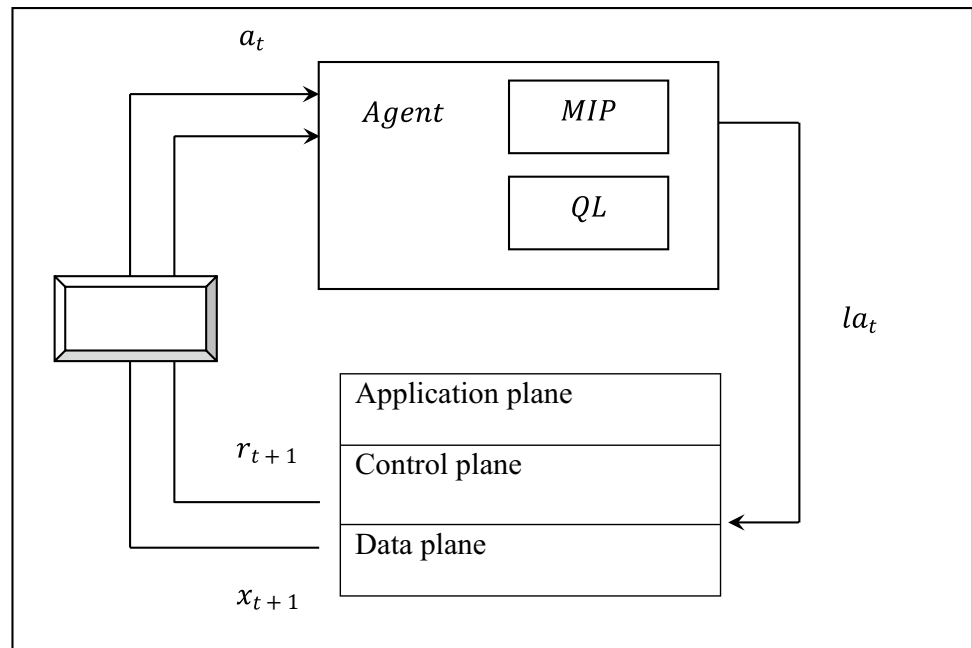


Fig. 1 Conceptual diagram of 5G-enabled SDN

Fig. 2 Block diagram of mixed-integer and reinforcement learned network optimization



system state ‘ x_T ’ at time T, and performs local action ‘ la_T ’. The agent sends to new state ‘ x_{T+1} ’ and attains a reward ‘ r_{T+1} ’. The KPI, in turn, stabilizes the control process in the control plane.

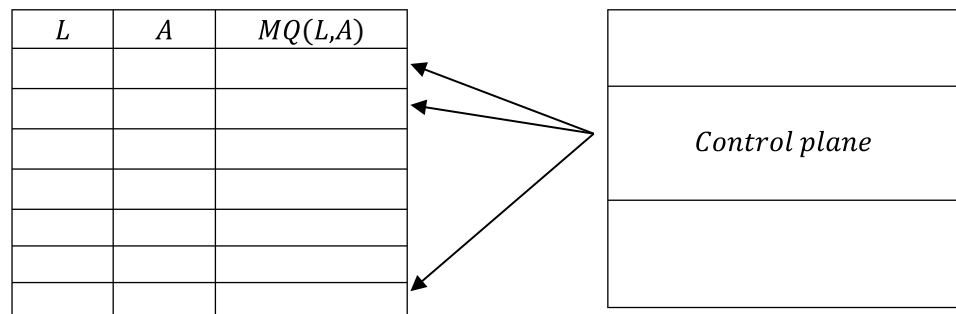
The Q-learning (QL) algorithm studies optimal policy which maps states to control actions. For continuous state space, we propose to use the MI-RLNO which combines Mixed Integer Programming with the QL algorithm. In the MI-RLNO method, the controlled system is presented as a communication and computation model. The cooperative learning model uses global reward includes rewards of each learning agent. In MI-RLNO nomenclature, we employ Labels ‘L’ that indicates discrete states, Actions ‘A’ corresponding to MIP rules. Through learning, the agents apply strategy by feeding the single MQ-table. In addition to a fast convergence time, this model benefits throughput via learning by the cooperating agents. Figure 3 given below shows the MQ-table.

The components of MI-RLNO are state, actions and strategy, and utility function. They are explained in the following sections. The input state vector to the MI-RLNO is defined as follows. The input state vector to the MI-RLNO is obtained from the communication and computation model given as below. Let us design a communication model where the ‘UEs’ communicate with the controllers ‘Cs’ using a one-hop network structure.

As the intervention from the controllers using the same channel with the transmitting device, the Spectrum Efficiency ‘SE’ for the communication link between the ‘jth’ controllers ‘Cs’ and the ‘ith’ User Equipment ‘UEs’ is mathematically formulated as given below.

$$SE_{ij}^C = \text{Log}_2 \left(1 + \frac{t_i c g_{ij}}{\sum t_j c g_{jk} + \delta} \right) \tag{1}$$

Fig. 3 Sample MQ table



From the above Eq. (1), ‘ t_i ’ represents the transmission power of ‘ ith ’ User Equipment ‘ UEs ’, ‘ cg_{ij} ’ and ‘ cg_{jk} ’ represents the channel gain between ‘ UE_iC_j ’ and ‘ UE_jC_k ’ respectively and ‘ δ ’ denotes the noise. In Eq. (1), the spectrum efficiency is the optimized use of spectrum or bandwidth. Then, the higher amount of data is broadcasting with the minimum error.

Then, the data rate ‘ $DR_{C,i}^j$ ’ of ‘ UE_i ’ distributed by ‘ C ’ is mathematically formulated as given below.

$$DR_{C,i}^j = BWSE_{ij}^C \tag{2}$$

From the above Eq. (2), ‘ BW ’, refers to the bandwidth of the available spectrum between ‘ UE ’ and ‘ C ’. ‘ SE_{ij}^C ’ denotes the Spectrum Efficiency. For the controller ‘ C ’ computing model, the ‘ ith ’ UE transfer the computation task ‘ $Task_i$ ’ to the controller ‘ C ’ through a wireless connection between UEs (i.e., IoT users, mobile users, cloud users) and controllers.

Then, the controllers evaluate the task for the ‘ UEs ’. Hence, the computational model for executing the task includes communication time and computation time. The communication time is measured based on the size of input data ‘ D_{size} ’ and the data rate ‘ $DR_{C,i}^j$ ’ of ‘ UE ’ distributed by ‘ C ’ and hence is mathematically formulated as given below.

$$T_{i,comm}^C = \frac{D_{size}}{DR_{C,i}^j} \tag{3}$$

In Eq. (3), ‘ $T_{i,comm}^C$ ’ represents the communication time. Let ‘ $CR_{C,i}$ ’ corresponds to the computation resource of the ‘ C ’ assigned to UE ‘ i ’. Then, the computation time for task ‘ $Task_i$ ’ is mathematically formulated as given below.

$$T_{i,comp}^C = \frac{A_i}{CR_{C,i}^j} \tag{4}$$

Therefore, the total execution time of the task of UE ‘ i ’ distributed by ‘ C ’ is mathematically formulated as given below.

$$T_i^C = T_{i,comm}^C + T_{i,comp}^C \tag{5}$$

From Eq. (5), ‘ T_i^C ’ denotes the total execution time of each task. ‘ $T_{i,comm}^C$ ’ represents the communication model and ‘ $T_{i,comp}^C$ ’ denotes the computation model. Then, the state vector is mathematically formulated as given below.

$$X_C = [SE_{ij}^C, DR_{C,i}^j, T_i^C] \tag{6}$$

From the above Eq. (6), the input state vector ‘ X_C ’ is formalized based on the spectrum efficiency ‘ SE_{ij}^C ’, data rate ‘ $DR_{C,i}^j$ ’ and the total execution time of each task ‘ T_i^C ’ for the corresponding UE. With the obtained state value, the action

is then deduced. The action is the reduced number of hops and convergence time allocated by a controller to the user equipment ‘ UE ’.

The optimization problem (i.e. latency optimization and convergence time optimization) with a mixed integer programming is given below. Our first action is to decrease the latency and the second action is to decrease the convergence time. The MIP model to minimize the latency to the controller is mathematically formulated as given below.

$$\begin{aligned} & \text{MinimizeLat} \\ & \text{Subjectto} f_{ij}^k = 0, \text{ if } Dis_{ij} > P_{max} \end{aligned} \tag{7}$$

$$\sum f_{ij}^k - \sum f_{ji}^k = Dev_i [D_{out} + D_{in}] \tag{8}$$

$$f_{ij}^k = Lat \tag{9}$$

Equation (7) ensures that the distance between user equipment ‘ i ’ and user equipment ‘ j ’ is greater than the ‘ P_{max} ’ at maximum power level. Then, this user equipment communicates with each other. Equation (8) refers to the user equipment flow ‘ f ’ stabilizing restriction and states that for all devices ‘ Dev_i ’, the amount of data flowing is equal to the incoming flow ‘ D_{in} ’ plus the outgoing flow ‘ D_{out} ’. Finally, Eq. (9) is utilized to measure the total number of latencies in SDN. The second model is used to minimize the convergence time (CT). The MIP formulation for convergence time is given as follows.

$$\begin{aligned} & \text{MinimizeCT} \\ & \sum f_{ij}^k - \sum f_{ji}^k = Dev_i n [D_{out} + D_{in}] \end{aligned} \tag{10}$$

Equation (10) is the user equipment flow stabilizing restriction. Each device generates a data packet ‘ D ’ at each round and during the convergence of learning generates ‘ n ’ amount of data packets. The strategy of a controller ‘ C ’, ‘ π_C ’ is a mapping between the state of controller ‘ X_C ’ and the action is the set of possible actions for the controller ‘ A_C ’ is mathematically formulated as given below,

$$\pi_C : X_C \rightarrow A_C \tag{11}$$

With the above-stated optimization problem for latency and convergence time, finally, the utility is derived. The controller of user equipment optimizing utility function described through the sum of discounted rewards. The optimization problem is expressed as below.

$$\max_{\pi_C \in \pi_p} : R_C = \left[\sum_{t=1}^n \beta^t r_C(X_{CT}, A_{CT}) \right] \tag{12}$$

From Eq. (12) ‘ π_p ’ is a set of allowable policies for controller ‘ C ’, ‘ $r_C(X_{CT}, A_{CT})$ ’ is an instantaneous reward seen

by controller ‘ C ’ in-state ‘ X_{CT} ’ when taking action ‘ A_{CT} ’ at a time ‘ T ’ and ‘ β ’ refers to a cut-off factor varying between ‘ $[0,1]$ ’. When the cut-off factor ‘ β ’ is lesser, the significance given by the controller provides the rewards concerning future ones.

Finally, with throughput optimization, the solution to the maximization problem given in (12) utilizes the action-value function under the policy ‘ π ’. It is measured as the sum of

discounted rewards when starting from state ‘ $X_0 = X$ ’ at ‘ T_0 ’ and formulated as below.

$$Q^\pi(X, A) = \left[\sum \alpha r(X_t, A_t) | X_0 = X, A_0 = A \right] \quad (13)$$

The pseudo-code representation of mixed integer reinforcement learning is given below.

Algorithm 1: Mixed Integer Reinforcement Learning

Input: Controller ‘ $C = C_1, C_2, \dots, C_n$ ’, User equipment ‘ $UE = UE_1, UE_2, \dots, UE_n$ ’, size of input data ‘ D_{size} ’

Output: network-level optimization

- 1: **Initialize** transmission power ‘ t_i ’, channel gain ‘ $UE_i C_j$ ’ and ‘ $UE_j C_k$ ’
 - 2: **Begin**
 - 3: **For** each Controller ‘ C ’ with User equipment ‘ UE ’
 - 4: Measure spectrum efficiency using (1)
 - 5: Measure data rate using (2)
 - //input state vector
 - 6: Evaluate communication time and computation resource using (3) and (4)
 - 7: Obtain state vector using (6)
 - //Action and Strategy using MIP – Optimization problem
 - 8: Formulate minimization of latency using (7), (8) and (9)
 - 9: Formulate minimization of convergence time using (10)
 - 10: Evaluate strategy using (11)
 - //Utility function
 - 11: Evaluate utility function optimization using (12)
 - 12: Obtain solution to maximization problem using (13)
 - 13: **If** ‘ $(X^1 \text{ is } L_j^1) \dots$ ’ and ‘ $(X^n \text{ is } L_j^n)$ ’
 - 14: **Then** ‘ $A = OA_j$ ’ with ‘ $Q(L_j, OA_j)$ ’
 - 15: **Endif**
 - 16: **If** convergence is attained
 - 17: **Then** stop learning
 - 18: **End if**
 - 19: **End**
 - 20: **End**
-

From the above mixed integer reinforcement learning algorithm, the objective of the work remains in network type optimization in terms of fairness, latency, convergence time. In this work, an integrated mixed integer programming function with the reinforcement Q learning is applied to attain the network optimization. First, the input state vector to the MI-RLNO is attained from the communication and computation model. The communication and computation model is designed along with the data rate for SDN to address the QoS requirement. Next, mixed integer programming is used to select the action and strategies for reducing the latency and convergence time. Next, the utility function is described by sum of discounted rewards. Finally, SDN with its action-value function can achieve better network utilization via network-wide optimizations.

4 Results and validation

In this section, the results of the MI-RLNO method is compared with Cross-layer control based on SDN and SDR [1] and IPro [2] for SDN monitoring. In this section, experiments are conducted with data retrieved for a sensor within a specified time window extracted from <https://electrosense.org/api-spec>. Similarly, the web API permits for retrieval of raw and aggregated spectrum data for a list of different sensors. Performance measures are made in terms of latency, convergence time, and throughput rate.

Experiments are conducted in python for network-wide optimization and comparative analysis is made with the proposed MI-RLNO for SDN monitoring with the existing, Cross-layer control based on SDN and SDR [1] and IPro [2]. In the simulation, SDN scenarios include several SDN switches which are arbitrarily positioned in 2000 km × 2000 km square region. In other words, the switches position follows uniform arbitrary distribution in the simulation region. Besides, it is assumed that a random connection between any two switches exists in the region. The link transmission rate is selected randomly from a set of values. The three-link characteristics are provided in Table 1.

For simulation, the data packet size in ‘ D_{in} ’ is set as 160 bytes and the arrival rate of packet message requests from

Table 1 Link transmission rate

Link cases	Link transmission rate (Mbps)
Case 1	50, 100, 150, 200, 300, 500
Case 2	50, 75, 100
Case 3	150, 200, 300

Table 2 Data packet request

Scenarios	$\lambda \in [1k, 10k]$	$\lambda \in [10k, 100k]$	$\lambda \in [100k, 200k]$
Scenario 1	100%Switches	0	0
Scenario 2	60%Switches	20%Switches	20%Switches
Scenario 3	70%Switches	20%Switches	10%Switches

user equipment is indicated as ‘ λ ’. To observe the difference of ‘ λ ’ three message request scenarios from user equipment is presented as simulation. Table 2 lists the simulation parameters. Simulation results are averaged over 500 independent processes and instances (Figs. 4, 5, 6, 7, 8).

4.1 Scenario 1: convergence time

Convergence time is calculated as how fast a group of user equipment (provided with data as message request) to the controllers reaches a state of convergence. It is a significant performance indicator for network-wide optimization that runs the protocol to rapidly and reliably converge. It is measured as given below.

$$CT = \sum_{i=1}^n C_i * Time [Dev_i, n [D_{out} + D_{in}]] \quad (14)$$

From Eq. (14), the convergence time ‘ CT ’ is calculated depending on the number of controllers ‘ C_i ’ and time consumed for ‘ n ’ number of packets. It is measured in milliseconds (ms). Table 3 given below demonstrates the converging time tabulation results for the MI-RLNO method, cross-layer control based on SDN and SDR [1], and IPro [2].

Figure 9 shows the relationship between the convergence time concerning 20 different controllers with three different methods. From the figure, it is inferred that for case 1 with 2 controllers, the convergence time decreases. However, with controllers in the network reaching a value (i.e. 20), the convergence time starts to increase when the number of controllers increases. This is due to the reason that the increase in the controller results in the increase in the inter-controller delay and therefore results in an increase in the convergence time. However, from the figure, the proposed method outperforms than the existing method [1, 2]. The reason is that the algorithm in [1, 2] mainly enhances the monitoring accuracy among controllers and switches and ignores the convergence time among controllers and switches. Also, the simulation results show that the convergence time with 2 controllers offers the best latency, while convergence time with 20 controllers offers the worst latency. This is because the service request of switches for 2 controllers is low as compared to other higher numbers of controllers.

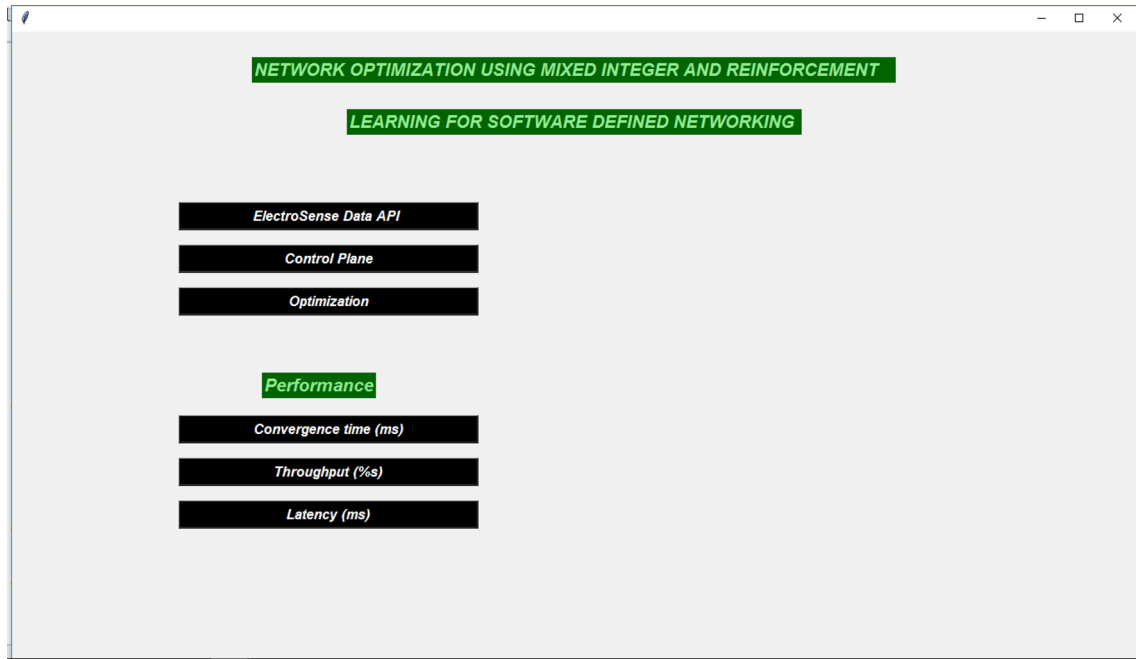


Fig. 4 Construction of electro sense Data API

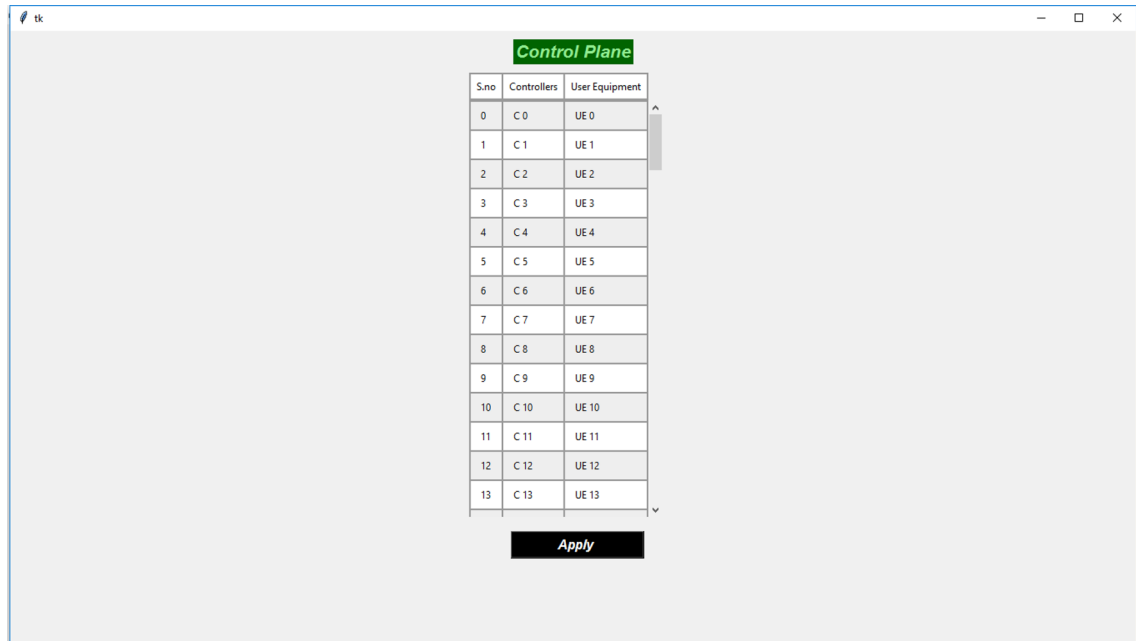


Fig. 5 Apply control plane

S.no	Controllers	User Equipment	spectrum efficiency	data rate
0	C-0	UE 0	88.0	80
1	C-1	UE 1	94.0	72
2	C-2	UE 2	33.0	68
3	C-3	UE 3	96.0	35
4	C-4	UE 4	63.0	84
5	C-5	UE 5	39.0	100
6	C-6	UE 6	59.0	22
7	C-7	UE 7	84.0	93
8	C-8	UE 8	69.0	30
9	C-9	UE 9	98.0	31
10	C-10	UE 10	63.0	86
11	C-11	UE 11	68.0	32
12	C-12	UE 12	29.0	30
13	C-13	UE 13	98.0	22

Fig. 6 Output of control plane

S.no	Controllers	Time (t)	cut off factor
0	C 0	12.0	0
1	C 1	17.0	1
2	C 2	20.0	0
3	C 3	19.0	0
4	C 4	4.0	1
5	C 5	20.0	0
6	C 6	13.0	0
7	C 7	19.0	0
8	C 8	10.0	0
9	C 9	17.0	0
10	C 10	10.0	1
11	C 11	3.0	0
12	C 12	12.0	0
13	C 13	9.0	0

Apply

Fig. 7 Apply optimization

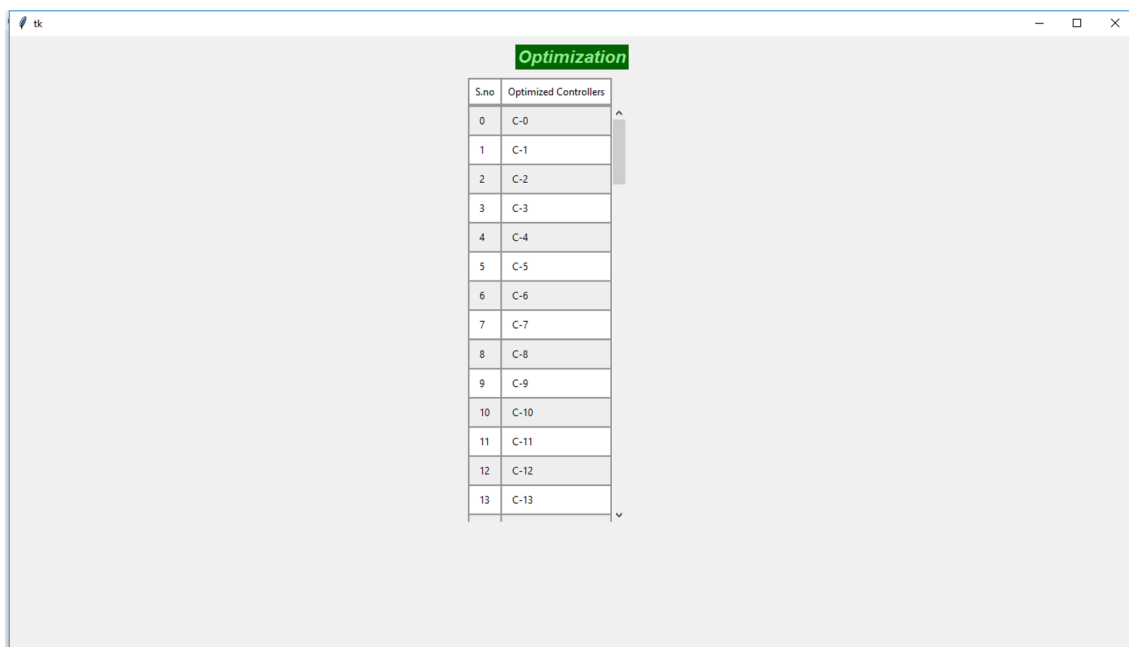


Fig. 8 Output of optimization

- As a result, the proposed MI-RLNO method reduces the convergence time by 27% as compared to [1] and 22% as compared to [2].

4.2 Scenario 2: throughput

Throughput refers to the percentage ratio of successfully received data packets by the controller to the total data packets being sent from the user equipment 'UE'. The throughput is measured as given below.

$$T_{put} = \frac{D_C}{D_{UE}} * 100 \tag{15}$$

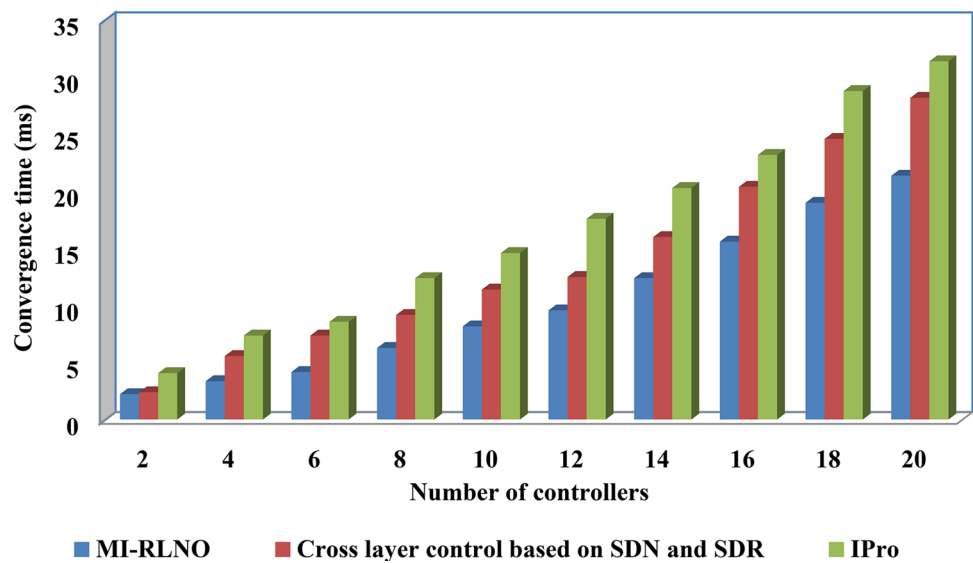
From the above Eq. (15), throughput 'Tput' is measured based on the 'DC' and 'DUE'. It is measured in terms of percentage (%). Table 4 given below demonstrates the throughput tabulation results for the MI-RLNO method, cross-layer control based on SDN and SDR [1], and IPro [2].

Figure 10 given above illustrates the relationship between the throughput rate and data packets in the range of 15–150 sent by the different user equipment to the controller at different time intervals. Ten different simulation runs were conducted. With data packet requests for scenario 2 considered as simulation, to start with an increase in the number of data packets causes a decrease in throughput and at the fifth simulation, runs start increasing and so on. However, simulations conducted with 15 data packets, throughput were observed to be 86.66% using the MI-RLNO method, 80% using [1], and 73.33% using [2]. From this simulation, it is inferred that fairness during the resource allocation is said to be improved by applying the MI-RLNO method and therefore increasing the overall system throughput. This is because of the application of Mixed Integer Programming as an input state in the Mixed Integer Reinforcement Learning algorithm. Thus in turn based on the communication and computation model performs message request allocation to the corresponding user equipment.

Table 3 Convergence time results

Number of controllers	Convergence time (ms)		
	MI-RLNO	Cross layer control based on SDN and SDR	IPro
2	2.21	2.37	4.05
4	3.35	5.55	7.35
6	4.15	7.35	8.55
8	6.26	9.15	12.35
10	8.15	11.35	14.55
12	9.55	12.45	17.55
14	12.35	15.95	20.25
16	15.55	20.35	23.15
18	18.95	24.55	28.75
20	21.32	28.15	31.35

Fig. 9 Convergence time



- Thus, the throughput of the MI-RLNO method is improved by 7% when compared to [1] and 17% compared to [2].

4.3 Scenario 3: latency

Latency in terms of network optimization refers to the time taken for a request (i.e. data packet request) from the sender (i.e. user equipment) to the controller and for the controller to process that request. In other words, it is also referred to as the round trip time from the user equipment to the controller in SDN.

$$L = \sum_{i=1}^n C_i * Time(D_{UE} - D_C - D_{UE}) \tag{16}$$

Table 4 Throughput results

Data packets (user equipments)	Throughput (%s)		
	MI-RLNO	Cross layer control based on SDN and SDR	IPro
15	86.66	80	73.33
30	83.25	78.25	72.15
45	81.45	77.15	70.35
60	80.25	75.35	68.15
75	79.15	71.25	70.25
90	77.35	78.35	72.35
105	79.45	77.45	68.45
120	80.25	75.35	65.15
135	82.35	72.15	68.35
150	80.45	70.25	65.55

From the above Eq. (16), latency ‘L’ is measured based on the round trip time ‘(D_{UE} - D_C - D_{UE})’ and the number of controllers ‘C_i’. It is measured in terms of milliseconds (ms). Table 5 given below demonstrates the throughput tabulation results for the MI-RLNO method, cross-layer control based on SDN and SDR [1], and IPro [2].

Figure 11 given above demonstrates the relationship between the number of controllers and latency. The performance of latency is analyzed through the dataset with two existing methods. The same input is used to analyze the latency for proposed and existing methods. From that, the proposed work reduces the latency than the existing [1, 2]. From the figure, it is inferred that for case 1 with 2 controllers, the convergence time decreases. Latency in our work is measured based on the round-trip time. From the figure, it is inferred that increasing the number of controllers, the time consumed between the sender and receiver increases and therefore the latency also increases. But with the simulations conducted for 2 controllers, the latency was observed to be 0.25 ms using the MI-RLNO method, 0.33 ms using [1] and 0.41 ms using [2]. The latency improvement in the proposed method over [1, 2] is due to the application of the Mixed Integer Reinforcement Learning algorithm. By applying this algorithm, first, communication and computation model is used along with the data rate as input, therefore improving the bandwidth utilization. Next, utilizing MIP, actions, and strategies based on the optimization factors are used to form strategies. Finally, with the discounted rewards, the allocation is made by the controller.

- Therefore, the latency of the MI-RLNO method is reduced by 12% as compared to [1] and 26% as compared to [2].

Fig. 10 Throughput

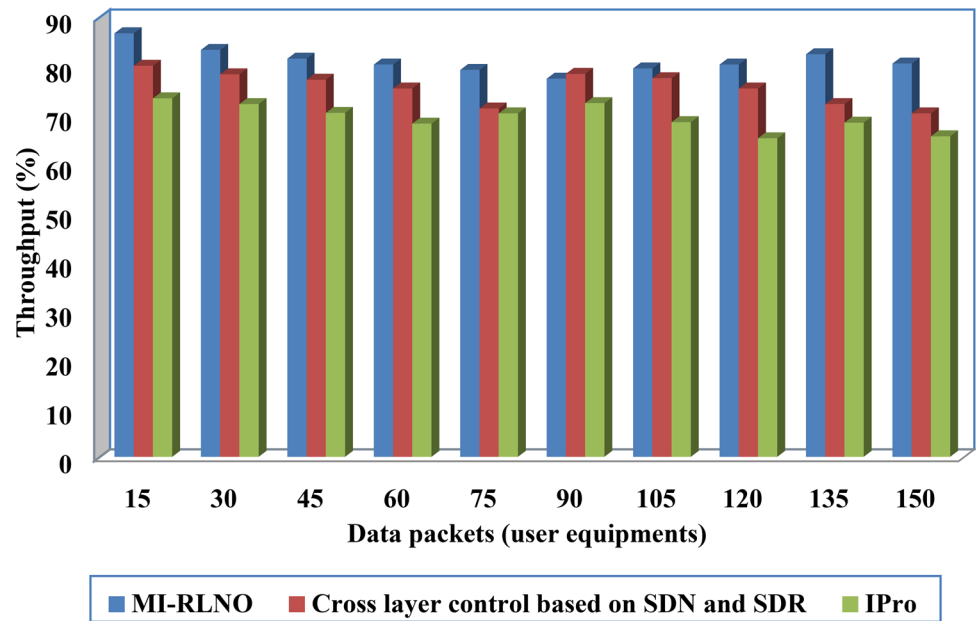


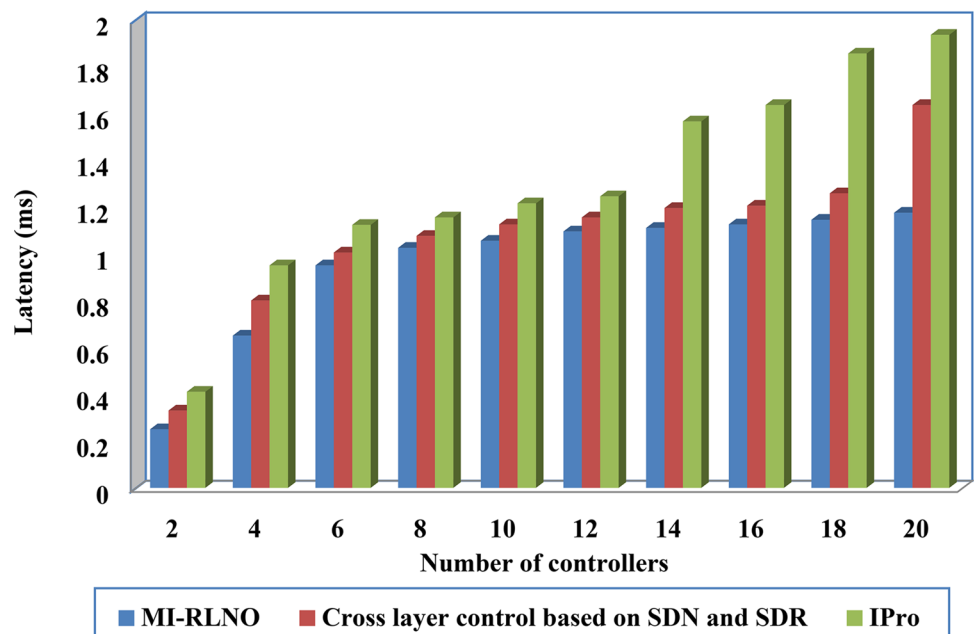
Table 5 Latency results

Number of controllers	Latency (ms)		
	MI-RLNO	Cross layer control based on SDN and SDR	IPro
2	0.25	0.33	0.41
4	0.65	0.8	0.95
6	0.95	1.005	1.123
8	1.025	1.076	1.155
10	1.055	1.125	1.215
12	1.095	1.155	1.245
14	1.11	1.195	1.565
16	1.125	1.205	1.635
18	1.145	1.258	1.855
20	1.175	1.635	1.935

5 Conclusions

MI-RLNO method is proposed to improve the throughput of SDN and optimize system utility (i.e. latency and convergence time). The controller placement mode of SDN is proposed in this algorithm and diverse traffic transmission demands are scrutinized based on the transmission process. The algorithm using Q Learning selects the user equipment to be allocated in the network with the required request based on the communication and computation time. Next, Mixed Integer Programming is used as a means of selecting the action and strategy according to the optimization factors. Finally, an optimal strategy is learned via the sum of discounted rewards to maximize the utility benefits. From the simulation results, the MI-RLNO method provides enhanced performance than the others.

Fig. 11 Latency



References

- Barmounakis S, Maroulis N, Papadakis M, Tsiatsios G, Soukaras D, Alonistioti N (2019) Network slicing-enabled RAN management for 5G: cross-layer control based on SDN and SDR. *Comput Netw* 166:106987
- Castillo EF, Granville LZ, Ordonez A, Rendon OMC (2020) IPro: an Approach for intelligent SDN monitoring. *Comput Netw* 170:107108
- Gifty R, Bharathi R (2020) Weibull cumulative distribution based real-time response and performance capacity modeling of cyber-physical systems through software defined networking. *Comput Commun* 150:235–244
- Lin K, Li C, Tian D, Ghoneim A, Hossain MS, Amin SU (2019) Artificial-intelligence-based data analytics for cognitive communication in heterogeneous wireless networks. *Artif Intell Cogn Wirel Commun* IEEE Wirel Commun 26:83–89
- Dai Y, Xu D, Maharjan S, Chen Z, He Q, Zhang Y (2019) Blockchain and deep reinforcement learning empowered intelligent 5G beyond. *IEEE Netw* 33:10–17
- Huang X, Yu R, Kang J, He Y, Zhang Y (2017) Exploring mobile edge computing for 5G-enabled software defined vehicular networks, emerging technology for 5G enabled vehicular networks. *IEEE Wirel Commun* 24:55–63
- Zhou Z, Feng J, Zhang C, Chang Z, Zhang Y, Huq KMS (2018) SAGECELL: software-defined space-air-ground integrated moving cells, orchestration of ultra-dense 5G networks. *IEEE Commun Mag* 56:92–99
- Guan I, Huang Y, Chen M, Wu H, Ohtsuki T, Zhang Y (2017) Exploiting interference for capacity improvement in software-defined vehicular networks. *IEEE Access* 5:10662–10673
- Huang X, Yu R, Kang J, Gao Y, Maharjan S, Gjessing S, Zhang Y (2017) Software defined energy harvesting networking for 5g green communications. *IEEE Wirel Commun* 24:38–45
- Li Q, Yao H, Mai T, Jiang C, Zhang Y (2015) Reinforcement and belief learning-based double auction mechanism for edge computing resource allocation. *IEEE Internet Things J* 7(7):5976–5985
- Huang X, Yu R, Kang J, Xia Z, Zhang Y (2018) Software defined networking for energy harvesting internet of things. *IEEE Internet Things J* 5:1389–1399
- Assefa BG, Özkasap Ö (2019) Resdn: A novel metric and method for energy efficient routing in software defined networks. *IEEE Trans Netw Serv Manag* 17(2):736–749
- Jamali S, Badirzadeh A, Siapoush MS (2019) On the use of the genetic programming for balanced load distribution in software-defined networks. *Digit Commun Netw* 5:288–296
- Priyadarsini MK, Mukherjee JC, Bera P, Kumar S, Jakaria AHM, Rahman MA (2019) An adaptive load balancing scheme for software-defined network controllers. *Comput Netw* 164:106918
- Albahar MA (2019) Recurrent neural network model based on a new regularization technique for real-time intrusion detection in SDN environments. *Secur Commun Netw* 2019:1–9
- Chen Y, Lv K, Hu C (2018) A dynamic hidden forwarding path planning method based on improved Q-learning in SDN environments. *Secur Commun Netw* 2018:1–12
- Singh AK, Maurya S, Srivastava S (2019) Varna-based optimization: a novel method for capacitated controller placement problem in SDN. *Front Comput Sci* 14:143402
- Sahoo KS, Panda SK, Sahoo S, Sahoo B, Dash R (2019) Toward secure software-defined networks against distributed denial of service attack. *J Supercomput* 75:4829–4874
- Liu W, Wang Y, Zhang J, Liao H, Liang Z, Liu X (2019) AAM-con: an adaptively distributed SDN controller in datacenter networks. *Front Comput Sci* 14:146–161
- Chen H, Li L, Ren J, Wang Y, Zhao Y, Wang X, Wang S, Xu S (2015) A Scheme to optimize flow routing and polling switch selection of software defined networks. *PLoS ONE*. <https://doi.org/10.1371/journal.pone.0145437>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.