Intelligent Routing Control for MANET Based on Reinforcement Learning

Fang Dong[a], Ou Li and Min Tong

National Digital Switching System Engineering & Technology Research Centre, Zhengzhou 450000, China

Abstract. With the rapid development and wide use of MANET, the quality of service for various businesses is much higher than before. Aiming at the adaptive routing control with multiple parameters for universal scenes, we propose an intelligent routing control algorithm for MANET based on reinforcement learning, which can constantly optimize the node selection strategy through the interaction with the environment and converge to the optimal transmission paths gradually. There is no need to update the network state frequently, which can save the cost of routing maintenance while improving the transmission performance. Simulation results show that, compared with other algorithms, the proposed approach can choose appropriate paths under constraint conditions, and can obtain better optimization objective.

1 Introduction

With the rapid development and continuous improvement, the Mobile Ad Hoc Networks (MANET) has been developing from simple information transmission to comprehensive service for various network businesses. Such complex network systems confront a myriad of challenges including management, maintenance and network traffic optimization. New types of businesses have put forward higher requirement on Quality of Service (QoS) [1]. However, traditional MANET just provides the best effort routing mechanism, which cannot guarantee the customized service quality for different network businesses. Therefore, the design of routing control algorithms for service quality guarantee is always one of the research emphases of networks.

Different from traditional algorithms, routing control algorithm for service quality guarantee needs to find feasible paths under multiple constraint conditions and choose a path with optimal parameters. The parameters can be divided into three types. The first one is concavity parameter, which can restrict any link on the path, such as the minimum bandwidth. The second one is additive parameter, which restricts the sum of parameters of all the links on the path, such as the maximum transmission delay. The last one is multiplicative parameter, which restricts the product of parameters of all the links on the path, such as the maximum packet loss rate. Because of the different types of parameters, the problem of routing control with multiple parameters is usually NP-hard [2].

Over the past few decades, machine learning (ML) has been exploited to intelligently dictate traffic control in wired/wireless networks. Reinforcement learning (RL) [3] is a ML technique that attempts to learn about the optimal action with respect to the dynamic operating environment. The agent observes state and reward from the operating environment and takes the action which leads to the optimal performance as time goes by. For each state-action pair, the agent keeps track of its quality, which accumulates the rewards for the action taken under the state and selects an optimal action in order to optimize the performance. Reinforcement learning does not need a model of the operating environment, which means that an agent can learn and make optimal decisions without prior knowledge. Therefore, it is appropriate to use reinforcement learning to adjust network parameters according to the environment and network service requirements and constraints.

In this paper, we propose an intelligent network routing control algorithm based on reinforcement learning (RL-INRC). Firstly, we describe the objective of intelligent network routing control as an optimal solution to a multi-objective and multi-restriction problem. Then we introduce the theorem of Group dynamics and propose a multi-hop cooperative structure for routing control to formulate the relationship between nodes. Finally, the reinforcement learning algorithm is running on each node to explore the possible solutions and find the optimal one without prior knowledge. With the aid of reinforcement learning, RL-INRC can obtain optimal routings through the examination of action policy, quality function and reward function. RL-INRC has the three crucial features

1)RL-INRC can make each node to learn its local policy through experiences and rewards from neighbours and achieve the global optimum objective, which is composed of different types of parameters.

2)RL-INRC has fast adaptation to the current traffic states and constraint conditions for the time-varying environment as well as network topology.

* Corresponding author: [a]chxachxa@163.com

3)RL-INRC can support customized requirements because of the tenable parameters and generic design of the reward function.

The rest of the paper is organized as follows. Related works are given in Section 2. Section 3 introduces the system model and Section 4 presents the proposed intelligent network routing control algorithm. Section 5 presents the performance evaluation and Section 6 summarizes the paper.

2 Related Works

There are already some works on the adaptive routing control with multiple parameters. References [4] extended the Bellman-Ford algorithm for QoS routing problem with multiple constraint conditions. Each node in the network needed to maintain the routing information between source node and itself, share the information with neighbour nodes, and update its routing periodically. However, because of the multiple constraint conditions, the computational complexity was too high to be applied to large networks. References [5] deduced the relationship between multiple parameters in some specific networks, so that the optimization objective was transformed into a polynomial with only one parameter. For example, if Weighted Fair Queuing was used, transmission delay, packet loss rate and queue length were functions of bandwidth. References [6, 7] tried to discretize the parameters into finite values to reduce the complexity of searching paths, so the problem could be easily solved by comparing and selecting the best combination of parameters from finite values. However, the algorithm could not avoid losing information when dispersing serial data, so it was not able to obtain the optimal result. References [8, 9] optimized the OSPF weight setting for multiple parameters. The parameters of each link were weighted combined as the cost of link, and the path with minimum cost was calculated by Dijkstra method, but the algorithm could calculate the optimal path only if all of the parameters were additive.

In recent years, artificial intelligence has been introduced to solve the problem of routing control with multiple parameters. In references [10, 11], multi-objective genetic algorithm was used to choose appropriate paths under constraint conditions. References [12, 13] used ant colony algorithm as a strategy of heuristic search to find optimal routings. Simulated annealing algorithm was also used in [14]. However, the algorithms above needed a lot of invalid search with low efficiency and often converged to locally optimal solutions. Some researches adopted machine learning to manage the paths intelligently. References [15] proposed a load-aware multicast routing algorithm based on neural networks. References [16, 17] presented a preliminary traffic control system facilitated by deep learning-based routing. The supervised approaches performed better than traditional algorithms, but the processing of a lot of heterogeneous traffic was computationally expensive and prone to errors due to the imbalanced nature of the input data and the potential for overfitting. References [18] computed the state transition probability, which was the probability of a transition from one state to another when an action was undertaken, and then used model-based reinforcement learning to select a next-hop node for packet transmission to extend the lifetime of network. A reinforcement learning technique was also used in references [19] to autonomously realize an efficient, adaptive and QoS-provisioning routing in multi-layer hierarchical software defined networks. Machine learning could autonomously self-organize the network and implement intelligent routing, but the research was still at a preliminary stage.

3 System Model of Intelligent Network Routing Control

We model the network as an edge-weighted graph $G=(V,E)$, in which d_e, l_e and b_e ($e\in E$) are the transmission delay, packet loss rate and available bandwidth of link *e*, respectively. The routing requirement is $r(o,t,D,L,B)$, in which *o* and *t* are the source and destination node, while D, L and B are the longest transmission delay, the maximum packet loss rate and the minimum bandwidth of the appropriate path respectively. The fundamental objective of intelligent network routing control is to find a path which can reduce the transmission delay and packet loss rate, balance the bandwidth of network and satisfy the constraint conditions simultaneously. This objective can be mathematically described as

$$\text{Minimize } \phi_1\sum_{i\in P} d_i+\phi_2\left(1-\prod_{i\in P}(1-l_i)\right)+\phi_3 \operatorname{var}(band) \tag{1}$$

Subject to

$$\sum_{i\in P} d_i \le D \tag{2}$$

$$1-\prod_{i\in P}(1-l_i)\le L \tag{3}$$

$$b_i \ge B, \forall i\in P \tag{4}$$

where *P* is the final transmission path and *Band* is the set of final bandwidth of all the links in the network. ϕ_1 , ϕ_2 and ϕ_3 in Eq. (1) are the factors introduced to adjust the importance of transmission delay, packet loss rate and bandwidth balancing. Eqs. (2)-(4) ensure that the path can satisfy the constraint conditions of *r*.

It is not easy to solve the problem above in a distributed network because each node can only optimize its local routing, which is usually not a global optimum solution. Group dynamics shows that, if an agent can keep communication with enough neighbours and adjust its action according to the information shared by neighbours, the group can achieve the global optimum result. That theorem is also effective in distributed routing problems, so we should build a structure to formulate the relationship between nodes and find a mechanism to adjust the local routing of each node to optimize the global objective in Eq. (1).

Therefore, we propose a multi-hop cooperative structure for routing control, in which nodes are divided into several sets along the path with the minimum hops between source and destination. Each node can communicate directly with any node in its set, the previous set and the next set. The mechanism is implemented using reinforcement learning, which is widely used in optimization. Nodes can learn the optimal policy through experiences and rewards from neighbours without the need of prior knowledge of network states.

Figure 1 shows an example of multi-hop cooperative structure. The data packets need to be forwarded from source o to destination t via a multi-hop routing. To establish the number of cooperative node sets, we generate a path with the minimum hops firstly. The relay nodes on the path are called as reference nodes ($V_1, V_2 \ldots V_n$). Then we choose a set of nodes around each reference node as a cooperative node set ($C_1, C_2 \ldots C_n$) (reference node is also in the set). Each cooperative node can communicate directly with any node in its set, the previous set and next set. Once a node receives a data packet, it chooses a node in the next cooperative node set and transmit the packet to the node through reinforcement learning algorithm.

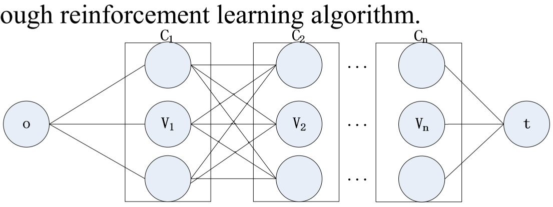


Figure 1. Multi-hop cooperative structure for routing control.

4 Intelligent Network Routing Control Based on Reinforcement Learning

4.1. Action Selection Policy

In the system model of intelligent network routing control, S means the current value of parameters including transmission delay and packet loss rate when a node receives a data packet, and A means all of the nodes in the next cooperative node set. Action policy is the decision rules that will be taken by each node. The design challenge comes from the balance between exploration and exploitation, because the node should exploit the past actions with great quality to maximize the cumulative reward and explore the system for better unknown actions at the same time. There are three policies widely used in related literature, which are the greedy policy, ε-greedy policy and softmax policy. For the greedy policy, action with the highest quality is always selected, which means the agent will not explore unknown actions for possibly higher quality and offen converge to locally optimal solution. Moreover, ε-greedy policy can balance the trade off between exploitation and exploration. Agent follows greedy policy with probability 1-ε and takes a random action with probability ε. However, it may select an action whose quality is very bad and make this exploration meaningless.

Towards this, we use softmax policy to assign the probability of action selection. The probability $\pi_t(s_t, a_i)$ of selecting action a_i with state s_t follows

$$\pi_t(s_t, a_i) = \frac{\exp\left(Q_t(s_t, a_i)/\tau_m\right)}{\sum_i^n \exp\left(Q_t(s_t, a_i)/\tau_m\right)} \tag{5}$$

where n is the number of possible actions, $Q_t(s_t, a_i)$ denotes the quality function of selecting action a_i with state s_t, and τ_m is a parameter called temperature. Temperature controls the trade-off between exploration and exploitation, because the probabilities of all actions tend to be equal when τ_m is high and the probability of action with the maximum quality is nearly to 1 when τ_m is low. Temperature should be annealed over the training phase with better exploration in the beginning and better exploitation near the end. To achieve the learning convergence in finite time, temperature can be set as a linear function over training phase as

$$\tau_m = \tau_0 - \frac{(\tau_0 - \tau_T)m}{T} \tag{6}$$

where T denotes the total training times, m is the current training time, τ_0 is the initial temperature and τ_T is the final temperature.

4.2. Reward Function Design

Reward function represents the quality of the current action decision, which should be designed according to the fundamental objective of intelligent network routing control. There are three parameters needing to be optimized, in which d transmission elay and packet loss rate are cumulative along the path and bandwidth balancing can be observed only when the packet is transmited to destination. Therefore, we design three kinds of rewards when a node makes the decision. If data packet is received by the node in the next cooperative node set (state 1), a reward calculated according to transmission delay and packet loss rate is fed back. If data packet is received by destination (state 2), a reward calculated according to transmission delay, packet loss rate and the variance of bandwidth is fed back. If data packet forwarding fails because of the constraint conditions (state 3), a constant negative reward is fed back. The reward function of transmission from node i to node j can be described as

$$R = \begin{cases} 1/\phi_1\left(d_i + d_{ij}\right) + 1/\phi_2\left(1-(1-l_i)\times\left(1-l_{ij}\right)\right) & \text{state 1} \\ 1/\phi_1\left(d_i + d_{ij}\right) + 1/\phi_2\left(1-(1-l_i)\times\left(1-l_{ij}\right)\right) + 1/\phi_3 \operatorname{var}(band) & \text{state 2} \\ -100 & \text{state 3} \end{cases} \tag{7}$$

where d_i and l_i are transmission delay and packet loss rate when the packet is forwarded to node i , while d_{ij} and l_{ij} are transmission delay and packet loss rate between node i and node j.

4.3. Quality Function

The quality function $Q_t(s_t, a_i)$ shows the quality for selecting action a_i at state s_t and should be stored in each node as a table entry. However, transmission delay and packet loss rate are continuous variables, which makes the number of states infinite. It is necessary to discretize the possible values of transmission delay and packet loss rate when the node is ready to forward data packet. For example, if the range of transmission delay is from 5ms to 10ms, the possible values can be discretized as 5.5ms, 6.5ms, 7.5ms, 8.5ms and 9.5ms. The number of possible values influences the precision of our algorithm and the storage complexity. In this paper, we set the number of possible values of each parameter as 5, which makes the storage complexity sufficient for the problem domain and leads to a little lower precision. Q function approximation algorithms such as deep reinforcement learning may be an appropriate approach to optimize the storage complexity in the future researches.

The initial values of Q are zero, and they are updated accoring to current reward and long-term revenue until they converge to stable values for optimal solution. The well-known Q-learning algorithm updates the quality function as follows

$$Q_{t+1}(s_t,a_i)=(1-\alpha)Q_t(s_t,a_i)+\alpha\left(R_t+\gamma\max Q_t(s_{t+1},a)\right) \quad (8)$$

where $\gamma\in[0,1)$ is the discount factor which determines the importance of future rewards, and $\alpha\in[0,1)$ is the learning rate which determines the proportion of newly acquired information. Eq. (8) shows that the quality for selecting action a_i at state s_t is a weighted sum of quality for selecting action a_i at state s_t in the previous state, the current reward of action a_i and the maximum quality at future state s_{t+1}.

The intelligent routing control in a distributed network is clearly a cooperative problem and would benefit from a team-based learning approach, so we update the quality function with a multi-agent Q-learning algorithm as follows

$$Q_{t+1}(s_t,a_i)=(1-\alpha)Q_t(s_t,a_i)+\alpha\left(R_t+\gamma\max Q_t^{k+1}(s_{t+1},a)+\omega\sum Q_t^*(s_t,a_i)\right) \quad (9)$$

where, $\sum Q_t^*(s_t,a_i)$ is the sum of the qualities of other cooperative nodes in the same set with the same state and action, and $\omega\in[0,1)$ is the discount factor which determines the importance of qualities of cooperative nodes. In Eq.(9), the quality function is not only updated according to current reward and maximum quality at future state, but also related to the qualities of cooperative nodes, which can accelerate the process of reinforcement learning.

Algorithm 1 shows the proposed RL-INRC algorithm. Each node will repeat step 3-7 during the training phase and tends to select the actions which can increase the current reward and long-term revenue gradually. Finally, the actions selected by the nodes in network will converge to the optimal transmission path.

Algorithm 1: RL-INRC algorithm

```
1.  Initialize $Q_0(s,a)=0$, $t=1$
2.  For the node to forward data packet at time t
3.  Pick up $s_t=\{d,l\}$
4.  Calculate $\pi(s_t,a_i)$ by Eq.(5)
5.  Select next hop $a_i$ with probability $\pi(s_t,a_i)$
6.  Calculate $R_t$ by Eq. (7) and observe $s_{t+1}$
7.  Update $Q_{t+1}(s_t,a_i)$ by Eq. (9)
8.  Go to step 2 with $t=t+1$
```

5 Performance Evaluation

5.1. Simulation Design

To evaluate the performance of the proposed algorithm, we design the simulation environment as follows: The network topology is the same in Figure 1. There are $n_1\left(n_1=\{1,2,3,4,5\}\right)$ cooperative node sets between source and destination, and $n_2\left(n_2=\{1,2,3,4,5\}\right)$ nodes in each set. Each node can communicate directly with all of the nodes in its set, the previous and next sets. The transmission delay ($d_e\in[1,2]$), packet loss rate ($l_e\in[0,0.1]$) and available bandwidth ($b_e\in[5,10]$) of each link are randomly generated in line with evenly distribution. The learning parameters of RL-INRC algorithm are chosen as follows. τ_0 and τ_T are set as 10 and 0.1 respectively, so that τ_m can be annealed over the training phase with betterexploration in the beginning and better exploitation near the end. α and γ are set as 0.1 and 0.5 according to related works on Q-learning algorithm. ω is also set as 0.5 so that the the importance of future rewards is the same as qualities of cooperative nodes.

We compare the performance of RL-INRC algorithm with two routing algorithms, OSPF algorithm and ant colony algorithm. OSPF is the most widely used intra-domain routing protocol nowadays and is an adaptive link-state protocol. The three parameters of link i are weighted combined as the cost of link i, shown as Eq. (10), and Dijkstra method is used to calculate the routing with minimum cost between source and destination

$$c_i=\phi_1 d_i+\phi_2 l_i+\phi_3/b_i \quad (10)$$

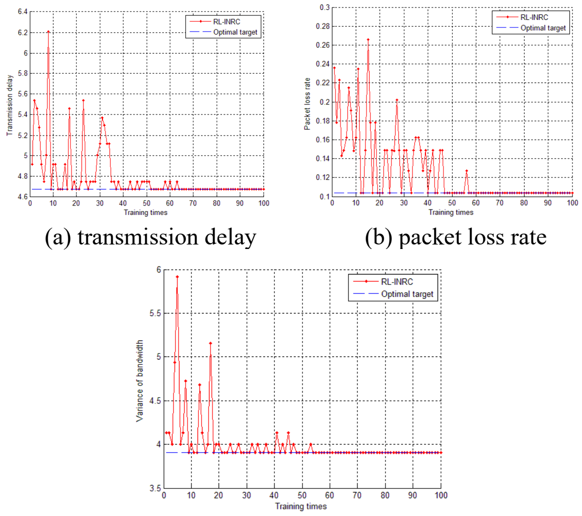
Ant colony algorithm is a parallel computational paradigm that allows exploitation of positive feedback as a search mechanism for QoS routing. Ants are more likely to follow the trail with more pheromone. The calculation of pheromone is shown in Eq.(11), where t_i is the pheromone of link i, t_i^k is the pheromone increased by ant k, and ρ is pheromone evaporation rate which is usually set as 0.5. t_i^k is calculated by Eq. (12) in this simulation, where d and l are the transmission delay and packet loss rate of the path of ant k, and *band* is the set of final bandwidth of all the links.

$$t_i=(1-\rho)t_i+\sum_k t_i^k \quad (11)$$

$$t_i^k=\begin{cases}\phi_1 d+\phi_2 l+\phi_3\,\mathrm{var}(band) & \text{if ant } k \text{ traverses link } i\\ 0 & \text{others}\end{cases} \quad (12)$$

5.2. Performance of RL-INRC algorithm

Firstly, we simulate and evaluate the performance of RL-INRC algorithm to optimize one parameter with no constraint. To optimize transmission delay, the weights ϕ_2 and ϕ_3 in Eq. (7) are set to zero, and so on for packet loss rate and bandwidth balancing. Figure 2 shows the optimization process of RL-INRC on transmission delay, packet loss rate and the variance of bandwidth, respectively, where $n_1 = n_2 = 3$. The dotted line in the figures means the theoretically optimal target. It is easy to see that the proposed RL-INRC algorithm can converge to the optimal transmission paths gradually during the training phase.



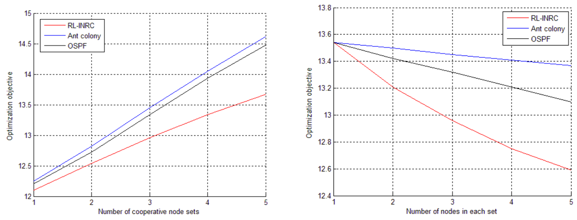
(a) transmission delay (b) packet loss rate

)c) variance of bandwidth

Figure 2. Optimization process.

Then we simulate the performance of the three algorithms when the number of nodes in the network changes. The results are shown in Figure 3. With the increase of transmission hops, the optimization objectives of ant colony algorithm and OSPF algorithm increase faster than the proposed RL-INRC algorithm. That is because the weights of non-additive parameters in OSPF algorithm induce greater errors with the increase of links on path, and it is harder for ant colony algorithm to explore the optimal solution with the increase of possible solutions. With the increase of cooperative nodes in each set, there are more possible solutions for selection, which means that it is more likely to find a better solution with lower optimization objective. It is easy to see that the optimization objective of RL-INRC decreases fast but the optimization objective of ant colony algorithm varies little. Because RL-INRC algorithm can always converge to the optimal solutions with enough training time, it works better than the other two algorithms in networks of different sizes.

Finally we compare the performance of RL-INRC and ant colony algorithm to optimize the objective described by Eqs. (1)-(4). We do not consider OSPF algorithm in this simulation because it is not able to keep away from the paths that do not meet constraint conditions. We set the objective as 30 if the path does not meet constraint conditions. The colony algorithm can converge with fewer training times, but it still converges to the locally optimal solution. RL-INRC can avoid the paths that do not meet constraint conditions after 30 times training and converges to the optimal path after 65 times training.



)a) cooperative node sets (b) nodes in each set

Figure 3. Comparison of optimization objective.

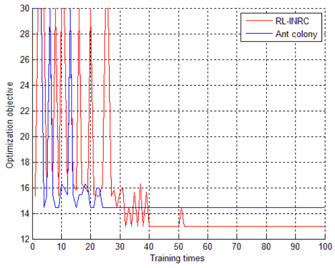


Figure 4. Optimization process with constraint conditions.

6 Conclusion

Aiming at the adaptive routing control with multiple parameters for universal scenes, we propose an intelligent routing control algorithm for MANET based on reinforcement learning. We propose a multi-hop cooperative structure and optimize the policy of node selection in each cooperative node set through the interaction with the environment. There is no need to update the network state frequently, which can save the cost of routing maintenance. Compared with other algorithms, the proposed approach can choose appropriate paths under constraint conditions, and obtain better optimization objective. In the future, we will study the problem of intelligent network routing control based on Q function approximation algorithms such as deep reinforcement learning, which should be a more meaningful and challenging work.

References

1. J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami. Internet of things (iot): a vision, architectural elements, and future directions. *Future Generation Computer Systems, 29*(7), 1645-1660(2012)
2. Z. Wang, J. Crowcroft. Quality-of-service routing for supporting multimedia applications. *IEEE Journal on Selected Areas in Communications, 14*(7), 1228-1234(2002)
3. S. Sutton, G. Barto. Introduction to reinforcement learning. *Machine Learning, 16*(1), 285-286(2005)

4. R. Widyono. The design and evaluation of routing algorithms for real-time channels(1994)
5. C. Pornavalai, G. Chakraborty, N. Shiratori. QoS based routing algorithm in integrated services packet networks. *International Conference on Network Protocols, 1997. Proceedings* (Vol.7, pp.167-174)(1997)
6. S. Chen, M. Song, S. Sahni. Two techniques for fast computation of constrained shortest paths. *IEEE/ACM Transactions on Networking, 16*(1), 105-115(2008)
7. Y. Cui, K. Xu, J.Wu. Precomputation for multiconstrained QoS routing in high-speed networks. *Joint Conference of the IEEE Computer and Communications. IEEE Societies* (Vol.2, pp.1414-1424 vol.2)(2003)
8. P. Bose, P. Morin. Competitive online routing in geometric graphs. *Theoretical Computer Science, 324*(2), 273-288(2001)
9. D. B. Magnani, I. A. Carvalho, T. F. Noronha. Robust optimization for ospf routing. IFAC-PapersOnLine, 49(12), 461-466(2016)
10. Y. Sun, L. Li, J. Qi. Cognitive networks qos routing optimization based on multi-objective genetic algorithm. *Journal of Convergence Information Technology, 7*(12), 215-225(2012)
11. D. T. Hai. Multi-objective genetic algorithm for solving routing and spectrum assignment problem. *Seventh International Conference on Information Science and Technology* (pp.177-180)(2017)
12. C. H. Qiu, Y. Gong, K. X.Zhou. The Research on QoS Routing Algorithm Based on Improved Optimization Sorting Ant Colony Algorithm. *National Conference on Electrical, Electronics and Computer Engineering*(2016)
13. R. M. Entz, H. A. Porto, R. F. D. Oliveira, R. A. D. Lima. Efficient Aircraft Routing Algorithm Based on Ant Colony Optimization. *Aiaa/issmo Multidisciplinary Analysis and Optimization Conference*(2015)
14. L. Zhang, L. B. Cai, M. Li, F. H. Wang. A method for least-cost qos multicast routing based on genetic simulated annealing algorithm. *Computer Communications, 32*(1), 105-110(2009)
15. M. Ramezani, M. Jahanshahi. Load-aware multicast routing in multi-radio wireless mesh networks using fca-cmac neural network. *Computing*(4), 1-29(2017)
16. Z. M. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue, et al. State-of-the-art deep learning: evolving machine intelligence toward tomorrow’s intelligent network traffic control systems. *IEEE Communications Surveys & Tutorials, 19*(4), 2432-2455(2017)
17. N. Kato, Z. M. Fadlullah, B. Mao, F. Tang, O. Akashi, T. Inoue, et al. The deep learning vision for heterogeneous network traffic control: proposal, challenges, and future perspective. *IEEE Wireless Communications, PP*(99), 2-9(2016)
18. T. Hu, Y. Fei. Qelar: a machine-learning-based adaptive routing protocol for energy-efficient and lifetime-extended underwater sensor networks. *IEEE Transactions on Mobile Computing, 9*(6), 796-809(2010)
19. S. C. Lin, I. F. Akyildiz, P. Wang, M. Luo. QoS-Aware Adaptive Routing in Multi-layer Hierarchical Software Defined Networks: A Reinforcement Learning Approach. *IEEE International Conference on Services Computing* (pp.25-33)(2016)