MULTIMEDIA COMMUNICATIONS TECHNICAL COMMITTEE
http://www.comsoc.org/~mmc

MMTC Communications - Frontiers

Vol. 15, No. 1, January 2020

CONTENTS

Special Issue on Intelligence for Internet-of-Things

Guest Editor: Kuan Zhang
University of Nebraska–Lincoln
kuan.zhang@unl.edu

This special issue of Frontiers focuses on Intelligence for Internet-of-Things (IoT), which embraces intelligent services with connecting a wide variety of physical objects for quality service. With the advanced networking technologies and artificial intelligence (AI), IoT can benefit a variety of new applications, including autonomous driving, healthcare, and smart city. Various research topics and projects in this area are witnessed, and the advancement can be foreseen in the future.

The first paper, by Gao et al., investigates the automatic driving technology in Internet of vehicles, in terms of perception, learning, and decision process. The main modeling scenes include confluence control of automatic vehicles, obstacle avoidance, and traffic signal control. Existing research data shows that the reinforcement learning algorithm can be used to solve the automatic driving problem in these scenes.

In the second paper, Mei et al. propose a new intelligent RAN slicing strategy with two-layered control granularity. It aims to maximize the long-term QoS of services and spectrum efficiency of network slices. A novel hierarchical deep reinforcement learning (DRL) framework is proposed based on the multi-time scale Markov decision process. This hierarchical DRL framework is the convergence of a modified deep deterministic policy gradient and double deep-Q-network algorithm. Simulation results are shown to validate the proposed framework.

The third paper focuses on two-level access management framework. The authors propose functions of five core components for this framework, including two blockchain network. The low-level blockchain is designed for local area access control of two devices that can communicate directly. The high-level blockchain network is used for global area access control for devices that cannot communicate directly. The authors also show simulations results to validate the proposed two-level access management framework especially for scalable IoT scenarios.

Finally, I want to thank all the submitting authors and people involved in the effort of producing this issue. My gratitude and recognition go to all these contributors, and hope people will find this special issue informative and useful.

Kuan Zhang joined the Department of Electrical and Computer Engineering at the University of Nebraska–Lincoln (UNL) as an assistant professor in September 2017. His research interests cover broad areas of cyber security and cyber physical systems, including network and system security, privacy, big data analysis, social network, e-healthcare system, vehicular communications, blockchain, cloud/edge computing, and Internet-of-Things. He was the recipient of Best Paper Award in IEEE WCNC 2013, Securecomm 2016, and BigDataSE 2019.

Reinforcement Learning for Automatic Driving in Internet of Vehicles

Pengzun Gao and Long Zhao
Intelligent Computing and Communications Lab, Beijing University of Posts and Telecommunications
gaopengzun@bupt.edu.cn

1. Introduction

In today's transportation system, there are often a plenty of traffic jams and traffic accidents. According to a research report released by the National Bureau of Statistics of China, a total of 2.03 million traffic accidents occurred in China in 2017, resulting in approximately 63,000 deaths. Traffic jams can happen in different areas, especially in the morning and evening rush hours. With the related research of the intelligent transportation system, more and more researchers began to study the Internet of Vehicles [1,2]. They hope to solve the problem of traffic congestion and try to avoid the occurrence of traffic accidents through the Internet of Vehicles related technologies [3]. In recent years, intelligent control and other aspects have made progress, because of the rapid development of artificial intelligence technologies such as deep learning and deep reinforcement learning [4-6]. Artificial intelligence technology can be applied in the field of Internet of Vehicles to develop automatic driving technology.

On one hand, automatic driving technology can replace human drivers for driving. That is to say, automatic driving technology can reduce traffic accidents caused by human factors such as drunk driving and fatigue driving. On the other hand, automatic driving technology can well interact with environmental information to provide effectively path planning, control traffic light and improve the operation efficiency of the transportation system.

Automatic driving technology is mainly divided into six levels [7], as shown in Table 1. Currently, the development of automatic driving is mainly in the L3 to L4 level, and the L3 level of automatic vehicles can basically achieve mass production. Automatic driving is a multi-agent interaction process, which needs to continuously obtain information from the environment and makes corresponding control and decisions, so it is a big challenge to design a supervised learning data set covering all scenarios [8]. At the same time, considering that deep reinforcement learning has the advantage of automatically learning action strategies, so the reinforcement learning for research on automatic driving technology in the Internet of Vehicles has become one of the most popular directions.

Table 1: Automatic driving level [9]

Automatic Driving Level	Definition
L0	The human driver fully operates the car and can be assisted by warning and protection systems during driving.
L1	The driving environment provides driving support for the steering wheel or speed, and other driving actions are operated by human drivers.
L2	The driving environment provides driving support for many operations in the steering wheel and speed, and other driving actions are operated by human drivers.
L3	The automatic driving system completes all driving operations. According to the system request, the human driver provides an appropriate response.
L4	The automatic driving system completes all driving operations. According to the system request, the human driver does not necessarily need to respond to all system requests under limiting the road and environmental conditions.
L5	The automatic driving system completes all driving operations. The human driver takes over the car in any time under all road and environmental conditions.

2. Automatic Driving in IoV

2.1. Structure of Automatic Driving

The automatic driving vehicle in the Internet of Vehicles contains three important processes, the perception process, the learning process and the decision process respectively [6]. Their relationships with the surrounding environment are shown in Figure 1. Automatic driving technology can be achieved only when the three parts work closely together.

2.1.1 Perception Process

The first process is the perception process, which mainly refers to the process of automatic driving vehicles collecting information from the surrounding environment. The information generally includes state about the environment outside the vehicle, such as roads, pedestrians, and traffic signals, state about the vehicle such as the temperature inside the vehicle, and the location of the vehicle. Most of the first two types of information are collected by traditional vehicles with normal sensors and radars. However, automatic driving vehicles are equipped with a variety of lidar and millimeter-wave radars, multiple high-definition cameras, and a large number of high-resolution sensors [10] to collect

them. For position information, GPS, high-precision maps and other positioning systems are utilized to achieve vehicles' position. Meanwhile, V2X and V2V communication are also used to transmit vehicle position information within the Internet of Vehicles [11].

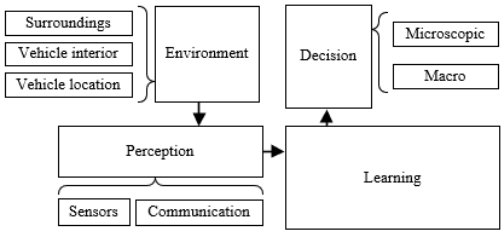


Figure 1: Important structures of automatic driving technology

2.1.2 Learning Process

The second part is the learning process, which uses deep learning and deep reinforcement learning to train the model of automatic driving vehicles. In order to drive safely and efficiently for vehicles in the transportation system, CNN and RNN can be used to deal with the data obtained by the perception system. Then DQN and DDPG algorithms can be used to train the model of automatic driving vehicles. Automatic driving vehicles which are trained by reinforcement learning algorithms can perform well to unexpected situations in the actual transportation system, because reinforcement learning has good robustness.

2.1.3 Decision Process

The last part is the decision process, which mainly makes decisions on the actual situation according to the model trained in the learning process, including the microscopic and macro aspects. The microscopic aspects include vehicle control, such as brake control, steering wheel control. The macro aspects include obstacle avoidance and traffic signal planning for automatic driving vehicles.

2.2. The main scenes of automatic driving technology

The main scenes of automatic driving technology in the Internet of Vehicles are vehicle confluence, obstacle avoidance and crossing passing. All three scenes require the interaction of automatic driving vehicles or devices in the Internet of Vehicles with the environment, which is more suitable to use reinforcement learning technology to achieve.

2.2.1 Vehicle Confluence

The first is the scene of vehicle confluence which is shown in Figure 2. Automatic driving vehicles must pass from the auxiliary road to the main road where vehicles may randomly appear. The relevant modeling of the Internet of Vehicles needs to consider the traffic rules and actual traffic conditions.

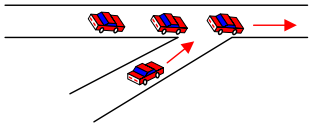


Figure 2: The scene of vehicle confluence

2.2.2 Obstacle Avoidance

The second is the obstacle avoidance scene. As shown in Figure 3, automatic driving vehicles need to identify possible obstacles and avoid them in time. This kind of scene is very common, for example, pedestrians who do not follow the traffic rules cross the road in the city. As an automatic driving vehicle, it is necessary to know the dangerous behavior of the pedestrian in no time and brake urgently to ensure the safety of pedestrians and other vehicles.

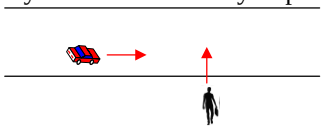


Figure 3: The scene of obstacle avoidance

2.2.3 Crossing Passing

The last scene is the crossing passing scene, as shown in Figure 4. The efficiency of vehicles passing through the crossing is related to the distribution of the length of traffic lights time. The length of traditional traffic lights time is

fixed, which may cause inefficiencies in passing crossing. In the Internet of Vehicles, we hope that the length of the green light time can be adjusted according to the actual situation. Automatic driving vehicles can use the camera and V2X communication technology to achieve the information of the traffic light time, which can reasonably plan vehicles' travel routes and speed to improve the passing efficiency of crossing.

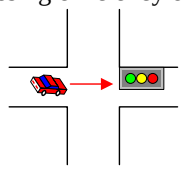


Figure 4: The scene of crossing passing

3. Application of Reinforcement Learning in Internet of Vehicles

3.1 Reinforcement Learning Overview

Reinforcement Learning (RL) refers that an agent which can perceive changes in the environment interacts with the environment to exchange state information and continuously changes its own action strategy to receive optimal reward value. The process of maximizing and then achieving the optimal result is generally shown in Figure 5. In fact, reinforcement learning can effectively obtain information from the environment, make corresponding actions, and continuously optimize until the result is optimal [7]. At the same time, reinforcement learning does not require sample labeling, and obtains information directly from the environment. That is to say, reinforcement learning is very robust. In recent years, the main reinforcement learning algorithms include Q-learning, Deep Q-Networks (DQN) and Deep Deterministic Policy Gradient (DDPG)

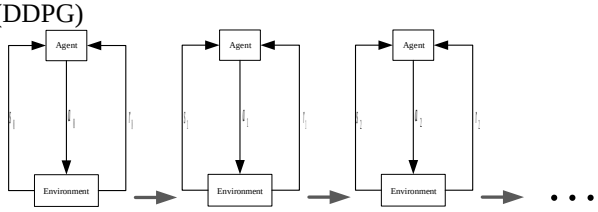


Figure 5: Reinforcement learning process

3.1.1 Q-learning Algorithm

Q-learning algorithm is a classic reinforcement learning algorithm, which is a model-free asynchronous strategy algorithm [12]. The main idea is to establish a table of Q values corresponding to all possible actions in each state, and then continuously optimize the state-action value function by selecting the largest Q value to obtain the optimal value. The specific update process is shown in Equation 1.

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left(r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right) \qquad (1)$$

Among them, s',a' respectively represent the state and action at the next moment, r represents the instant reward value at the current moment, α and γ respectively represents the learning rate and discount rate.

The Q-learning algorithm is very simple, but its application is more limited. It is applicable to the case where the state space and the action space are both discrete and small.

3.1.2 Deep Q-Networks Algorithm

Mnih proposed deep Q network algorithm [6] in order to solve the problem of Q-learning algorithm mentioned above. The DQN algorithm does not need to construct a big Q value table, but instead approximates the value function by the neural network approximate output, as shown in Equation 2.

$$Q(s,a;\theta) \approx Q^{\pi}(s,a) \qquad (2)$$

The neural network of the DQN algorithm learns the parameters θ to make it approximate the optimal value function. Similar to the Q value update process of the Q-learning algorithm, the neural network of the DQN algorithm needs to update the network weights using gradient descent (Equation 3):

$$\begin{aligned} \nabla_{\theta} L(\theta) &= E\left[y \nabla_{\theta} Q(s,a;\theta)\right] \\ y &= r + \gamma \max_{a'} Q(s',a';\theta) - Q(s,a;\theta) \end{aligned} \qquad (3)$$

The flow chart of the DQN algorithm is shown in Figure 6.

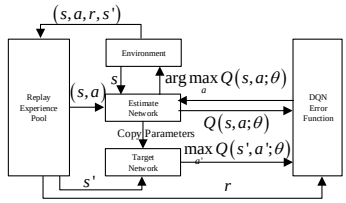


Figure 6: DQN algorithm flow

3.1.3 Deep Deterministic Policy Gradient Algorithm

To achieve reinforcement learning in continuous action space, Google ’s DeepMind team proposed Deep Deterministic Policy Gradient algorithm based on DQN [13]. The DDPG algorithm not only absorbs the structures of the DQN algorithm but also increases experience playback and a fixed target network. At the same time, the Actor-Critic algorithm is used. The actor network is used to give actions and input to the critic network for scoring. The overall flow chart of DDPG is shown in Figure 7.

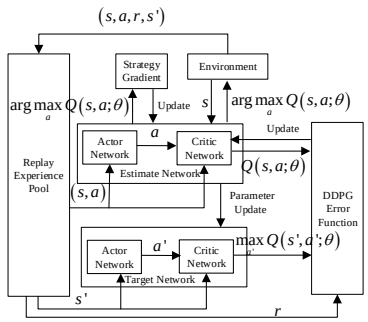


Figure 7: DDPG algorithm flow

The weights of DDPG's estimated network and target network update as the same as DQN’s. Only the estimate network needs to be trained and updated in the DDPG algorithm, which includes two parts named as the actor network and the critic network. And the parameter of the two networks are updated according to Equation 4 and Equation 5, respectively.

$$\begin{aligned}\nabla_{\theta^\mu} J &\approx E\left[\nabla_{\theta^\mu} Q\left(s,a;\theta^Q\right)\Big|_{s=s_t,a=\mu\left(s_t;\theta^\mu\right)}\right] \\ &= E\left[\nabla_a Q\left(s,a;\theta^Q\right)\Big|_{s=s_t,a=\mu\left(s_t;\theta^\mu\right)} \nabla_{\theta^\mu}\mu\left(s;\theta^\mu\right)\Big|_{s=s_t}\right]\end{aligned} \tag{4}$$

$$\begin{aligned}\nabla_\theta L(\theta) &= E\left[y_q \nabla_\theta Q\left(s,a;\theta^Q\right)\Big|_{\substack{s=s_t,\\ a=\mu\left(s_t;\theta^\mu\right)}}\right] \\ y_q &= y - Q\left(s,a;\theta^Q\right)\Big|_{\substack{s=s_t,\\ a=\mu\left(s_t;\theta^\mu\right)}} \\ y &= r + \gamma \max_{a'} Q\left(s',a';\theta\right)\end{aligned} \tag{5}$$

To get an action with a high Q value, the actor network only needs to update itself by the feedback score of the critic network. The DDPG algorithm can be applied in a continuous state space and action space, while also maintaining high computational efficiency, stability, and the irrelevance between data.

3.2 The Application of Reinforcement Learning

Reinforcement learning can interact with the environment and output the optimal behavior, which meet the requirement of automatic driving technology. Reinforcement learning can treat the Internet of Vehicles

system as a whole, increasing the robustness of the system. According to the three scenes proposed in the second part, three kinds of model were built by reinforcement learning.

3.2.1 Automatic Driving Vehicle Confluence

To achieve the convergence of automatic driving vehicles, we must first solve the problem of vehicle control. Chae suggested to use DQN to establish a braking system [14], including four levels of braking from no braking to strong braking. Chae took advantage of the characteristic that the DQN algorithm can only be applied to the discrete action space, discretizing the action space of the brake system. And the final results show that the brake system based on DQN performs ideal control in various uncertain environments behavior.

Based on the DQN algorithm, a vehicle confluence model for automatic driving can be established, considering the environmental data at the current and historical moments of the auxiliary road. There is no limit to the speed of the remaining environmental vehicles. The vehicles in the auxiliary road can choose the speed freely. In this scene, the state space of the DQN algorithm (including main road vehicles, agent vehicle speeds, main road vehicles and agent vehicle longitudinal distances, main road vehicles, agent vehicle lateral distances) is continuous, and action spaces (the measures that the smart car will take, namely acceleration or deceleration) are discrete. Through simulation verification, it is found that as the number of trainings increases, the probability of successful vehicle confluence continues to increase, which can eventually reach more than 90%. At the same time, the model can take effective measures under different environmental vehicles' speeds to get safe vehicle confluence, which is fully expected.

3.2.2 Automatic Driving Vehicle Obstacle Avoidance

Faced with the obstacle avoidance problem of automatic driving vehicles as shown in Figure 3, the DDPG-based obstacle avoidance strategy can be used. The reason for choosing the DDPG is to make up for the deficiencies of the DQN algorithm in order to accurately control automatic driving vehicles and avoid safety problems in practical applications. At the same time, in order to solve the problems of the limited sample space of the replay experience pool of the DDPG algorithm, the training process is divided into two stages according to the training epoch. In the first stage, a smaller sample space is used, and samples with good performance and poor performance are randomly collected. In the second stage, a larger sample space is selected, and the number of samples with better performance in the later period is increased. This can improve training efficiency, ensure the stability of training and eliminate the correlation of training data. There is also a way to improve the replay experience pool, which is to increase an experience judgment mechanism to distinguish successful and failed experiences and put them into the corresponding replay experience pool. In the early stage of training, less failure experience is used to accelerate convergence, and in the later stage, the failure experience is increased to avoid overfitting. Experiments show that the two improved algorithms can get the desired results at a faster speed.

3.2.3 Traffic Signal Adjustment

As early as 1994, Mikami proposed the use of reinforcement learning to control the length of traffic lights time [15]. In recent years, the use of reinforcement learning to optimize the duration of traffic lights is increasing. You can use the existing Smooth signal control system in combination with Q-learning in reinforcement learning, which has been shown to be feasible. Since Q-learning is a model-free asynchronous strategy reinforcement learning algorithm, it does not need to determine the reward function for modeling according to the environment, but directly updates the action a in the state s [16,17], which is more suitable for existing systems.

The DQN algorithm is also used to model the traffic control system. According to the dynamic characteristics of the traffic intersection, the change of the intersection environment can be sensed in real time and the optimal control scheme can be selected. First, divide the road into equal parts named cells. Next determine the state space, which includes the number of vehicles and the average speed of each cell.

Finally, determine the action space, including only two cases to maintain the current phase increase 5 seconds green light time, and switch to the next phase. There are four phases at a single intersection, as shown in Figure 8. The design of the reward function considers the effects of delay time and vehicle speed.

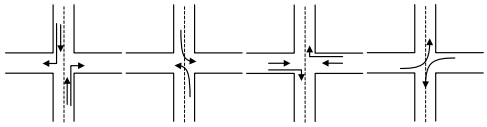


Figure 8: Four phase diagram

Similarly, the use of DQN algorithm to control urban traffic signals can also achieve effective control of urban large-scale road network traffic. First of all, the traffic system of the whole area is regarded as an agent, and the road is divided into cells according to a certain length. Secondly, the local average speed matrix and the vehicle position matrix are established, and the elements in the two matrices respectively represent the local average speed and number of vehicles in the cell. Finally, these two matrices are inputted into the DQN network as a state space. The action space is the green light time of each intersection, and the size of the action space is 2^N, which means that the entire system can effectively control the N internal traffic intersections. The reward function is set to the difference of the average speed of the global vehicle in continuous time steps. Simulation results show that the scheme can effectively control traffic, reduce traffic congestion, and improve the passing efficiency of intersections.

4. Conclusion

Automatic driving technology in Internet of Vehicles mainly includes perception, learning, and decision processes. The main modeling scenes include confluence control of automatic vehicles, obstacle avoidance, and traffic signal control. Existing research data shows that the reinforcement learning algorithm can be used to solve the automatic driving problem in the above three scenes. And the vehicle can pass the proposed scene safely and efficiently with a high probability.

References

[1] K Zheng, H Meng, P Chatzimisios, et al, “An SMDP-based resource allocation in vehicular cloud computing systems,” IEEE Transactions on Industrial Electronics, 2015, 62 (12): 7920-7928.
[2] K Zheng, Q Zheng, H Yang, et al, “Reliable and efficient autonomous driving: the need for heterogeneous vehicular networks,” IEEE Communications Magazine, 2015, 53 (12): 72-79.
[3] X Zhou, J Taylor, “DTALite: A queue-based mesoscopic traffic simulator for fast model evaluation and calibration,” Cogent Engineering, 2014, 1(1): 961345-9613463.
[4] V Mnih, K Kavukcuoglu, D Silver, et al, “Playing Atari with deep reinforcement learning,” arXiv, 2013, 1312.5602.
[5] C Dong, J Dolan, B Litkouhi, “Intention estimation for ramp merging control in autonomous driving,” Proceedings of 2017 IEEE Intelligent Vehicles Symposium, Washington D. C., USA: IEEE Press，2017:1584-1589.
[6] V Mnih, K Kavukcuoglu, D Silver, et al, “Human-level control through deep reinforcement learning,” Nature, 2015, 518(7540): 529–533.
[7] Q Zheng, K Zheng, H Zhang, et al, “Delay-optimal virtualized radio resource scheduling in software-defined vehicular networks via stochastic learning”, IEEE Transactions on Vehicular Technology 65 (10), 7857-7867, 2016.
[8] SAE International, “Taxonomy and Definitions for Terms Related to On—Road Motor Vehicle Automated Driving Systems,” 2014.
[9] SAE International, “Taxonomy and Definitions for Terms Related to On—Road Motor Vehicle Automated Driving Systems,” 2014.
[10] D Zermas, I Izzat, N Papanikolopoulos, “Fast segmentation of 3D point clouds: A paradigm on LiDAR data for autonomous vehicle applications,” 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017.
[11] K Zheng, L Hou, H Meng, et al, “Soft-defined heterogeneous vehicular network: architecture and challenges,” IEEE Network, 2016, 30 (4): 72 – 80.
[12] C Watkins, P Dayan, “Q-learning,” Machine Learning, (1992), 8(3-4): 279–292.
[13] P Timothy, J Jonathan, P Alexander, et al, “Continuous control with deep reinforcement learning,” Computer Science, 2015, S (6): A187.
[14] H Chae, C Kang, B Kim, et al, “Autonomous braking system via deep reinforcement learning,” 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Japan, 2017.
[15] S Mikami, Y Kakazu, “Genetic reinforcement learning for cooperative traffic signal control,” Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence, Orlando, 1994.
[16] R Sutton, A Barto, “Reinforcement learning: An introduction,” IEEE Transactions on Neural Networks, 1998, 9(5): 1054.
[17] Y Lecun, Y Bengio, G Hinton, “Deep learning,” Nature, 2015, 521(7553): 436-444.

Pengzun Gao received the B.E. degree in Information and Communication Engineering with high distinction from Beijing University of Posts and Telecommunications, China in 2019. His current research interests include Intelligent Transportation Systems, Reinforcement learning, Artificial Intelligence, and Deep Learning.

Long Zhao received the Ph.D. degree from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2015, where he is currently an associate professor. From April 2014 to March 2015, he was a Visiting Scholar at the Department of Electrical Engineering, Columbia University. His research interests include wireless communications and signal processing.

Intelligent Radio Access Network Slicing for Service Provisioning Toward 6G

Jie Mei, and Xianbin Wang
Department of Electrical and Computer Engineering, Western University, Canada
{jmei28, xianbin.wang}@uwo.ca

1. Introduction

Our society is undergoing a digitization transformation, by integrating digital technologies into different aspects of our daily life with an expected plethora of emerging services and applications, including high precision manufacturing, autonomous driving, smart home, intelligent transportation systems and more. These vertical services with far-reaching effects are characterized by a highly diverse set of quality of service (QoS) requirements [1]. However, as the deployment of fifth generation (5G) networks is ongoing, it appears that simply increase transmit data rate to an unprecedented level does not ensure the flexibility required to support diverse vertical services with heterogeneous QoS requirements [2] [3]. Therefore, to overcome the inherent limitation of 5G standards, in the 5G-beyond and sixth generation (6G) wireless systems, researchers need to design extremely flexible and adaptive network architecture that can truly integrate such diverse services into the same architecture.

One effective solution for diverse QoS provisioning is network slicing, which divides the physical network into multiple virtual logical networks, referred to as network slices, co-existing over a common shared physical network substrate. The dimensionity of each network slice is customized best to fulfill the specific QoS requirements [4]. However, network slicing will introduce much more complexity into networks [5]. To address this challenge, as elaborated in our previous work [6], it is necessary to utilize Artificial Intelligence (AI) technologies to achieve highly flexible network slicing scheme.

To improve the effectiveness of network slicing, this paper proposes an intelligent radio access network (RAN) slicing scheme with self-configuration and self-optimization capabilities, which is embedded in a customized DRL framework, to maximize the long-term QoS and spectrum efficiency (SE) of network slices. Particularly, our contributions include: i) a new intelligent RAN slicing strategy with two-layered control granularity, which aims at maximizing the long-term QoS of services and spectrum efficiency (SE) of network slices. The proposed method consists of an upper-level controller to ensure QoS performance, which enforces loose control by performing slice configuration adaption according to the long-term dynamics of service traffic. The lower-level controller is to improve SE of slices, by tightly scheduling radio resources to users at the small timescale.; ii). To achieve the proposed RAN slicing strategy, a novel hierarchical deep reinforcement learning (DRL) framework is proposed based on the multi-time scale Markov decision process. The hierarchical DRL framework is the convergence of a modified deep deterministic policy gradient (DDPG) and double deep-Q-network (Double DQN) algorithm.

2. Design of Intelligent RAN Slicing and Problem Formulation

Consider a typical downlink cellular network system with a single Base Station (BS). The time dimension is partitioned into Transmission Time Interval (TTI) of 1 *ms*, indexed by $t = \{0,1,\dots\}$. The bandwidth is divided into a set of physical resource blocks (PRBs), denoted as $\mathcal{F} = \{1,\dots,F\}$, for each TTI. Assume that the cellular network is split into a set $\mathcal{N} = \{1,\dots,N\}$ of network slices. The UEs associated with slice $n \in \mathcal{N}$ is denoted as the set $\mathcal{U}_n$, where $\mathcal{U} = \cup_{\forall n} \mathcal{U}_n$ and $\cap_{\forall n} \mathcal{U}_n = \emptyset$.

2-A: Hierarchical RAN slicing control strategy

In this study, following the idea of our previous work [6], a hierarchical control strategy, denoted as $\pi = \{\pi_{\mathrm{C}}, \pi_{\mathrm{R}}\}$, is proposed. As shown in Figure 1 and Table I, the upper-level controlling policy π_{C} is responsible for adjusting the configuration of slices according to the dynamics of service traffic at the large time-scale. Specifically, the basic time unit of upper-level control is defined as an epoch, indexed by $k = \{0,1,\dots\}$, each epoch is corresponding to ΔT consecutive TTIs. Based on the configuration determined by the upper-level, the lower-level controlling policy π_{R} directly manages radio resource scheduling according to the dynamics of physical layer at the fast timescale.

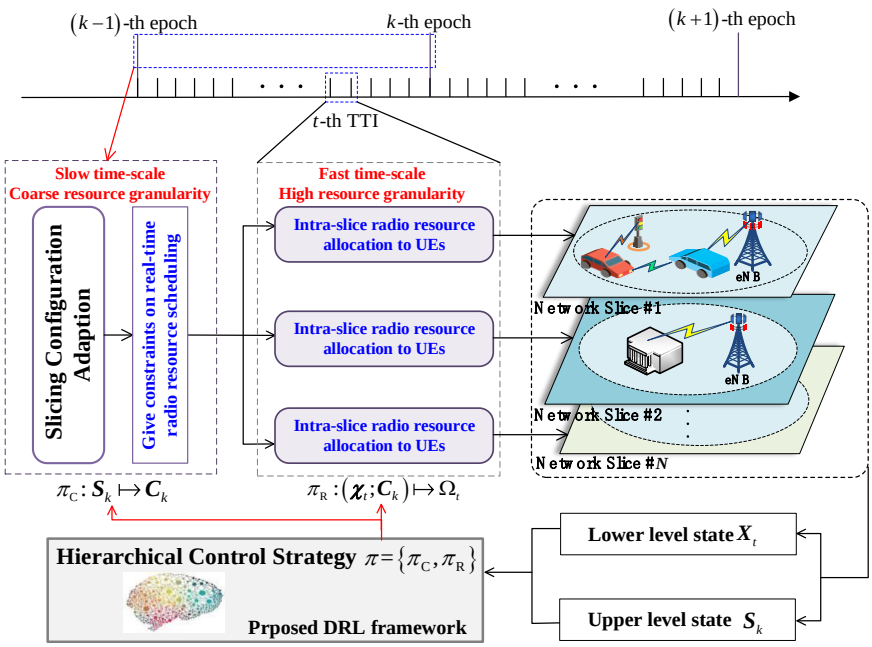


Figure 1: A conceptual diagram of the hierarchical RAN slicing control strategy.

Table I Control granularity of the proposed hierarchical control strategy.

	Upper-level control (Slice configuration adaption)	Lower-level control (Intra-slice radio resource management)
Granularity in the time domain	Slow timescale, typically hundreds of milliseconds	Fast timescale, every TTI (i.e., 1 ms in LTE)
Granularity in the radio resource domain	Set data rate constraints of UE in each network slice (loose control)	One PRB (tight control)

1). Upper-level Control

Based on the global status of the whole network, the upper-level controller will tune the slice configuration to ensure and improve the QoS of services. It is noteworthy that the upper-level controller is not directly involved in the real-time radio resource scheduling process. Essentially, the slice configuration has two main functions,

- *Ensuring QoS performance of slices*: It transforms the dynamics of service traffic as well as the QoS requirements of services into the data rate constraint of users in each slice. Here, we utilize the guaranteed bit rate (GBR) of UEs in each slice to ensure the QoS performance.
- *Ensuring isolation among slices*: It guarantees the traffic overload in one slice does not negatively affect the QoS performance experienced by other slices. Herein, each slice is imposed with the maximum bit rate (MBR) of UEs to ensure traffic isolation.

In this study, the upper-level controlling policy π_C the global state of the whole network at the beginning of each epoch and accordingly makes control decisions. Firstly, the upper-level state of the entire system at the k-th epoch is defined as, $\boldsymbol{S}_k = \{\boldsymbol{S}_{n,k} | \forall n \in \mathcal{N}\}$, where $\boldsymbol{S}_{n,k}$ is the upper-level state of slice n. $\boldsymbol{S}_{n,k}$ is characterized by i) the average packet arrival rate of UE (in terms of arrival packets per TTI); ii) the average packet latency (in TTI); and iii) the average packet reliability of active UEs in slice n within the last epoch.

Furthermore, define the upper-level controlling policy π_C to be a mapping from the global state of the whole network $\boldsymbol{S}_k$ to a suitable slice configuration $\boldsymbol{C}_k$, which is given by

$$\pi_C: \boldsymbol{S}_k \to \boldsymbol{C}_k,$$

where the slice configuration at k-th epoch is defined as $\boldsymbol{C}_k = \{R_n^{\min}, R_n^{\max} | \forall n \in \mathcal{N}\}$, $R_n^{\min}$ is the Guaranteed Bit Rate (GBR) of UE in slice n, and $R_n^{\max}$ is the Maximum Bit Rate (MBR) of UE in network slice n.

2). Lower-level Control

During the k-th epoch (i.e., from TTI $k\Delta t + 1$ to TTI $(k+1)\Delta t$), once the slice configuration $\boldsymbol{C}_k$ is taken by the upper-level controlling policy π_C, then the radio resource allocation scheme at each TTI is restricted by inequation (6) given by $\boldsymbol{C}_k$. Then, the remaining problem is to allocate PRBs and transmit power to active UEs according to the local state of slices. The lower-level state of the whole network at the t-th TTI is $\boldsymbol{X}_t = \{\boldsymbol{X}_{n,t} | \forall n \in \mathcal{N}\}$, where $\boldsymbol{X}_{n,t} =$

$\{Q_{i,t}, \boldsymbol{H}_{i,t} | \forall n \in \mathcal{N}\}$ is the lower-level state of slice n at the t-th TTI, $Q_{i,t}$ is the queue length of UE i and $\boldsymbol{H}_{i,t} = \{h_{i,j,t} | \forall j \in \mathcal{F}\}$ represents the channel state of UE i, which is a vector the channel gain $h_{i,j,t}$ on each PRB $j \in \mathcal{F}$.

Define the lower-level controlling policy π_{R} to be a mapping from the lower-level state of the whole network $\boldsymbol{X}_t$ to the PRB and power allocation of active UEs in each slice, which is expressed as

$$\pi_{\mathrm{R}}: (\boldsymbol{X}_t, \boldsymbol{C}_k) \rightarrow \Omega_t, k\Delta T \leq t < (k+1)\Delta T,$$

where Ω_t is the radio resource allocation scheme for UEs in network slice n.

2-B: Problem Formulation

In this study, we set the comprehensive utilization of the network slice, which is related to the QoS performance of service (i.e., packet latency and packet drop rate, PDR) and the spectrum efficiency (SE) of slice. Our objective is to obtain the optimal RAN slicing control strategy, which maximizes the expected long-term utility function of all network slices while fulfilling the constraints on radio resource scheduling.

In this study, the stochastic optimization problem $\mathcal{P}$ can be modeled as a twin time-scale Markov Decision Process (MDP) [7]**Error! Reference source not found.**: the upper-level control process is an infinite horizon MDP at the slow timescale, while the lower-level control process is a finite (ΔT-TTIs) horizon MDP at the fast timescale.

The proposed hierarchical control strategy $\pi = \{\pi_{\mathrm{C}}, \pi_{\mathrm{R}}\}$ is a *nested structure*, when the upper-level policy π_{C} and the lower-level policy π_{R} are trained simultaneously, the transition probabilities between the upper-level states will continue to change if the lower-level policy π_{R} continues to be updated. In a further way, under the non-stationary distribution of upper-level states, the DRL algorithms will struggle to learn the upper-level policy π_{C}, since the distribution of upper-level states should be stable for effective learning.

3. Solution based on Hierarchical Deep Reinforcement Learning Framework

We can first train the lower-level control policy π_{R} under all candidate slice configurations, then learn the upper-level control policy π_{C} based on the converged lower-level control policy π_{R}^*. Followed this idea, we propose a hierarchical DRL framework, which is comprised of two stages:

Stage I (Learning of the lower-level control policy): By utilizing both convex optimization tool and policy gradient method, learning the converged lower-level policy π_{R}^* under all candidate slice configuration $\boldsymbol{C}_k \in \mathcal{C}$, which can be achieved by maximizing the SE of slices during each epoch under slice configuration $\boldsymbol{C}_k$.

The subproblem in Stage I is a constrained MDP with huge mixed-integer action space. To fit this subproblem into a typical MDP and reduce the computational complexity, we propose a novel action space reducing approach and obtain the converged lower-level control policy π_{R}^* based on a modified Deep Deterministic Policy Gradient (DDPG) method.

Stage II (Learning of the upper-level control policy): Then, with the converged lower-level policy π_{R}^*, learning of the upper-level policy π_{C} can be realized by solving the reduced version of the target problem, which can be stated as a standard MDP problem, we utilize the double deep Q-network (Double DQN) algorithm to solve it.

4. Numerical Results

We consider the following reference scheme as well, i) Baseline 1 (Hard Slicing): The configuration of slices is pre-defined. Here, we set each slice with minimum configuration to avoid traffic overload; ii). Baseline 2 (Slicing with coarse control granularity only): This scheme is the service-demand based network slicing method proposed in [8]. Simulation results confirm the convergence and effectiveness of our proposed intelligent RAN slicing scheme.

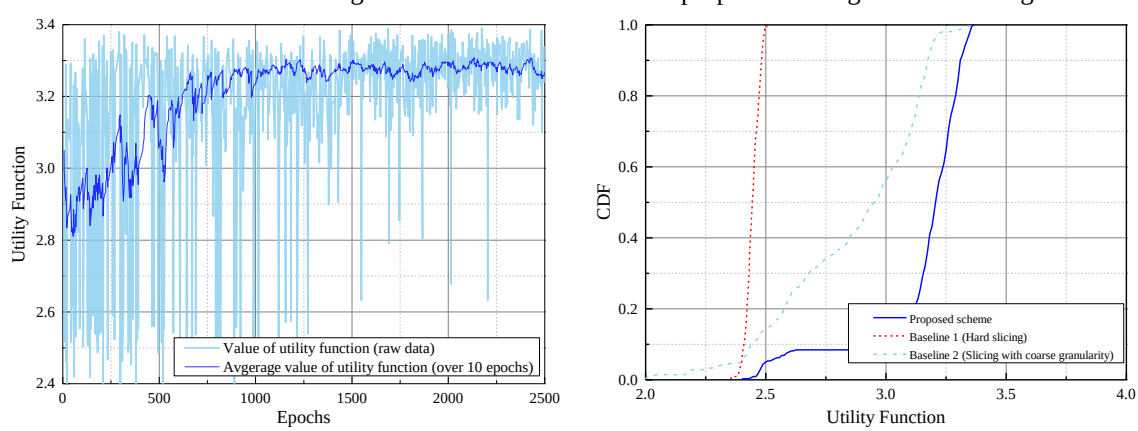


Figure 2: The utility function reflecting slicing control performance, (Left) the proposed network slicing control scheme during the training procedure and (Right) CDF of utility function with different network slicing schemes.

5. Conclusions

In this e-letter, we propose an intelligent RAN slicing strategy with multiple control granularity, which aims at maximizing the long-term QoS of services and SE of network slices. Specifically, the proposed strategy consists of an upper-level controller and lower-level controller. The upper-level controller adapts the slice configuration to improve QoS performance at coarse granularity, while the lower-level controller, at the fine granularity, schedules PRB and power allocation to active UEs in each network slice. Then, based on multi timescale MDP model, we propose a novel hierarchical deep reinforcement learning (DRL) framework, an integration of modified DDPG and double-DQN algorithm, for learning the optimal RAN slicing strategy. Simulation results show that the proposed scheme has stable convergence performance, and achieves higher QoS performance, fairness and data throughput compared to the baseline schemes. However, this paper does not consider the control signaling overhead generated by network slicing and its impact on control and QoS performance. This issue need to be further studied.

References

[1]. C. Yang, W. Shen, and X. Wang, "The internet of things in manufacturing: Key issues and potential applications," *IEEE Systems, Man, and Cybernetics Magazine*, vol. 4, no. 1, pp. 6–15, 2018.

[2]. "Setting the scene for 5G: Opportunities and challenges," http://handle.itu.int/11.1002/pub/811d7a5f-en, International Telecommunications Union, Oct. 2018.

[3]. E. Calvanese Strinati, S. Barbarossa, J. L. Gonzalez-Jimenez, D. Ktenas, N. Cassiau, L. Maret, and C. Dehos, "6G: The next frontier: From holographic messaging to artificial intelligence using subterahertz and visible light communication," *IEEE Vehicular Technology Magazine*, vol. 14, no. 3, pp. 42–50, Sep. 2019.

[4]. "5G network slicing for vertical industries," Ericsson, Stockholm, Sweden, Tech. Rep., Sep. 2017.

[5]. Z. Zhang, Y. Xiao, Z. Ma, M. Xiao, Z. Ding, X. Lei, G. K. Karagiannidis, and P. Fan, "6G wireless networks: Vision, requirements, architecture, and key technologies," IEEE Vehicular Technology Magazine, vol. 14, no. 3, pp. 28–41, Jul. 2019.

[6]. J. Mei, X. Wang, and K. Zheng, "Intelligent network slicing for V2X services toward 5G," *IEEE Network*, vol. 33, no. 6, pp. 196–204, Oct. 2019.

[7]. H. S. Chang, P. J. Fard, S. I. Marcus, and M. Shayman, "Multitime scale Markov decision processes," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 976–987, Jun. 2003.

[8]. R. Li et al., "Deep Reinforcement Learning for Resource Management in Network Slicing," IEEE Access, vol. 6, pp. 74429-74441, 2018.

Jie Mei (S'18, M'19) received his B.S. degree from Nanjing University of Posts and Telecommunications (NJUPT), China, in 2013. He received his Ph.D. degree in information and communication engineering at Beijing University of Posts and Telecommunications (BUPT) in June 2019. Since August 2019, he has been a postdoctoral associate with Electrical and Computer Engineering, Western University, Canada. His research interests include intelligent communications and Vehicle-to-Everything (V2X) communication.

Xianbin Wang (S'98, M'99, SM'06, F'17) is a professor and Tier-1 Canada Research Chair at Western University, Canada. His research interests include 5G technologies, Internet-of-Things, communications security, machine learning and intelligent communications. He has over 380 publications, in addition to 29 patents and several standards contributions. He is a Fellow of the Canadian Academy of Engineering, a Fellow of IEEE, and an IEEE Distinguished Lecturer. He has served as an editor/ associate editor/guest editor for more than 10 journals. He was involved in over 50 conferences with different roles such as symposium chair, tutorial instructor, track chair, session chair and TPC co-chair.

An Access Management Framework Based on Blockchain for Internet of Things

Chong Yu[1,2], and Kuan Zhang[1]

[1]. Department of Electrical & Computer Engineering, University of Nebraska-Lincoln

[2]. Department of Computer Science and Engineering, Northeastern University, China

cyu6@huskers.unl.edu, kuan.zhang@unl.edu

1. Introduction

The Internet of Things (IoT) has great potential in facilitating diversity of applications, such as smart home, intelligent transportation, automated factory, and other specialized applications that require data collection, processing, and analysis [1, 2]. IoT is a physical network consisting of devices with the capabilities of sensing, computing, and communications, permitting these devices to gather and interchange data to achieve remote monitoring, intelligent control, and unified management [3]. Working in IoT patterns has many benefits, including energy efficiency, convenience, and plenty of automation [1]. Despite the potential that IoT raises, there are many challenges required to be overcome. For small-scale IoT, centralized access management methods are applied to administrate devices since they are within the same domain generally [5].

With the continuous expansion of IoT applications, the scale of IoT increases and the scenarios are more dynamic because of the diversity and mobility of devices [6]. Traditional centralized access management methods provide access control services in the same domain while not ensuring high-load access control in different domains. Furthermore, a rapidly growing trend in establishing IoT with wireless networks results in higher security risks [7]. It is easier for illegal devices to access resources within IoT, leading to a waste of resources and information leakage. Therefore, designing decentralized and secure access management methods is one of the most urgent needs of the Internet of Things.

The appearance of blockchain technologies brings opportunities in overcoming the above challenges of IoT. Blockchain is a shared ledger and database recording all verified transactions. Peer to peer networks, distributed consensus, and cryptographic are key technologies for blockchain [8]. Entities are equal in a blockchain system and they could cooperate to validate transactions without the intervention of a trusted third party due to the application of peer to peer network and distributed consensus [9]. Moreover, attackers cannot modify transactions saved in blockchains because transactions must be verified by the majority of entities and recorded in the longest blockchain. Meanwhile, cryptographic mechanisms guarantee the integrity of data blocks in the blockchains. Given the above characteristics, it is feasible to use blockchain technology to build a distributed and secure access management framework.

In this paper, a two-level access management framework is proposed, and functions of five core components for this framework are defined. This management framework includes two blockchain network, the low-level blockchain is responsible for local area access control, that is, realize the access control of two devices that can communicate directly, and the high-level blockchain network is responsible for global area access control, that is, realize the access control of two devices that cannot communicate directly. Finally, simulations results show that the two-level access management framework is effective in specific scalable IoT scenarios.

2. Two-Level Access Management Framework

2-A: Structure of Two-Level Access Management Framework

A decentralized two-level access management framework based on blockchain is proposed. This framework consists of blockchain networks, terminal devices, cluster header (CH)/miner, manage, and smart contract, which is depicted as Figure 1.

- Blockchain networks: There are two blockchain networks. One is the high-level blockchain network, which plays a significant role in global access management. Another one is the low-level blockchain network, which is responsible for local access management. For simplicity, both blockchain networks are private blockchains. Although private blockchains can be read by anyone, only private nodes have the privilege to write it. Fully validating high-level miners store entire blockchain and have the ability to verify transactions so that the security and stability of the network are guaranteed.
- Terminal devices: A terminal device, which is consists of sensors module, a processing module, and communication module, has the capability of data collection, information processing, and data exchange with other terminal devices. In general, terminal devices are restricted in storage, energy, and processing power.

Terminal devices within one local area can communicate directly through wireless communication. The IP address of terminal devices can be regarded as globally unique identities.

- CH/miner: Different from terminal devices, a CH/miner is a powerful device with large storage space, high energy, and fast computational speed. Terminal devices connect to a CH/miner or several CH/miners, and CH/miner connect directly to the nearest high-level blockchain miner. As the name suggests, a CH/miner plays two roles in this system. On one hand, a CH/miner acts as an interface that relays the global access request to the high-level miner. On the other hand, a CH/miner and terminal devices connected to this CH/miner form a low-level blockchain. In this scenario, the CH/miner acts as a low-level blockchain miner, which keeps track of local transactions.
- Manager: A manager refers to an entity in charge of administering access control permissions of terminal devices. Generally, a manager works in a lightweight pattern so that there is no constraint about storage space and computational capability. Therefore, any entity in IoT able to register as a manager regardless of its storage space and computational capability. Terminal devices arbitrarily choose one or more managers to register, which means that a terminal device is controlled by at least one manager. The main role of managers is to define access control regulations for the terminal devices under its govern.
- Smart contract: In our framework, a single and unique smart contract is employed, and all actions must comply with the smart contract definition. It exists permanent in this system, and no one has the right to delete it. Managers are the only authorized entities that can define and add new policies to the smart contract. The actions defined in the smart contract are shown as follows: (1) Enroll a device as a manager, and disenroll a device as a manager. (2) Enroll a device under a manager's control, and disenroll a device under a manager's control. (3) Add access control, and deny access control. (4) Manager control list query. (5) Access control permission query.

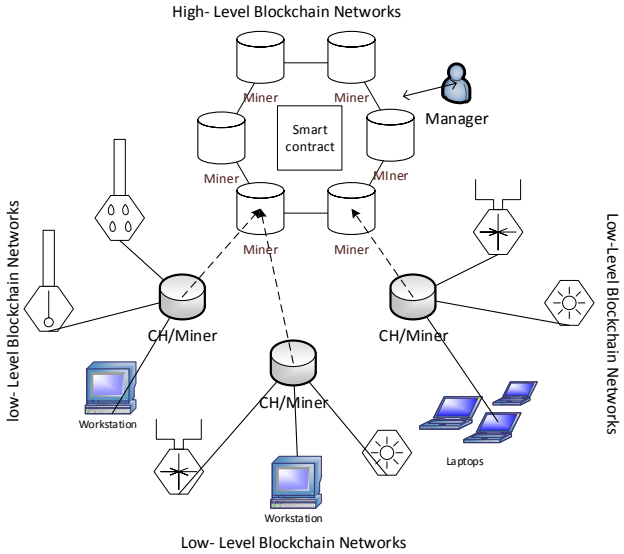


Figure 1: Two-level access management system.

2-B: Local Area Access Management

As mentioned above, terminal devices within a local area, like an office, communicate directly with each other. Each terminal device is able to ask information held by other terminal devices to provide some services, e.g. the air conditioning demands information from temperature sensors to switch on refrigeration automatically when room temperature exceeds 90 degrees Fahrenheit. Figure 2 gives detail about the local area access management. There are two terminal devices called terminal device 1 and terminal device 2. Terminal device 1 intends to access a resource R hosted by terminal device 2. In the first step, terminal device 1 transmits a request to terminal device 2 and requires a shared key from the CH/miner. Once the CH/miner receives this request, it authenticates the identity of terminal device 1 then asks for permission from terminal device 2. If terminal device 2 approves the permission, a shared key is generated by CH/miner and is allocated to devices that want to share data. After receiving a symmetric key, terminal device 1 can send resource R to terminal device 2 using this shared key. For the purpose of finish permission, the CH/miner signs the distributed key as invalid by sending a control message to devices. Devices access each other with this method brings many benefits. The first one is that the use of symmetric keys makes the communications between

devices safer. The second one is that the CH/miner holds a list of terminal devices that share data to prevent the invasion of illegal equipment.

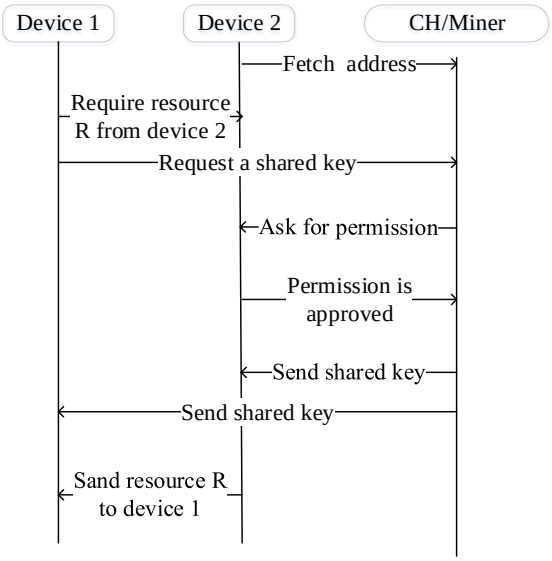


Figure 2: Local area access process.

2-C: Global Area Access Management

Apart from local access management, our proposed framework also supports global area access management. As shown in Fig. 3, there are two terminal devices named terminal device 1 and terminal device 2, and they belong to the different local areas. First, a smart contract is deployed into the high-level blockchain network. Once the smart contract is accepted, high-level blockchain nodes and CH/miners get its address. After that, high-level blockchain node M enrolls itself as a manager M. Then terminal device 1 is registered under manager M's govern. Since access control regulations for resources of a terminal device are defined by its managers, manager M adds a new access control regulation for resource R of terminal device 1 to the smart contract using the address of terminal device 1 and terminal device 2. When terminal device 2 intends to obtain resource R that owned by terminal device 1, it transmits this request to the CH/Miner which it connects. After receiving this request, the CH/miner relays this request to the nearest miner in the high-level blockchain network linked to it. High-level miners work in fully validating mode, so they store the blockchain. CH/miner inquires the access policy of terminal device 1 the high-level miner. Once this access policy is acquired by the CH/miner, it transmits this policy to terminal device 2. Terminal device 2 takes corresponding measures to access terminal device 1 based on the obtained policy. Finally, terminal device 2 received data of resource R from terminal device 1.

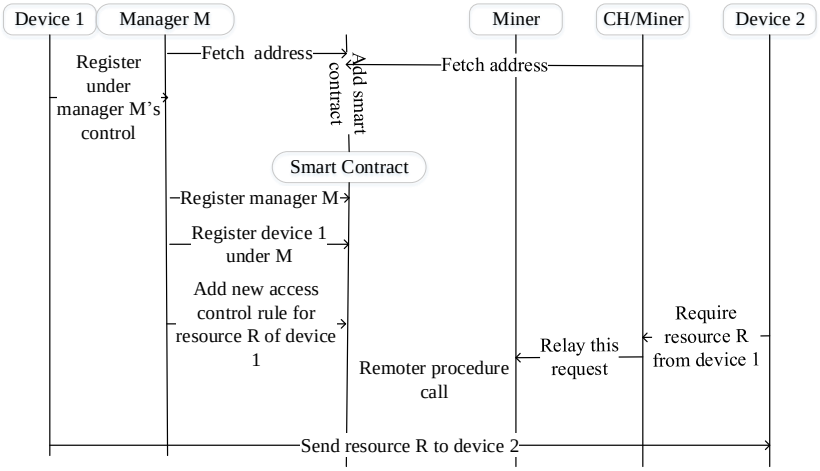


Figure 3: Global area access process.

3. Performance Evaluation

In this section, throughput of the proposed access management framework is carried out using Docker and Vertigo, and the security of this framework is analyzed.

3-A: Throughput

To evaluate the ability of the presented system to cope with high load, we chose throughput as a measure. Two scenarios are set to assess local access control and global access control. In the first scenario, devices request resources within one domain. In the second scenario, devices request resources from different domains through CH/Miner.

The throughput performed for both scenarios is shown in Figure 4. We can see that there is an increase in throughput for global access with the growth of clients' numbers in the beginning. When the number of clients over 10, Throughput is stable and maintained at around 1000 requests per second. Another observation from Figure 4 is that the raised of the number of clients leads to the throughput of local access to increase. When the number of clients reaches 10000, throughput is about 1500 requests per second. The throughput of local access is greater than that of global access throughout the whole process. This is because global access requires more steps than local access resulting in a longer delay. From Figure 4, we can conclude our framework can reach up to 1000 requests per second for global access and up to 1500 requests per second local access. The number of devices within a domain is limited and devices can connect with several CH/Miner, so the result is acceptable.

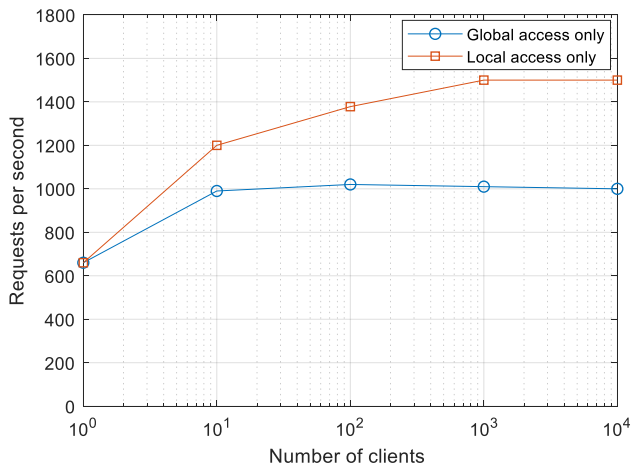


Figure 4: Throughput for global access and local access, respectively.

3-B: Security Analysis

The main security requirements for this access management framework is to prevent illegal devices to access IoT resources. For local access management, both parties to the resource exchange need to be authenticated by the CH/miner and a symmetric key is required during resource transmission. It is difficult for an illegal device to obtain authentication from CH/miner, so it cannot receive the shared key. For global access management, devices have to register under managers and all operations written in the smart contract cannot be deleted and modified. Illegal devices that do not belong to any manager cannot complete access according to the smart contract. In addition, confidentiality, integrity, and availability also should be taken into consideration [10]. Utilizing cryptographic achieves data confidentiality because the data cannot be decrypted without keys. Since the hashing function is used in blockchain networks, data integrity is assured. The existence of managers and the smart contract provides a guarantee of availability.

4. Conclusions

In this paper, we provide a solution for the scalability and security issues of access management in IoT. A fully decentralized two-level access management scheme based on blockchain is presented. Local area access control and global area access control are two main functions achieved by the scheme. Throughput evaluation results show that

the solution we propose can effectively cope with high-load situations, whether in local or global access scenarios. According to the analysis, we find that the security of this proposed scheme is guaranteed.

References

[9]. S. Soumyalatha, "Study of IoT: understanding IoT architecture, applications, issues and challenges," 1st International Conference on Innovations in Computing & Net-working (ICICN16), 2016.

[10]. I. Lee, and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431-440, 2015.

[11]. H. Dai, Z. Zheng, and Y. Zhang, "Blockchain for internet of things: A survey," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 8076-8094, 2019.

[12]. A. Dean, and M. Agyeman, "A study of the advances in IoT security," Proceedings of the 2nd International Symposium on Computer Science and Intelligent Control, 2018.

[13]. J Long, M Dong, K. Ota, et al, "Achieving source location privacy and network lifetime maximization through tree-based diversionary routing in wireless sensor networks," *IEEE Access*, vol. 2, pp. 633-651, 2014.

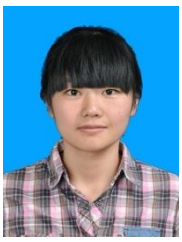
[14]. O. Novo, "Blockchain meets IoT: An architecture for scalable access management in IoT," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184-1195, 2018.

[15]. M Khan, and K Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395-411, 2018.

[16]. D. Dasgupta, and J. Shrein, "Gupta K D. A survey of blockchain from security perspective," *Journal of Banking and Financial Technology*, vol. 3, no. 1, pp. 1-17, 2019.

[17]. Z Huang, X Su, Y Zhang, et al, "A decentralized solution for IoT data trusted exchange based-on blockchain," 2017 3rd IEEE International Conference on Computer and Communications, 2017.

[18]. M. Olivier, "Database privacy: balancing confidentiality, integrity and availability," *ACM SIGKDD Explorations Newsletter*, vol. 4, no. 2, pp.20-27, 2002.



Chong Yu received the B.S. and M.S. degrees in Information Science and Engineering from the Northeastern University, China, in 2015 and 2017, respectively. She is currently pursuing the Ph.D. degree at the Department of Electrical and Computer Engineering, University of Nebraska-Lincoln, US. Her research interests include wireless sensor network, vehicle ad hoc network, and cyber security, Internet-of-Things.

Kuan Zhang joined the Department of Electrical and Computer Engineering at the University of Nebraska–Lincoln (UNL) as an assistant professor in September 2017. His research interests cover broad areas of cyber security and cyber physical systems, including network and system security, privacy, big data analysis, social network, e-healthcare system, vehicular communications, blockchain, cloud/edge computing, and Internet-of-Things. He was the recipient of Best Paper Award in IEEE WCNC 2013, Securecomm 2016, and BigDataSE 2019.