Task Migration Using Q-Learning Network Selection for Edge Computing in Heterogeneous Wireless Networks

Yi Liu[1], Jie Zheng[1](✉), Jie Ren[2], Ling Gao[1], and Hai Wang[1]

[1] Northwest University, Xi'an, China
jzheng@nwu.edu.cn
[2] Shaanxi Normal University, Xi'an, China

Abstract. For edge devices, pushing the task to other near devices has become a widely concerned service provision paradigm. However, the energy-constrained nature of edge devices makes optimizing for Quality of Service (QoS) difficult. We choose three factors as QoS: the delay limitation, the CPU usage of terminal and energy consumption. Due to the delay limitation of different tasks for edge computing and the different rates in heterogeneous wireless networks, we propose a network selection task migration algorithm based on Q-learning that captures the trade-off between QoS and energy consumption. Our approach can automatically choose a suitable network to perform task migration reasons about the task's QoS requirements and computing rate in 4G network, Wi-Fi, Device-to-device (D2D). We demonstrate a working prototype using the YOLOv3 on the Vivo X9 devices. Based on real hardware and software measurements, we achieve 27.79% energy saving and 35% reduction in delay.

Keywords: Edge computing · Heterogeneous wireless networks · Network selection · Task migration · Q-learning

1 Introduction

1.1 A Subsection Sample

Due to the limited computing and memory resource of mobile edge devices (e.g. drone, smartphone, wearable device, and automobile), they are not able to process compute-intensive tasks in time, such as image recognition, traffic navigation, and the violation of QoS may cause the economic loss, threat the human life [1]. Generally, the heavyweight tasks of edge devices are migrated to the cloud servers which equipped with high-performance computing power and large storage [2]. With the increasing network traffic and unstable network conditions, the cloud server cannot guarantee the QoS for each task. To solve this problem, researchers Jinming Wen and Hong Xing try to take advantage of other idle edge devices to process the part of the heavyweight tasks. Such as [3, 4] they focus on the D2D conditions, but miss opportunity of a wider range of devices under the heterogeneous network, and cannot achieve a global optimal results.

B. Li et al. (Eds.): IoTaaS 2019, LNICST 316, pp. 141–150, 2020.
https://doi.org/10.1007/978-3-030-44751-9_13

In this paper, we propose a network selection task migration algorithm based on Q-learning. The goal of network selection task migration algorithm based on Q-learning is to implement network selection. In this paper, we define QoS to represent the delay of the task [5]. Our approach selects the optimal network access. Through this network, we migrate tasks that cannot be completed by the edge device to the appropriate server. Our approach focuses on the performance energy consumption of the edge device and the task delay. Because of the performance limitations of the edge device, the task with high-performance requirements cannot be completed; the size of the task delay also has a great impact on the user experience. Therefore, we choose the edge devices energy consumption and the task delay as feedback variables and input them into the algorithm. The algorithm performs self-learning based on feedback. In the end, we get the most optimal network selection strategy based on Q-learning and automatically access the network. We conducted an experimental evaluation using the mobile phone Vivo X9 as an edge device. We use YOLOv3 [6] as a task model for experimental verification. Through experiments, our algorithm is compatible with edge computing heterogeneous wireless network.

The main contribution of this paper is the network selection task migration algorithm based on Q-learning, which can use data feedback to optimize the network selection strategy. Our algorithm considers the problem of resource utilization, the impact of edge device energy consumption and the task's QoS requirements. Our algorithm is compatible with edge computing. We have solved some of the problems in the network selection algorithm of heterogeneous wireless network. Our results show that based on our strategy when performing task migration in edge computing, the energy consumption of the terminal can be reduced, and the change of task delay also improves the user experience. Our technology can be widely applied to edge computing environments for large-scale, low-energy edge device deployments. This can effectively reduce the cost of edge computing deployment.

In this paper, Sect. 2 introduces the system model, including the task model and the network model; Sect. 3 provides the detailed algorithm flow of the task migration network selection strategy; In Sect. 4, we verify the effectiveness of the task migration network selection strategy through experimental simulation. We conclude this paper in Sect. 5.

2 System Model

2.1 Task Model

In this work, we use the convolutional neural network YOLOv3 to perform image recognition tasks. YOLOv3 uses the Darknet-53 basic feature extractor, which combines Darknet-19 and deep residual networks, and has 53 convolutional layers. Compared to other more advanced classifiers, Darknet-53 handles fewer floating-point operations, is faster, and achieves the highest measured floating-point operations per second. This can make better use of system performance, improve evaluation efficiency, and thus improve recognition speed.

In the experiment of this paper, we select five sets of the coco2014 dataset [7] as our experiment dataset.

2.2 Network Model

The heterogeneous wireless network system is composed of a plurality of heterogeneous network elements, including macrocell, picocell and femtocell [8]. They distinguish between each other by transmission energy, coverage, backbone and transmission characteristics. The heterogeneous wireless network system distributes macrocells within the coverage of the picocells [9]. Inside the macrocell, the picocells are set in the communication hotspot area, and femtocell is randomly constructed by the user in the indoor area using Wi-Fi.

We use 4G and Wi-Fi to build heterogeneous wireless networks. There are two industry-developed technologies, LTE-Advanced developed by 3GPP and WirelessMAN-Advanced developed by IEEE. In China, the 4G technologies adopted by China Unicom and China Mobile are mainly LTE TDD and LTE FDD. We use LTE TDD and LTE FDD in our experiments as 4G standard. LTE Peak download is 100 Mbit/s, and peak upload is 500 Mbit/s.

There are many versions of Wi-Fi, and we are currently using Wi-Fi 4 and Wi-Fi 5 (the leading technologies are IEEE 802.11a/b/g/n and IEEE 802.11ac). The Wi-Fi network we experimented with was also based on these two Wi-Fi standards. Their theoretical maximum data transfer rates are 72–600 Mbit/s and 433–6933 Mbit/s, respectively.

The purpose of constructing a heterogeneous wireless network scenario is that the picocell can better guarantee the communication quality of the hotspot area; the home base station can make the user more convenient to access the network in real time [10].

3 Network Selection Strategy for Task Migration

Before giving the network model, we first introduce the basics of the network model, a model-free reinforcement learning algorithm. Q-learning is to record the side of the learning, so tell the agent what action to take will have the greatest reward value [11]. Q-learning tells us to adopt different strategies in different situations through self-learning which does not require a model of any environment, including agents, environment, rewards, and actions. It abstracts the problem into a Markov decision process [12]. The ultimate goal of the entire process is to find the expectation of the strategy that accumulates the most rewards:

$$\underset{\pi}{max}\ E\left[\sum_{t=0}^{H}\left(\gamma^{t} R(S_t, A_t, S_{t+1})|\pi\right)\right] \tag{1}$$

in the Eq. (1), S define as set of states, A define as set of actions, $R = \left(s, a, s'\right)$ is reward function, H is horizon, γ represents discount factor.

The main advantage of Q-learning is its problem-solving system. By dynamically planning the Bellman equation, each state is valuably determined not only by the current state but also the former state. The value of the current state S can be obtained by expecting the cumulative reward of the previous state $V(s)$:

$$V_{\pi}(s) = E(U_t|S_t = s) \tag{2}$$

we will formulate the Eq. (2) as follows:

$$V_{\pi}(s) = E_{\pi}\left[R_{t+1} + \gamma\left[R_{t+2} + \gamma[\ldots]\right]|S_t = s\right] \tag{3}$$

after finishing we get the following equation,

$$V_{\pi}(s) = E_{\pi}\left[R_{t+1} + \gamma\left[R_{t+2} + \gamma V\left(s'\right)\right]|S_t = s\right] \tag{4}$$

we can obtain the optimal cumulative expectation:

$$V^*(s) = \max_{\pi} E\left[\sum_{t=0}^{H}\left(\gamma^t R(S_t, A_t, S_{t+1})|\pi, s_0 = s\right)\right] \tag{5}$$

and the optimal value action function denoted as:

$$Q^*(s) = \sum_{S'}\left(P\left(s'|s, a\right)\left(R\left(s, a, s'\right), \gamma \max_{a'} Q^*\left(s', a'\right)\right)\right) \tag{6}$$

by (5) and (6), and after iterating over Q, Q denoted as:

$$Q(s, a) = Q(s, a) + \alpha\left[r + \gamma \max_{a'} Q^*\left(s', a'\right) - Q(s, a)\right] \tag{7}$$

Where α is the learning rate, the range is 0 to 1. The larger the learning rate value, the faster the convergence of the whole algorithm, and vice versa; γ is the reward decay coefficient.

A. **System status**

According to the Q-learning algorithm theory, we define the state of the environment as:

$$S := (n, v, p) \tag{8}$$

Our environmental state mainly includes three parameters, the network connection state n, the network transmission rate v under the network connection state n, and the terminal energy consumption p.

B. **Action selection**

In this paper, the action is mainly to change the network status and find a suitable network transmission rate network. We denote a as action, and $A = \{a, a \in \{0, 1, 2, \ldots, N\}\}$ as a collection of all possible actions, the terminal accesses the network $N : a = N$.

C. **Reward function**

The reward function is mainly used to feedback the network selection strategy [10]. We consider the network transmission rate and terminal energy consumption as the evaluation indicators. Our reward is directly proportional to the network transmission rate and inversely proportional to the terminal energy consumption. The reward function R is denoted as:

$$R = \frac{V}{P} \tag{9}$$

Where V is the network transmission rate, and P is the terminal energy consumption.

D. **Algorithm**

Implementation of Network Selection Model Algorithm Based on Q-learning:

(a) Initialization. The weight of the network selection model is initialized, and α and γ in the Eq. (5) are set.
(b) Status input. When the image recognition starts, the task model is used to collect data of the current environmental state, and the current network state and network transmission rate data are collected, and finally, the state s is constructed according to Eq. (6).
(c) Action selection. Select an action with an arbitrary probability ε according to the obtained Q value.
(d) The iterative update. State s executes the action selected in step (c) and then reaches the next state s′, and the state s′ updates the Q value to $Q(\mathrm{s}')$ by the Eq. (7); The update is convenient for the next state to obtain the Q value.
(e) Network Update. After several iterations, the state of the final moment and the corresponding Q value are obtained. Obtain the corresponding option with the highest weight and output it to get the final output result of the network (Fig. 1).

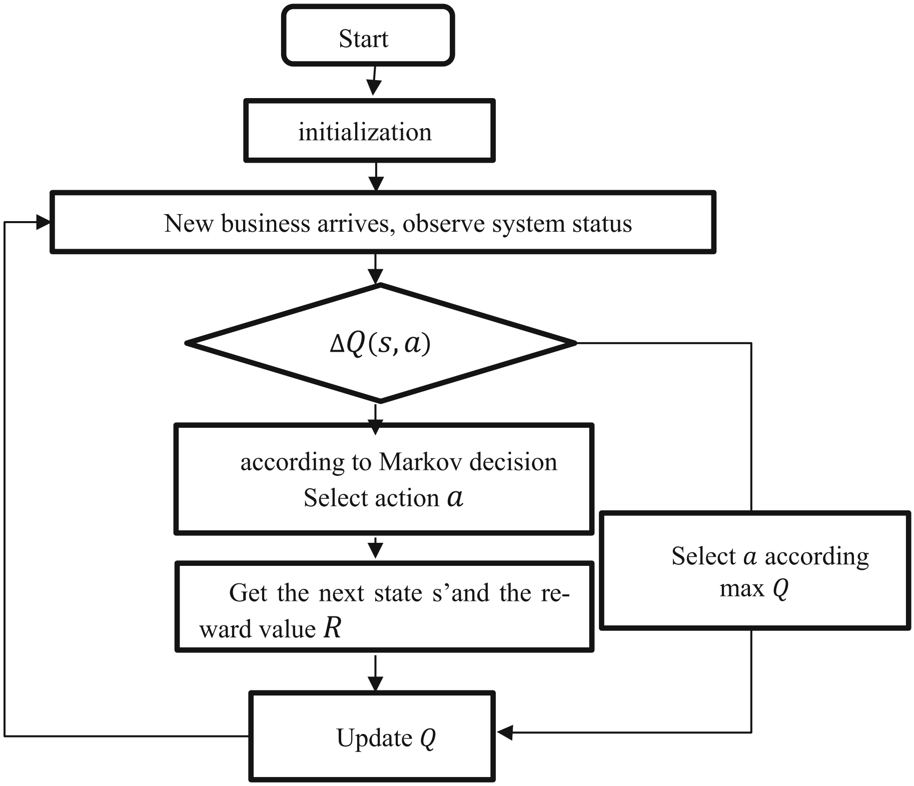

Fig. 1. Q-learning network selection algorithm flow chart

4 Experimental Results and Analysis

4.1 Experimental Platform

Our experimental platform is the Vivo X9 mobile phone. The system has an 8 cores A53 Qualcomm Snapdragon 625 running at 2.0 GHz, and Adreno 506 as GPU. We run the Android 7.1.2. We use YOLOv3 app to perform the local image recognition task test. We deploy YOLOv3 on the host as edge computing server and deploy YOLOv3 app on OnePlus 6 as another device in edge computing. Besides, we obtain the edge device's energy consumption information through the energy test tool written in our lab. We experimented with the network simulation tool Fiddler to simulate the network status of 4G, Wi-Fi and D2D [13].

4.2 Evaluation Methodology

Evaluation Method. We use the comparison experiment to evaluate our algorithm: verify the energy consumption and time-consuming of the image recognition task based on the Q-learning network selection algorithm in the edge computing, and verify the energy consumption and time-consuming of the image recognition task based on the random network selection algorithm in the edge computing. Finally, by comparing the indicators of the two strategies, we can evaluate our algorithm. Specifically, the terminal performs image recognition tasks in edge computing. We partition the images into 5 sets where each of them contains 100 images. We use our algorithm and random selection network algorithm respectively to do experiments. We repeat this process 3 times. Use the energy test tool to collect parameters during the experiment: time, CPU energy consumption.

Performance Report. We report the arithmetic mean and variance in the evaluation report. The arithmetic mean can well reflect the concentration trend within a phase. The variance can well reflect the degree of data fluctuations in a phase. For the system status, we collect three parameters: network connection status, network transmission rate, and CPU energy consumption data. To collect data, we set five different network environments including Wi-Fi (IEEE 802.11ac), 4G (LTE-TDD) and D2D (Bluetooth), then save the network connection status and network transmission rate. We excluded the impact of other apps on the experiment. To measure the energy consumption, we developed a lightweight tool to take read CPU information note running at a frequency of 100 samples per second. Then we matched the time stamps to compute the energy consumption.

4.3 Experimental Results

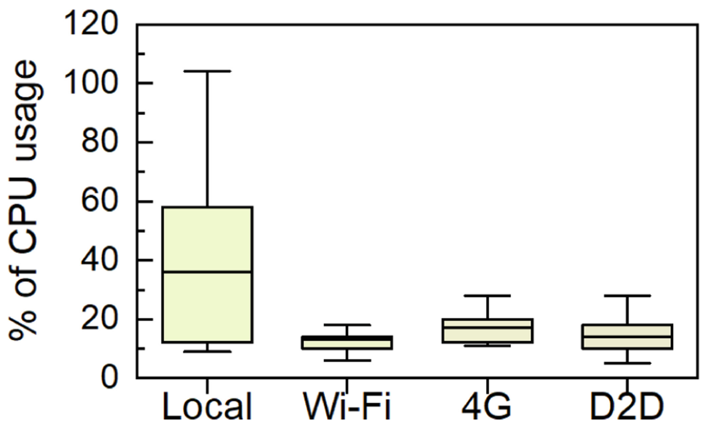

Fig. 2. Performing task CPU performance under different network conditions

In Fig. 2, we compare the performance of the CPU in four cases. Under the condition of performing the same tasks, the local image recognition task has higher requirements on the CPU, especially the energy consumption peak requires the CPU to work in a higher voltage environment. In the edge computing, that is, the task is performed in the environment of Wi-Fi, 4G, and D2D, the energy consumption of the terminal CPU is not high, and the image recognition task can be completed with only a small overhead. This diagram reinforces our works that we should make task migration when the performance of the edge device does not meet the mission requirements. The rest of the discussion in this paper explains how to solve the network selection problem when task migration in edge computing.

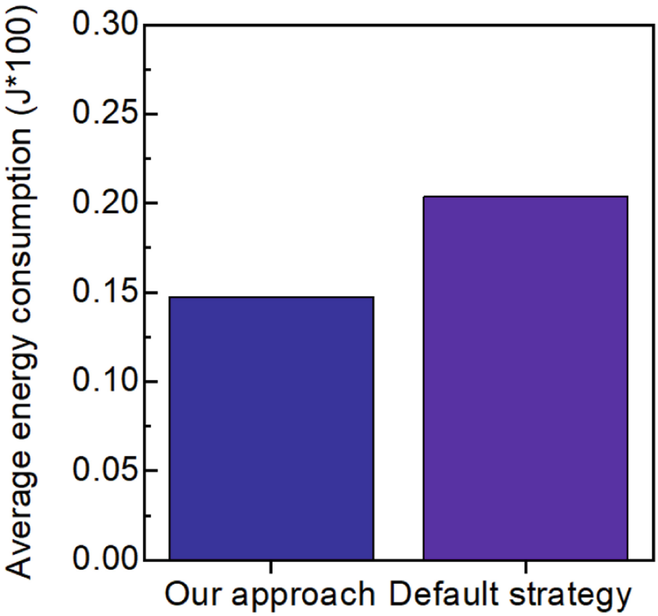

Fig. 3. Edge device energy consumption under two strategies.

Image recognition tasks are sent to the edge computing environment. In the edge computing environment, we use the Q-learning network selection algorithm and the random selection network algorithm for task processing. We collect terminal energy consumption of multiple experiments under the same conditions of the network environment. Figure 3 compares the terminal energy consumption under two strategies. When the Q-learning network selection algorithm is used for task processing, the average energy consumption of the terminal is reduced by 27.79%. The CPU usage of the terminal is reduced, which reduces the waste of resources. It indicates that the edge computing can perform larger tasks and improve resource utilization through the Q-learning network selection algorithm.

Figure 4 calculates the ratio of our approach's improvement in task delay compared to the random network selection algorithm. In short, the network selection task migration algorithm based on Q-Learning effectively reduce the task delay. In the case of the same two policy network environments, the task execution time based on our approach is reduced by an average of 35%. We also calculated variance of the task delay. The results show that the variance of the task execution time using our method is significantly smaller than the default strategy. Our approach is more robust. The result indicates that through

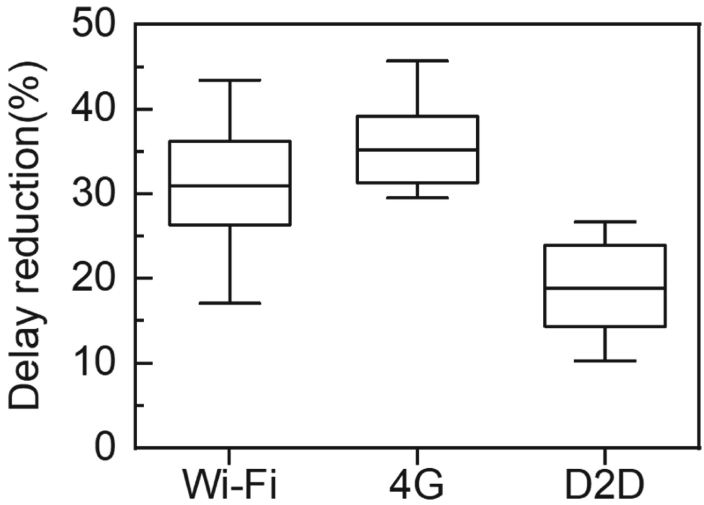

Fig. 4. Comparison of delays between the two strategies.

the monitoring of network transmission rate, our approach can make rational use of resources in the edge computing environment to avoid uneven use of network resources.

5 Conclusion

In this paper, we proposed a task migration based on Q-learning network selection for edge computing in heterogeneous wireless networks. With the proposed algorithm, we achieved nearly 30% energy savings. The performance requirements of the terminal are significantly reduced. At the same time, the delay of the task is also reduced 35%. It indicates that the edge computing based on Q-learning network selection strategy reduces the performance requirements of the terminal in terms of energy consumption, and the task with high delay requirement can also meet its requirements.

Acknowledgement. This work was supported in part by the National Natural Science Foundation of China (Grants nos. 61701400, 61572401 and 61672426), by the Project Funded by China Postdoctoral Science Foundation (Grants nos. 2017M613188 and 2017M613186), by the Natural Science Special Foundation of Education Department in Shaanxi (Grant no. 2017JQ6052), and Natural Science Special Foundation of Education Department in Shannxi (17JK0783).

References

1. Ahmed, A., Ahmed, E.: A survey on mobile edge computing. In: 10th IEEE International Conference on Intelligent Systems and Control (ISCO 2016). IEEE (2016)
2. Ahmed, E., et al.: Application optimization in mobile cloud computing: motivation, taxonomies, and open challenges. J. Netw. Comput. Appl. **52**, 52–68 (2015)
3. Wen, J., et al.: Energy-efficient device-to-device edge computing network: an approach offloading both traffic and computation. IEEE Commun. Mag. **56**(9), 96–102 (2018)

4. Xing, H., et al.: Joint task assignment and resource allocation for D2D-enabled mobile-edge computing. IEEE Trans. Commun. **67**(6), 4193–4207 (2019)
5. Qin, Q., et al.: To Compress, or Not to Compress: Characterizing Deep Learning Model Compression for Embedded Inference (2018)
6. Redmon, J., Farhadi, A.: YOLOv3: An Incremental Improvement (2018)
7. Lin, T.Y., et al.: Microsoft COCO: Common Objects in Context (2014)
8. Zheng, J., et al.: Joint load balancing of downlink and uplink for eICIC in heterogeneous network. IEEE Trans. Veh. Technol. **66**(7), 6388–6398 (2016)
9. Hu, L., Zhou, L.Q., Deng, J., et al.: A strategy of SDN based adaptive multipath load balancing. J. Xi'an Polytech. Univ. **32**(2), 243–247 (2018)
10. Zheng, J., et al.: Joint downlink and uplink edge computing offloading in ultra-dense HetNets. Mob. Netw. Appl. **24**(5), 1452–1460 (2019)
11. Watkins, C.J.C.H.: Technical note: Q-learning. Mach. Learn. **8**, 279–292 (1992)
12. Littman, M.L., Dean, T.L., Kaelbling, L.P.: On the complexity of solving Markov decision problems. In: Proceedings of International Conference on Uncertainty in Artificial Intelligence, pp. 394–402 (1995)
13. He, Y., et al.: D2D communications meet mobile edge computing for enhanced computation capacity in cellular networks. IEEE Trans. Wirel. Commun. **18**(3), 1750–1763 (2019)