

AI-Driven Energy-Efficient Content Task Offloading in Cloud-Edge-End Cooperation Networks

CHAO FANG ^{1,2} (Senior Member, IEEE), XIANGHENG MENG¹, ZHAOMING HU ¹, FANGMIN XU ³,
DEZE ZENG ⁴ (Senior Member, IEEE), MIANXIONG DONG ⁵ (Senior Member, IEEE),
AND WEI NI ⁶ (Senior Member, IEEE)

¹Faculty of Information Technology, Beijing University of Technology, Beijing 100021, China

²Purple Mountain Laboratories, Nanjing 210001, China

³School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

⁴School of Computer Science, China University of Geosciences, Wuhan 430079, China

⁵Department of Sciences and Informatics, Muroran Institute of Technology, Muroran 050-8585, Japan

⁶Data61, CSIRO, Marsfield, NSW 2122, Australia

CORRESPONDING AUTHOR: CHAO FANG (e-mail: fangchao@bjut.edu.cn)

This work was supported by the Beijing Nova Program of Science and Technology under Grant Z191100001119094 and in part by Beijing Natural Science Foundation under Grant L202016.

ABSTRACT To tackle a challenging energy efficiency problem caused by the growing mobile Internet traffic, this paper proposes a deep reinforcement learning (DRL)-based green content task offloading scheme in cloud-edge-end cooperation networks. Specifically, we formulate the problem as a power minimization model, where requests arriving at a node for the same content can be aggregated in its queue and in-network caching is widely deployed in heterogeneous environments. A novel DRL algorithm is designed to minimize the power consumption by making collaborative caching and task offloading decisions in each slot on the basis of content request information in previous slots and current network state. Numerical results show that our proposed content task offloading model achieves better power efficiency than the existing popular counterparts in cloud-edge-end collaboration networks, and fast converges to the stable state.

INDEX TERMS Cloud-edge-end cooperation networks, content popularity, content task offloading, deep reinforcement learning.

I. INTRODUCTION

As beyond 5G and 6G wireless communication technologies rapidly develop, emerging network services represented by virtual reality and 8K video transmission have brought a severe energy efficiency problem and stringent service requirements to the existing Internet [1]. Given that the centralized working paradigm of cloud computing generates huge cross-domain traffic and transmission delay [2], how to achieve green content transmission and meet differentiated service requirements is an urgent issue to solve in heterogeneous cloud-edge-end networks [3].

By enhancing caching and computing capacities at the access networks, edge computing can satisfy users' content requests and improve network energy efficiency [4], [5]. Liu et al. [6] presented an efficient file placement and

distribution scheme to balance energy efficiency, spectral and cache allocation by utilizing content popularity and users' preferences. Vu et al. [7], [8] first discussed the minimal energy problem caused by the backtrip and access links from the perspectives of non-encoded and encoded caching policies, and then optimized energy efficiency by pre-encoded caching while ensuring users' request rate. Xu et al. [9] investigated the impact of the cellular and D2D modes on content distribution, and adaptively selected the content delivery modes to promote energy efficiency by jointly considering transmitter deployment, channel state, transmission coverage and quality of service (QoS). Moreover, a random waypoint solution was proposed to handle the challenges caused by data explosion and reduce energy consumption [10]. Li et al. [11] formulated the energy minimization issue as a

two-stage stochastic mixed integer programming model, and discussed the performance under uncertain content requests. Hassan et al. [12] improved energy and spectral efficiency in multi-access edge computing (MEC)-assisted wireless scenarios by optimizing heterogeneous network resource allocation. Xu et al. [13] proposed an optimal energy saving model in green city systems, which chose the candidate content placement positions to improve content distribution and energy efficiency.

Although edge computing can reduce energy consumption by fast proceeding users' requests and providing their interested contents in access networks, the heterogeneity and limitation of edge resources constrain its service capability. The application of cloud-edge cooperative computing to the Internet to enhance network performance has been widely concerned by academia and industry. Wu et al. [14] designed a novel coded caching framework in cellular networks to minimize power consumption and satisfy users' quality of experience (QoE). Zhang et al. [15] proposed a cloud-edge coordinated content caching mechanism to promote energy efficiency, search accuracy and latency in cyber-physical systems. By introducing the cloud and edge computing paradigms, Yang et al. [16] reduced network delay and power consumption by jointly optimizing computation offloading and content caching. Chen et al. [17] presented a new energy-efficient service model in the cloud-edge-enabled IoT networks by simultaneously considering system runtime, switching, and computation energy of all network participants. Ning et al. [18] proposed an energy-efficient virtual network mapping architecture, which reduced energy consumption and improved sustainability in cloud of things by collaborative edge computing.

To guarantee system QoS in complex and dynamic network environments, the application of machine learning to cloud-edge collaborative networks has been increasingly concerned. Given that advantages of machine learning in QoS and QoE improvement, Dai et al. [19] proposed a novel deep reinforcement learning (DRL) algorithm in multi-access vehicle networks to solve the challenging edge cache problem brought by efficient content delivery and high mobility of vehicles. Javed et al. [20] presented a task-driven intelligent content caching architecture in vehicular edge computing to improve task processing and energy consumption. Kong et al. [21] developed a joint computing and caching system to minimize energy consumption of mobile network operators by utilizing deep deterministic policy gradient policy to make resource allocation decisions. He et al. [22] presented a DRL-based integrated framework to realize green wireless networks by optimizing cache and computation resource allocation. Ning et al. [23] proposed a DRL-based offloading policy in a three-layer vehicle system to improve energy efficiency while meeting the delay requirement. Chen et al. [24] designed a task offloading and caching decision-making strategy by using the deep Q network (DQN) algorithm to optimize transmission power allocation in heterogeneous cloud-edge cooperation environments.

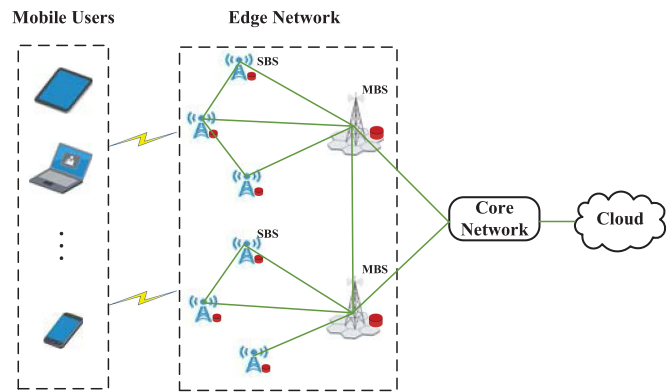


FIGURE 1. Network topology model of heterogeneous cloud-edge-end environments.

Although the existing work in cloud-edge collaboration networks can achieve energy-efficient content distribution, the service capacities of end-users have been largely overlooked. In this paper, we propose a new DRL-based content task offloading solution in cloud-edge-end environments to improve power efficiency.

The main contributions of the paper are as follows.

- We formulate the energy-efficient content task offloading problem as a minimal power model in cloud-edge-end cooperation networks, where requests arriving at a node for the same content can be aggregated in its queue and in-network caching is utilized in the system.
- We propose a new DRL algorithm to minimize power consumption by making collaborative caching and task offloading decisions in each slot according to content request information in previous slots and current network state.
- Evaluation results in different network scenarios show that our proposed content task offloading model achieves better power efficiency than the existing popular counterparts in cloud-edge-end collaboration networks, and fast converges to the stable state.

The rest of our paper is organized as follows. We present the network and power models, and formulated the optimization objective in Section II. In Section III, the power consumption minimization model is solved via the new DQN algorithm. In Section IV, we evaluate the proposed model in heterogeneous network environments and discuss the simulation results. Finally, this study is concluded in Section V.

II. SYSTEM MODEL

In this part, we illustrate network and power consumption models, and formulate the optimization objective in the cloud-edge-end cooperation networks.

A. NETWORK MODEL

The heterogeneous cloud-edge-end network is shown in Fig. 1, which consists of mobile users (MUs), small base stations (SBSs), macro base stations (MBSs), and the cloud. We assume that the MUs and BSs have limited caching

TABLE 1. Notations of Key Parameters.

Notations	Definitions
N_b	Number of BSs
\mathcal{B}	Set of BSs
\mathcal{N}	Set of all the network nodes
\mathcal{N}_t	Slot set of the system
F, \mathcal{F}	Number and set of different network files
$\mathcal{M}(n)$	Set of MUs in slot n
$\mathcal{M}_i(n)$	Set of MUs accessed to i th BS in slot n
\mathcal{A}_i	Set of the nodes directly connecting to node i
\mathcal{B}_i	Set of adjacent network devices of BS i at the same level
Ca_i	Maximal cache limit of node i
C_i	Computing capacity of node i
$q_{i,m}^f(n)$	Number of requests for content f from the m th MU to the i th BS in slot n
$p_{i,j}^{tr,f}$	Power consumption of transmitting from node i to node j
p_i^{com}	Computation power of node i in each CPU cycle
p_i^{ca}	Cache power efficiency about node i
$p_{l_{i,j}}$	Power efficiency about link $l_{i,j}$
P_i^s	Static power consumption of node i
$P_{l_{i,j}}^s$	Static power consumption of link $l_{i,j}$
$P_i^{tr,max}$	Maximal transmit power of node i
s^f	Size of the file f
D^f	Number of CPU cycles required to process the content request f
$Q_i^f(n)$	Boolean variable to indicate whether the content request f is in the processing queue of node i within slot n
$W_i^f(n)$	Boolean variable to indicate whether node i has enough computation capacity to process request f in slot n
$H_{i,m}(n)$	Boolean variable to indicate whether BS i is accessed by the m th MU in slot n
$f_{l_{i,j}}(n)$	Traffic on the link $l_{i,j}$ in slot n
$B_{l_{i,j}}$	Bandwidth of the link $l_{i,j}$

and computing capacities. The BS set is denoted by $\mathcal{B} = \{1, 2, \dots, N_b\}$, where N_b is the number of BSs. The i th BS in the slot n is accessed by a group of MUs $\mathcal{M}_i(n)$. The MUs set in the slot n is denoted by $\mathcal{M}(n)$. $\mathcal{F} = \{1, 2, \dots, F\}$ is the content set, where F is the amount of different files in the system. We assume that all the contents can be obtained from the cloud. In our system, content requests unsatisfied in the MUs are processed in accordance with the sequence of their accessed SBSs, MBSs, and the cloud to fetch the corresponding files. Each node has a request processing queue that aggregates the arriving same content requests, and its aggregated requests are processed only once. In order to reflect the dynamic characteristics of cloud-edge-end environments, the system is modeled under multiple slots.

B. POWER MODEL

In our system, the total power consumption is mainly caused by network nodes and wired links. The notations of key parameters are summarized in Table 1.

1) POWER CONSUMPTION OF MOBILE USERS

The MUs' power is consumed by static operation, signal transmission, request processing, and content caching. The transmit power consumption of the m th MU sending content requests to the accessed base station i at the n th slot is written

as

$$P_{m,i}^{tr}(n) = H_{i,m}(n) \sum_{f \in \mathcal{F}} q_{i,m}^f(n) p_{m,i}^{tr,f}, \quad (1)$$

where $H_{i,m}(n)$ is a boolean variable to indicate whether BS i is accessed by MU m in the slot n . $H_{i,m}(n)$ takes the value of 1 if the m th MU accesses the i th BS, and 0 otherwise. $q_{i,m}^f(n)$ is the amount of network requests to fetch file f , which sent by MU m of the i th base station at the n th slot. $p_{m,i}^{tr,f}$ is the power consumption of the MU m transmitting content request f to the base station i .

The computing power of MU m is consumed in the slot n to process the arriving content requests [25], which is written as

$$P_{i,m}^{com}(n) = \sum_{f \in \mathcal{F}} Q_{i,m}^f(n) W_{i,m}^f(n) D^f p_{i,m}^{com}, \quad (2)$$

where both $Q_{i,m}^f(n)$ and $W_{i,m}^f(n)$ are boolean variables. $Q_{i,m}^f(n)$ is set to 1 if the content request f is in the processing queue of the m th MU accessing the i th BS at the n th slot, and 0 otherwise. $W_{i,m}^f(n)$ takes the value of 1 if the m th MU of the i th BS has enough computation capacity to deal with the content request f at the n th slot, and 0 otherwise. D^f means the amount of CPU cycles, which is consumed to process the content task f . $p_{i,m}^{com}$ refers to the computation power of MU m in each CPU cycle.

The cache power of MU m accessed to BS i is consumed to store files in the slot n can be expressed as

$$P_{i,m}^{ca}(n) = \sum_{f \in \mathcal{F}} X_{i,m}^f(n) s^f p_{i,m}^{ca}, \quad (3)$$

where $X_{i,m}^f(n)$ is a boolean variable indicating whether the MU m caches the content f at the n th slot. $X_{i,m}^f(n)$ takes value 1 if the file f is stored in MU m of BS i in the n th slot, and 0 otherwise. s^f refers to the size about file f . $p_{i,m}^{ca}$ is the storage power efficiency about the MU m at the i th BS.

Therefore, based on [26], the total power consumption of MU m accessed to BS i in the slot n is expressed as

$$P_{i,m}(n) = P_{i,m}^s(n) + P_{i,m}^{com}(n) + P_{i,m}^{ca}(n) + P_{m,i}^{tr}(n), \quad (4)$$

where $P_{i,m}^s(n)$ is the static power of MU m in the slot n to maintain its normal operations.

2) POWER CONSUMPTION OF BASE STATIONS

Similarly, the transmit power consumption of the i th BS transmitting the requested contents to the m th MU in the slot n is written as

$$P_{i,m}^{tr}(n) = H_{i,m}(n) \sum_{f \in \mathcal{F}} q_{i,m}^f(n) p_{i,m}^{tr,f}, \quad (5)$$

where $p_{i,m}^{tr,f}$ is the power consumed by the base station i to transmit the content f to its accessed m th MU.

The computing power consumption of base station i in the slot n is expressed as

$$P_i^{com}(n) = \sum_{f \in \mathcal{F}} Q_i^f(n) W_i^f(n) D^f p_i^{com}, \quad (6)$$

where both $Q_i^f(n)$ and $W_i^f(n)$ are boolean variables. $Q_i^f(n)$ is set to 1 if the request f is in the processing queue of the i th BS at the n th slot, and 0 otherwise. $W_i^f(n)$ is set to 1 if the i th BS has enough computation capacity to deal with the content request f at the n th slot, and 0 otherwise. p_i^{com} is the computation power of the i th BS in each CPU cycle.

The cache power consumed by the BS i in the slot n to store contents can be expressed as

$$P_i^{ca}(n) = \sum_{f \in \mathcal{F}} X_i^f(n) s^f p_i^{ca}, \quad (7)$$

where $X_i^f(n)$ is a boolean variable indicating whether the i th BS stores the file f in the n th slot. $X_i^f(n)$ is set to 1 if the file f is stored by the i th BS at the n th slot, and 0 otherwise. p_i^{ca} is the cache power efficiency of the i th BS.

Thus, the total power consumed by BS i in the slot n is written as

$$P_i(n) = P_i^s(n) + P_i^{com}(n) + P_i^{ca}(n) + \sum_{m \in \mathcal{M}_i(n)} P_{i,m}^{tr}(n), \quad (8)$$

where $P_i^s(n)$ means the static power consumed by BS i in the slot n .

3) POWER CONSUMPTION OF THE CLOUD

The power consumption of the cloud in the slot n is caused by static operating, content caching, and request processing, denoted by $P_c^s(n)$, $P_c^{ca}(n)$, and $P_c^{com}(n)$, respectively. $P_c^s(n)$ and $P_c^{com}(n)$ is expressed as

$$P_c^{ca}(n) = \sum_{f \in \mathcal{F}} s^f p_c^{ca}, \quad (9)$$

$$P_c^{com}(n) = \sum_{f \in \mathcal{F}} Q_c^f(n) D^f p_c^{com}, \quad (10)$$

where p_c^{ca} is the cache power efficiency about the cloud, and p_c^{com} is the computation power of the cloud in each CPU cycle. $Q_c^f(n)$ is a boolean variable, which is set to 1 if the content request f is in the processing queue of the cloud at the n th slot, and 0 otherwise.

Therefore, the power consumed by the cloud in the slot n is written as

$$P_c(n) = P_c^s(n) + P_c^{com}(n) + P_c^{ca}(n), \quad (11)$$

where $P_c^s(n)$ is the static power consumption of the cloud in the slot n .

4) POWER CONSUMPTION OF WIRED LINKS

The power consumption of wired links consists of static and dynamic link power. The total wired link power consumption

about link $l_{i,j}$ in the slot n is expressed as

$$P_{l_{i,j}}(n) = P_{l_{i,j}}^s(n) + f_{l_{i,j}}(n) p_{l_{i,j}}, \quad (12)$$

where $P_{l_{i,j}}^s(n)$ and $f_{l_{i,j}}(n)$ are the static power and traffic generated by link $l_{i,j}$ in the slot n , respectively. $p_{l_{i,j}}$ refers to the power efficiency about link $l_{i,j}$.

C. PROBLEM FORMULATION

Based on the analysis above, the content task offloading issue in cloud-edge-end cooperation networks can be formulated as a minimal power model, which can improve power efficiency by jointly optimizing the computation, cache and communication resources.

$$\begin{aligned} \min \quad & \sum_{n \in \mathcal{N}_t} \left[\sum_{i \in \mathcal{B}} \left(\sum_{m \in \mathcal{M}_i(n)} P_{i,m}(n) + P_i(n) \right) + P_c(n) \right. \\ & \left. + \sum_{i \in \mathcal{B}} \sum_{j \in \mathcal{A}_i} P_{l_{i,j}}(n) \right] \\ \text{s.t.} \quad & C1 : \sum_{f \in \mathcal{F}} X_i^f(n) s^f \leq C a_i, \forall i \in \mathcal{B} \cup \mathcal{M}(n), n \in \mathcal{N}_t \\ & C2 : \sum_{m \in \mathcal{M}_i(n)} H_{i,m}(n) P_{i,m}^{tr}(n) \leq P_i^{tr, \max}, \forall i \in \mathcal{B}, n \in \mathcal{N}_t \\ & C3 : f_{l_{i,j}}(n) \leq B_{l_{i,j}}, \forall i \in \mathcal{N}, j \in \mathcal{A}_i, n \in \mathcal{N}_t \\ & C4 : \sum_{f \in \mathcal{F}} Q_i^f(n) W_i^f(n) D^f \leq C_i, \forall i \in \mathcal{B} \cup \mathcal{M}(n), n \in \mathcal{N}_t \\ & C5 : \sum_{i' \in \mathcal{B}_i} X_i^f(n) \leq 1, \forall i \in \mathcal{B}, f \in \mathcal{F}, n \in \mathcal{N}_t \\ & C6 : X_i^f(n), X_{i,m}^f(n), Q_i^f(n), Q_c^f(n), Q_{i,m}^f(n), W_{i,m}^f(n), \\ & W_i^f(n) \in \{0, 1\}, \forall i \in \mathcal{B}, m \in \mathcal{M}_i(n), f \in \mathcal{F}, n \in \mathcal{N}_t \end{aligned} \quad (13)$$

where \mathcal{N} and \mathcal{N}_t are the node set and slot set of the system, respectively. \mathcal{A}_i is the node set directly connecting to node i , and \mathcal{B}_i is the set of adjacent network devices of BS i at the same level.

In the above constraints, C1 indicates that the cached content size of node i cannot exceed its maximal storage capacity $C a_i$. C2 presents that the sum of transmit power consumed by BS i cannot exceed its maximal transmit power $P_i^{tr, \max}$. C3 means that the sum of network traffic through the link $l_{i,j}$ must be less than its bandwidth $B_{l_{i,j}}$. C4 indicates that the consumed computation resources of node i cannot exceed its computing capacity C_i . C5 presents that the same content will not be cached in its directly connected BSs at the same layer. C6 requires that all the boolean variables are 0 or 1.

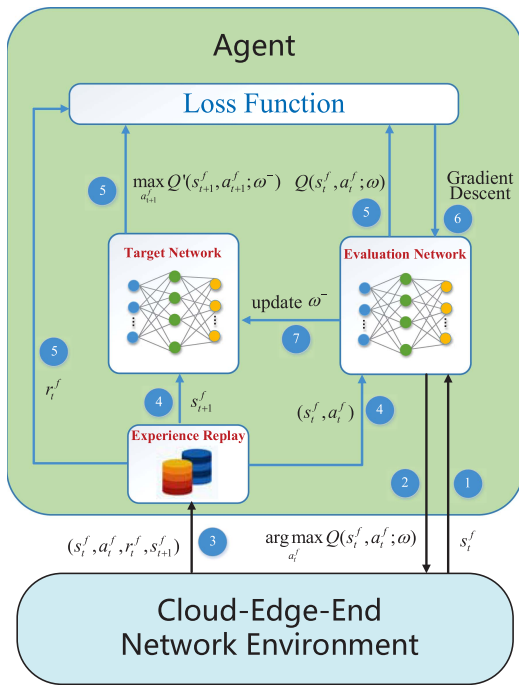


FIGURE 2. The framework and workflow of our deep reinforcement learning.

III. DEEP REINFORCEMENT LEARNING BASED COLLABORATIVE CONTENT TASK OFFLOADING

The optimization problem (13) is a Markov decision process (MDP), which can be solved by the reinforcement algorithms by making intelligent caching and offloading decisions. The MDP model can be defined by a tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{P}(s_{t+1}|s_t, a_t), \mathcal{R}(s_t, a_t)\}$. \mathcal{S} is the set of states, which is a description of the current environment. \mathcal{A} is the set of all possible actions of the MDP. $\mathcal{P}(s_{t+1}|s_t, a_t)$ indicates the probability of transforming from the state s_t to the state s_{t+1} after performing the action a_t . $\mathcal{R}(s_t, a_t)$ represents the received reward when the action a_t is performed under the state s_t . In MDP, the main target of the agent is to find the optimal strategy to maximize the cumulative reward. The Q-learning algorithm can optimize the reward by dynamically obtaining environment state information and storing action values [27]. It is challenging to utilize one table to cache the values of all actions in the complex and dynamic cloud-edge-end cooperation environments [28]. As a branch of the integrated deep learning and reinforcement learning, DQN can overcome the dimensionality disaster problem by using neural networks to automatically obtain low-dimensional features. In this section, a new DQN-based content task offloading policy is designed to improve power efficiency by making collaborative caching and offloading decisions.

Fig. 2 illustrates the working process of the proposed DQN-based content task offloading algorithm. The evaluation and target networks have the same neural network structure with different parameters. In order to improve the system stability, the target network copies network parameters from

the evaluation network to update its own neural network in each specific training cycle. The evaluation network selects an action by an ξ -greedy strategy with the probability of $\xi \in (0, 1)$ or a random action with the probability of $1 - \xi$ according to the known state at time t . During the learning process, the evaluation and target networks randomly extract a set of historical information from an experience replay for training, and modify the related parameters by using gradient descent method.

When our DQN model works, the network state generated by the file request f at the time t is defined as $s_t^f = \{n_t, \mathcal{A}_{n_t}, \mathbf{X}_{n_t}, f_{l_{n_t,j}}, j \in \mathcal{A}_{n_t}\}$. n_t is the node to process the current content request f . \mathcal{A}_{n_t} is the set of adjacent nodes of n_t . $\mathbf{X}_{n_t} = (X_{n_t}^1, X_{n_t}^2, \dots, X_{n_t}^F)$ is the caching state about node n_t . $f_{l_{n_t,j}}$ is the traffic of the link $l_{n_t,j}$. The action for the arriving content request f at the time t is defined as $a_t^f = \{n_{t+1}, n_{t+1} \in \mathcal{A}_{n_t}\}$ to indicate its next hop. The reward obtained by processing the file request f at the time t is $r_t^f = \frac{Y_t^f \beta}{P_t^f} + (1 - Y_t^f)\eta$. Y_t^f is boolean variable, which takes the value of 1 if the content request f is satisfied after action a_t^f , and 0 otherwise. β is a coefficient that adjusts the ratio of power consumption to the reward. P_t^f is the power consumed by a mobile user to send the content request f and obtain the corresponding file at time t . η is the penalty parameter set by the system when the content request f is unsatisfied after action a_t^f . When the current node can satisfy service requirements of the arriving request f , the routing process is terminated and content f is returned to the corresponding end-user. Otherwise, our DQN model will sent the state information s_t^f to the evaluation network to get an action a_t^f , then perform the action to get the reward r_t^f and the state s_{t+1}^f at time $t + 1$. Meanwhile, a piece of historical information $(s_t^f, a_t^f, r_t^f, s_{t+1}^f)$ is stored in the experience replay for the next training. The loss function for the content request f is defined as the mean square error $L(\omega)^f = E \left\{ \left[r_t^f + \gamma \max_{a_{t+1}^f} Q'(s_{t+1}^f, a_{t+1}^f; \omega^-) - Q(s_t^f, a_t^f; \omega) \right]^2 \right\}$, where γ is the discount rate, $Q(s_t^f, a_t^f; \omega)$ is the predicted Q-value generated by evaluation network, $r_t^f + \gamma \max_{a_{t+1}^f} Q'(s_{t+1}^f, a_{t+1}^f; \omega^-)$ is the actual Q-value, and $\max_{a_{t+1}^f} Q'(s_{t+1}^f, a_{t+1}^f; \omega^-)$ is obtained from the target network.

For the content request f from an end user at the time t , the objective of our DQN policy is to choose the optimal offloading decision a_t^f according to state space s_t^f to achieve the maximal reward r_t^f . Specifically, when processing a file request in a slot, a node makes task offloading decisions to minimize power consumption on the basis of its cache state and available adjacent link bandwidth. At the end of each slot, the cache status of each node is updated to improve network performance by utilizing the content request history arriving it.

IV. SIMULATION AND RESULTS

In this section, we present the simulation environments and analyze the numerical results in different cloud-edge-end scenarios.

A. SIMULATION SETTINGS

In this part, our proposed DRL-based green content task offloading scheme is evaluated in heterogeneous cloud-edge-end cooperation networks. We assume that content popularity of the whole system obeys a Zipf distribution, which mainly depends on its skewness factor [29]. A larger value of the skewness factor indicates that more popular content requests are sent by mobile users. In the simulation, the skewness factor varies between 0.6 and 1.5 [30], [31]. Besides, cache size is abstracted as a ratio of the amount of different files cached by a network node to F , and its range is from 0.1% to 1% [32], [33]. We also assume that each network node has a request service queue, and its arrival requests for the same content in a slot will be aggregated in the queue and processed once [34], [35].

To evaluate the advantages of our model, we compare the proposed DRL-based green content task offloading scheme, denoted by “DQN,” with the three existing popular counterparts in cloud-edge-end cooperation networks, denoted by “Without Cache,” “Popularity,” and “LRFU,” respectively. Request aggregation is considered in all the solutions. In “Without Cache,” caches are not deployed in network nodes. In “Popularity,” network files are collaboratively cached among adjacent nodes on the basis of the known whole content popularity distribution [36]. In “LRFU,” the BSs and MUs dynamically update their cached content states according to the historical spatio-temporal request information in each slot [37].

B. RESULTS DISCUSSION

Fig. 3 demonstrates power consumption of all solutions when the cache size of each node varies. When cache size grows, more files that users are interested in are cached in the network, reducing power consumption of the three policies with in-network caching by meeting user requests nearby. Since all the content requests are satisfied by the cloud, the performance of “Without Cache” is unaffected by the varying cache sizes. With the increase of storage capacity, the proposed “DQN” scheme always performs best, and the performance gap between “DQN” and other solutions enlarges. This is because “DQN” can achieve optimal caching and offloading in each slot on the basis of content request information in previous slots and current network state, adapting to dynamic cloud-edge-end networks.

Fig. 4 shows the performance of all schemes when content popularity varies. When the content popularity grows, power consumption of all the policies is decreasing. A larger content popularity means that more mobile users are interested in popular contents, reducing the power consumption of “without cache” because of the improved request aggregation. For the schemes with deployed caching capacity, the promoted cache

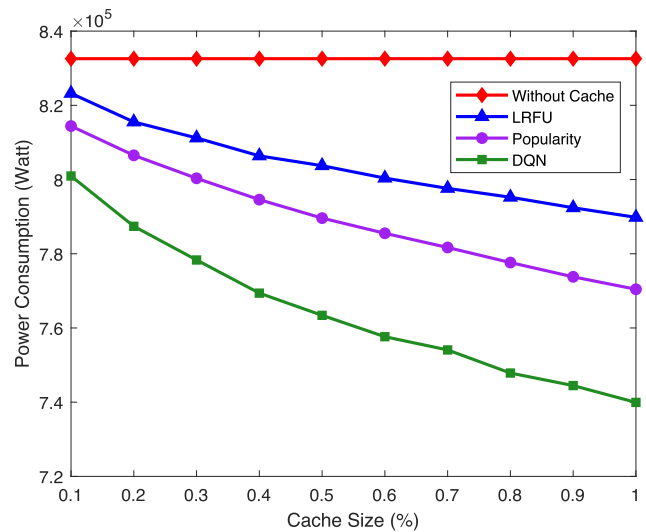


FIGURE 3. Network performance versus cache size.

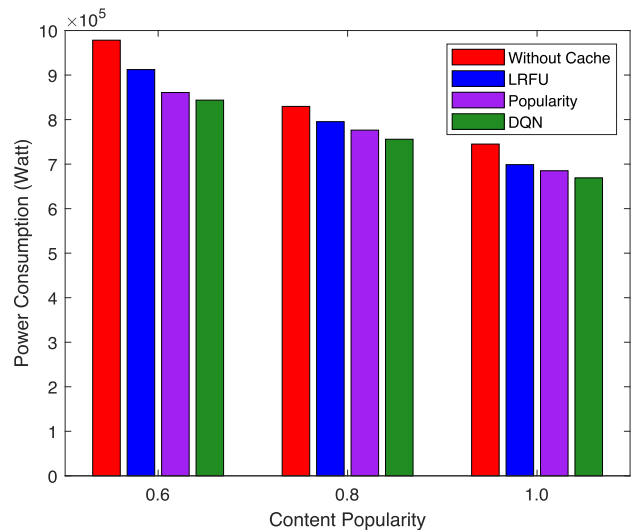


FIGURE 4. Network performance versus content popularity.

hit rate further improves their power efficiency. As the content popularity grows, the performance gap between “DQN” and other schemes is narrowed down. The reason is that more requests for popular contents reduces the relative advantages of our “DQN” scheme in terms of caching efficiency and request aggregation.

Fig. 5 shows power consumption of all solutions when a queue capacity varies. A large queue length can make each network node aggregate more content requests and avoid their redundant transmission, hence improving the performance of the four schemes. As shown in Fig. 5, the performance of “DQN” is worse than those of “Popularity” and “LRFU,” and close to that of “Without Cache” in a small queue capacity. The reason is that the deteriorated request aggregation makes more content requests in the our proposed “DQN” solution first explored and then transmitted to the cloud to obtain their

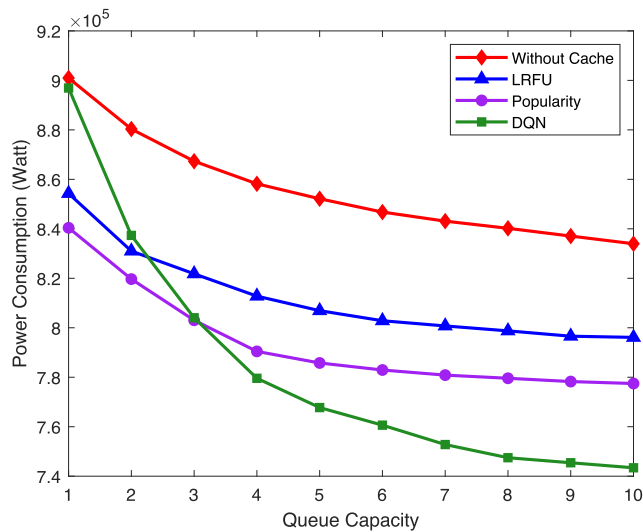


FIGURE 5. Network performance versus queue capacity.

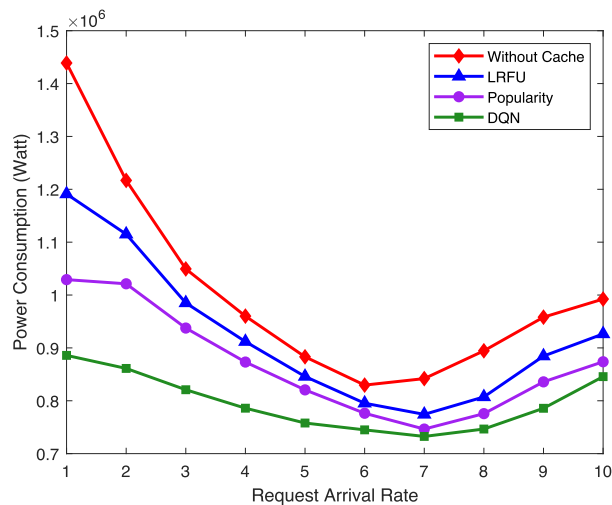


FIGURE 6. Network performance versus request arrival rate.

interesting files, leading to a lower power efficiency. As the queue capacity increases, intelligent caching and offloading decision advantages makes “DQN” perform much better than other solutions.

Fig. 6 demonstrates the performance of all schemes when the request arrival rate varies. The request arrival rate is represented by the number of content requests sent by each user in a slot. As shown in Fig. 6, network power consumption of the four strategies first decreases and then increases when less content requests arrive at a end-user in a slot grows. The reason is that request aggregation in each node is improved when the request arrival rate begins to grow, which leads to gradual reduction of network power consumption. However, power efficiency of all the solutions is deteriorated when the request arrival rate continues to increase. The reason is that a larger request arrival rate in a slot restricts the effect of request aggregation under the limited queue capacity, which

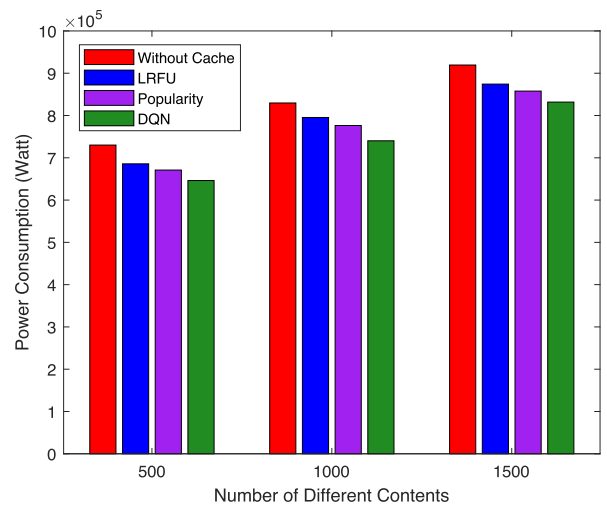


FIGURE 7. Network performance versus content variety.

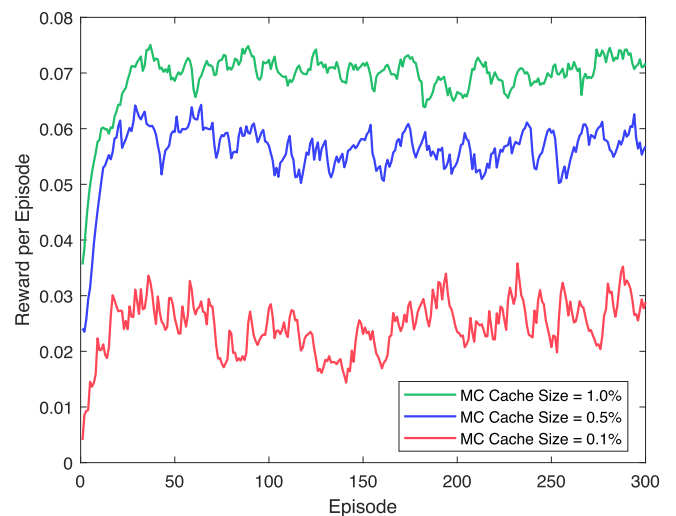


FIGURE 8. Average weighted reward sum versus cache size.

makes more requests route to the cloud to fetch contents and brings about more lost packets. When the request arrival rate varies, “DQN” always performs best by making intelligent caching and routing decisions to be suited for the dynamic cloud-edge-end environments.

Fig. 7 shows power consumption of all schemes when the amount of different network files changes. When content diversity increases, network power consumption of each solution is growing. The growth of the amount of different files indicates that user requests have an obvious diversity phenomenon. Specifically, more unpopular contents and less popular files are accessed by mobile users, which worsens cache hit rate and the effect of request aggregation, and consumes more network power. In this process, “DQN” has the optimal power efficiency by intelligently offloading content tasks according to current network state and historical request information.

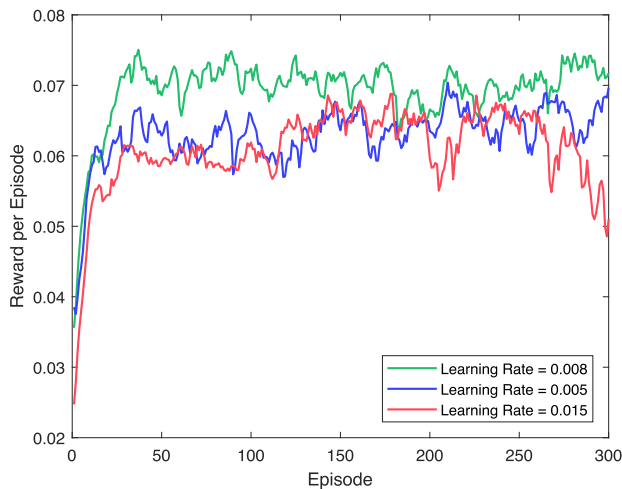


FIGURE 9. Average weighted reward sum versus learning rate.

Figs. 8 and 9 show the average weighted reward for a content request per slot under different cache sizes and learning rates, respectively. As shown in Fig. 8, a larger storage capacity indicates that more popular contents are cached at access networks, hence reducing the power consumption and achieving a convergence state with less fluctuation. In the small cache size scenario, more content requests are satisfied in the cloud, which deteriorates the learning effect of our proposed model in the edge network. A larger learning rate indicates that the old Q value will have a stronger impact on the new one when the system makes cooperative caching and task offloading decisions. As shown in Fig. 9, our “DQN” solution always fast converges under different learning rates, and performs the best when when the learning rate is 0.008.

V. CONCLUSION

In this paper, we proposed a DRL-based green content task offloading scheme in cloud-edge-end environments to realize cooperative caching and computing resource allocation. The energy-efficient content task offloading problem was first formulated as a power minimization model, where requests arriving at a node for the same content can be aggregated in its queue and in-network caching is widely deployed in heterogeneous environments. Then, a new DRL algorithm was designed to minimize the power consumption by making collaborative cache and computation resource allocation decisions on the basis of the predicted spatio-temporal content popularity distribution. Numerical results showed that our proposed content task offloading model achieved better power efficiency than the existing popular counterparts in cloud-edge-end collaboration networks, and fast converged to the stable state.

REFERENCES

[1] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, “Toward low-latency and ultra-reliable virtual reality,” *IEEE Netw.*, vol. 32, no. 2, pp. 78–84, Mar./Apr. 2018.

[2] H. Lu, Q. Liu, D. Tian, Y. Li, H. Kim, and S. Serikawa, “The cognitive internet of vehicles for autonomous driving,” *IEEE Netw.*, vol. 33, no. 3, pp. 65–73, May/Jun. 2019.

[3] A. H. Sodhro et al., “Towards 5G-enabled self adaptive green and reliable communication in intelligent transportation system,” *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 5223–5231, Aug. 2021.

[4] C. Liang, Y. He, F. R. Yu, and N. Zhao, “Energy-efficient resource allocation in software-defined mobile networks with mobile edge computing and caching,” in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2017, pp. 121–126.

[5] S. Bi, L. Huang, and Y.-J. A. Zhang, “Joint optimization of service caching placement and computation offloading in mobile edge computing systems,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 7, pp. 4947–4963, Jul. 2020.

[6] D. Liu, B. Chen, C. Yang, and A. F. Molisch, “Caching at the wireless edge: Design aspects, challenges, and future directions,” *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 22–28, Sep. 2016.

[7] T. X. Vu, S. Chatzinotas, B. Ottersten, and T. Q. Duong, “Energy minimization for cache-assisted content delivery networks with wireless backhaul,” *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 332–335, Jun. 2018.

[8] T. X. Vu, S. Chatzinotas, and B. Ottersten, “Edge-caching wireless networks: Performance analysis and optimization,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2827–2839, Apr. 2018.

[9] Y. Xu and S. Wang, “Mode selection for energy efficient content delivery in cellular networks,” *IEEE Commun. Lett.*, vol. 20, no. 4, pp. 728–731, Apr. 2016.

[10] J. Xu, K. Ota, and M. Dong, “Saving energy on the edge: In-memory caching for multi-tier heterogeneous networks,” *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 102–107, May 2018.

[11] L. Li, D. Shi, R. Hou, R. Chen, B. Lin, and M. Pan, “Energy-efficient proactive caching for adaptive video streaming via data-driven optimization,” *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5549–5561, Jun. 2020.

[12] M. Z. Hassan, M. J. Hossain, J. Cheng, and V. C. Leung, “Energy-spectrum efficient content distribution in fog-RAN using rate-splitting, common message decoding, and 3D-resource matching,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 4929–4946, Aug. 2021.

[13] Q. Xu, Z. Su, Q. Zheng, M. Luo, and B. Dong, “Secure content delivery with edge nodes to save caching resources for mobile users in green cities,” *IEEE Trans. Ind. Informat.*, vol. 14, no. 6, pp. 2550–2559, Jun. 2018.

[14] X. Wu, Q. Li, X. Li, V. C. Leung, and P. Ching, “Joint long-term cache updating and short-term content delivery in cloud-based small cell networks,” *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 3173–3186, May 2020.

[15] P. Zhang, X. Li, D. Wu, and R. Wang, “Edge-cloud collaborative entity state data caching strategy toward networking search service in CPSs,” *IEEE Trans. Ind. Informat.*, vol. 17, no. 10, pp. 6906–6915, Oct. 2021.

[16] X. Yang, Z. Fei, J. Zheng, N. Zhang, and A. Anpalagan, “Joint multi-user computation offloading and data caching for hybrid mobile cloud/edge computing,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11018–11030, Nov. 2019.

[17] X. Chen, J. Zhang, B. Lin, Z. Chen, K. Wolter, and G. Min, “Energy-efficient offloading for DNN-based smart IoT systems in cloud-edge environments,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 3, pp. 683–697, Mar. 2022.

[18] Z. Ning, X. Kong, F. Xia, W. Hou, and X. Wang, “Green and sustainable cloud of things: Enabling collaborative edge computing,” *IEEE Commun. Mag.*, vol. 57, no. 1, pp. 72–78, Jan. 2019.

[19] Y. Dai, D. Xu, Y. Lu, S. Maharjan, and Y. Zhang, “Deep reinforcement learning for edge caching and content delivery in internet of vehicles,” in *Proc. IEEE/CIC Int. Conf. Commun. China*, 2019, pp. 134–139.

[20] M. A. Javed and S. Zeadally, “AI-empowered content caching in vehicular edge computing: Opportunities and challenges,” *IEEE Netw.*, vol. 35, no. 3, pp. 109–115, May/Jun. 2021.

[21] X. Kong et al., “Deep reinforcement learning based energy efficient edge computing for internet of vehicles,” *IEEE Trans. Ind. Informat.*, vol. 18, no. 9, pp. 6308–6316, Sep. 2022.

[22] Y. He, Z. Zhang, and Y. Zhang, “A Big Data deep reinforcement learning approach to next generation green wireless networks,” in *Proc. IEEE Glob. Commun. Conf.*, 2017, pp. 1–6.

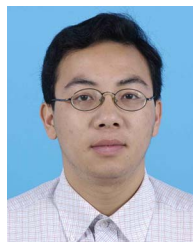
- [23] Z. Ning et al., "Deep reinforcement learning for intelligent internet of vehicles: An energy-efficient computational offloading scheme," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 4, pp. 1060–1072, Dec. 2019.
- [24] Q. Chen, Z. Kuang, and L. Zhao, "Multi-user computation offloading and resource allocation for cloud-edge heterogeneous network," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3799–3811, Mar. 2022.
- [25] U. Saleem, Y. Liu, S. Jangsher, Y. Li, and T. Jiang, "Mobility-aware joint task scheduling and resource allocation for cooperative mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 360–374, Jan. 2021.
- [26] M. Moghimi, A. Zakeri, M. R. Javan, N. Mokari, and D. W. K. Ng, "Joint radio resource allocation and cooperative caching in PD-NOMA-based HetNets," *IEEE Trans. Mobile Comput.*, vol. 21, no. 6, pp. 2029–2044, Jun. 2022.
- [27] Z. Su, M. Dai, Q. Xu, R. Li, and S. Fu, "Q-learning-based spectrum access for content delivery in mobile networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 1, pp. 35–47, Mar. 2020.
- [28] J. Wang, L. Zhao, J. Liu, and N. Kato, "Smart resource allocation for mobile edge computing: A deep reinforcement learning approach," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 3, pp. 1529–1541, Jul.–Sep. 2021.
- [29] N. Golrezaei, A. G. Dimakis, and A. F. Molisch, "Scaling behavior for device-to-device communications with distributed caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 7, pp. 4286–4298, Jul. 2014.
- [30] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. IEEE Conf. Comput. Commun.*, 1999, pp. 126–134.
- [31] N. Choi, K. Guan, D. C. Kilper, and G. Atkinson, "In-network caching effect on optimal energy consumption in content-centric networking," in *Proc. IEEE Int. Conf. Commun.*, 2012, pp. 2889–2894.
- [32] J. Li, B. Liu, and H. Wu, "Energy-efficient in-network caching for content-centric networking," *IEEE Commun. Lett.*, vol. 17, no. 4, pp. 797–800, Apr. 2013.
- [33] H. Xie, G. Shi, and P. Wang, "TECC: Towards collaborative in-network caching guided by traffic engineering," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2012, pp. 2546–2550.
- [34] C. Yi, A. Afanasyev, L. Wang, B. Zhang, and L. Zhang, "Adaptive forwarding in named data networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 3, pp. 62–67, Jun. 2012.
- [35] Y. Chen, D. Li, Y. Hua, and W. He, "Effective and efficient content redundancy detection of web videos," *IEEE Trans. Big Data*, vol. 7, no. 1, pp. 187–198, Mar. 2021.
- [36] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. Leung, "Cache in the air: Exploiting content caching and delivery techniques for 5G systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 131–139, Feb. 2014.
- [37] D. Lee et al., "LRFU: A spectrum of policies that subsumes the least recently used and least frequently used policies," *IEEE Trans. Comput.*, vol. 50, no. 12, pp. 1352–1361, Dec. 2001.



XIANGHENG MENG received the B.S degree from the Faculty of Information Technology, Beijing University of Technology, Beijing, China. His research interests include future network architecture design, Big Data for networking, computational intelligence, mobile edge computing, and machine learning.



ZHAOMING HU received the B.S and M.S degrees in 2021, from the Taiyuan University of Science and Technology, Taiyuan, China, where he is currently working towards the Ph.D. degree with Faculty of Information Technology. His research interests include future network architecture design, software-defined networking, Big Data for networking, computational intelligence, and machine learning.



FANGMIN XU received the M.S. and Ph.D. degrees in communication engineering from the Beijing University of Posts and Telecommunication (BUPT), Beijing, China, in 2003 and 2008, respectively. He is currently an Associate Professor with the School of Information and Communications Engineering, BUPT. From 2008 to 2014, he was with Samsung Electronics, Seoul, South Korea, where he actively contributed to 3GPP LTE/LTE-A and IEEE 802.16m. He has authored two books, 20 peer reviewed international research papers, and 50 standard contributions, and is the inventor of 15 issued or pending patents, among which four have been adopted in the specifications of 4G, 3GPP LTE/LTE-A, and IEEE 802.16m standards. His research interest includes advance technologies in wireless networks, especially the Internet of Things field.

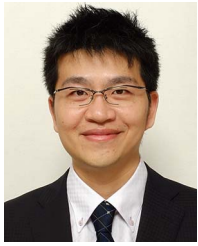


DEZE ZENG (Senior Member, IEEE) received the Ph.D. degree from the University of Aizu, Aizuwakamatsu, Japan, in 2013. He is currently the Full Professor with the China University of Geosciences, Wuhan, China. His research interests include wireless sensor networks, network function virtualization, software-defined networking, cloud computing, and edge computing. He is also on the Editorial Boards of Elsevier JNCA, FCS, IEEE OJ-CS.



CHAO FANG (Senior Member, IEEE) received the B.S degree in information engineering from the Wuhan University of Technology, Wuhan, China, in 2009, and the Ph.D. degree in information and communication engineering from the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China, in 2015. He joined the Beijing University of Technology, Beijing, China, in 2016, and is currently an Associate Professor. From August 2013 to August 2014, he had been

funded by China Scholarship Council to visit Carleton University, Ottawa, ON, Canada, as a Joint Doctorate. He is also a Visiting Scholar with the University of Technology Sydney, Ultimo NSW, Australia, Commonwealth Scientific and Industrial Research Organization, Canberra, ACT, Australia, Hong Kong Polytechnic University, Hong Kong, Kyoto University, Kyoto, Japan, and Murooran Institute of Technology, Murooran, Japan. Dr. Fang is the Vice Chair of Technical Affairs Committee in IEEE ComSoc Asia/Pacific Region (2022–2023). He is the leading Editor of electronics and symmetry special issues. He was the Session Chairs of ICC NGN'2015 and ICC NMNRM'2021, and Poster Co-Chair of HotICN'2018. His research interests include future network architecture design, information-centric networking (ICN), software-defined networking (SDN), big data for networking, mobile edge computing, resource management, and content delivery.



MIANXIONG DONG (Senior Member, IEEE) is currently the Vice President and Professor of Muroran Institute of Technology, Muroran, Japan. He was the recipient of NISTEP Researcher 2018 from MEXT (one of only 11 people in Japan) in recognition of significant contributions in science and technology, Hokkaido Science and Technology Incentive Award 2019, and the Young Scientists' Award 2021 from MEXT. He is the Program Officer of JST Support for Pioneering Research Initiated by the Next Generation. He is also a Clarivate Analytics 2019, 2021 Highly Cited Researcher (Web of Science) and a Foreign Fellow of The Engineering Academy of Japan.

ivate Analytics 2019, 2021 Highly Cited Researcher (Web of Science) and a Foreign Fellow of The Engineering Academy of Japan.



WEI NI (Senior Member, IEEE) received the B.E. and Ph.D. degrees in electronic engineering from Fudan University, Shanghai, China, in 2000 and 2005, respectively. He is currently the Group Leader with CSIRO, Sydney, Australia, and an Adjunct Professor with the University of Technology Sydney, Ultimo, NSW, Australia, and Honorary Professor with Macquarie University, Macquarie Park, NSW, Australia. Prior to this, he was the Postdoctoral Research Fellow with Shanghai Jiaotong University, Shanghai, China, from

2005 – 2008, Deputy Project Manager with the Bell Labs R&I Center, Alcatel/Alcatel-Lucent from 2005 – 2008, and Senior Researcher with Devices R&D, Nokia from 2008 – 2009. His research interests include stochastic optimization, game theory, graph theory, and their applications to network and security. Dr Ni has been the Chair of IEEE NSW VTS Chapter since 2020, and an Editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS since 2018, Secretary and Vice-Chair of IEEE NSW VTS Chapter from 2015 – 2019, Track Chair for VTC-Spring 2017, Track Co-chair for IEEE VTC-Spring 2016, and Publication Chair for BodyNet 2015. He was Student Travel Grant Chair for WPMC 2014, Program Committee Member of CHINACOM 2014, TPC member of IEEE ICC'14, ICC'15, EICE'14, and WCNC'10.