IEEE TRANSACTIONS ON
EMERGING TOPICS IN COMPUTING

Received 28 August 2017; revised 5 January 2018; accepted 6 February 2018.
Date of publication 13 February 2018; date of current version 8 September 2020.

Digital Object Identifier 10.1109/TETC.2018.2805718

Green Resource Allocation Based on Deep Reinforcement Learning in Content-Centric IoT

XIAOMING HE, KUN WANG, (Senior Member, IEEE), HUAWEI HUANG, (Member, IEEE), TOSHIAKI MIYAZAKI, (Senior Member, IEEE), YIXUAN WANG, AND SONG GUO, (Senior Member, IEEE)

X. He and Y. Wang are with the Jiangsu Engineering Research Center of Communication and Network Technology, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210003, China
K. Wang is with the National Engineering Research Center of Communications and Networking, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210003, China, and with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
T. Miyazaki is with the School of Computer Science and Engineering, The University of Aizu, Aizu-Wakamatsu City 965-8580, Japan
S. Guo and H. Huang are with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong
CORRESPONDING AUTHOR: K. WANG (kwang@njupt.edu.cn)

ABSTRACT In the era of information, the green services of content-centric IoT are expected to offer users the better satisfaction of Quality of Experience (QoE) than that in a conventional IoT. Nevertheless, the network traffic and new demands from IoT users increase along with the promising of the content-centric computing system. Therefore, the satisfaction of QoE will become the major challenge in the content-centric computing system for IoT users. In this article, to enhance the satisfaction of QoE, we propose QoE models to evaluate the qualities of the IoT concerning both network and users. The value of QoE does not only refer to the network cost, but also the Mean Opinion Score (MOS) of users. Therefore, our models could capture the influence factors from network cost and services for IoT users based on IoT conditions. Specially, we mainly focus on the issues of cache allocation and transmission rate. Under this content-centric IoT, aiming to allocate the cache capacity among content-centric computing nodes and handle the transmission rates under a constrained total network cost and MOS for the whole IoT, we devote our efforts to the following two aspects. First, we formulate the QoE as a green resource allocation problem under the different transmission rate to acquire the best QoE. Then, in the basis of the node centrality, we will propose a suboptimal dynamic approach, which is suitable for IoT with content delivery frequently. Furthermore, we present a green resource allocation algorithm based on Deep Reinforcement Learning (DRL) to improve accuracy of QoE adaptively. Simulation results reveal that our proposals could achieve high QoE performance for content-centric IoT.

INDEX TERMS Green resource allocation, QoE, content-centric computing, IoT, deep reinforcement learning

I. INTRODUCTION

Closely connected with the improvement of wireless network and development of smart devices, large numbers of multimedia and other services with high transmission rate of data are coming true, which leads to great interests to construct novel wireless networks [1]. One promising direction is the deployment of content-centric computing system, which can provide better content delivery for users than other systems [2], [3]. Meanwhile, so as to meet the growing service needs for IoT users, new cache allocation mechanisms are required to improve the current contents utilization.

At the same time, the level of appreciation is guaranteed of Quality of Experience (QoE) for different services is an important issue [4]–[7] of future wireless networks. Various services will be offered to IoT users. The success of any kind of service bases on not only the performances of the novel wireless networks but also the Mean Opinion Score (MOS) [8] of users. Therefore, both the academia and industry have turned their attention from the IoT Quality of Service (QoS) [9] parameters including throughput, jitter, packet loss, and delay to the QoE concept.

The International Telecommunication Union (ITU-T) has defined QoE as *the whole thing of acceptability of the services* subjectively perceived by the IoT users. According to the definition of the European Qualinet society, QoE is the *degree* of being happy or annoying for IoT users of the services. This is because the utility and/or the expectations of the services are implemented based on the users personality and current situation [10]. In summary, the common understanding of QoE is largely not the different: QoE is a new indicator for IoT where the indicator is determined by the qualities of any parameters.

Recently, many QoS-based approaches that aim to optimize the overall system performance have been proposed [11]–[13]. Although QoS parameters work well in offering a perfect objective indicator, they could not affect the perceived quality by network conditions and users directly. This may result in a waste of network resources. On the other hand, QoE can directly reflect the user's satisfaction [14]. In contrast to QoS, QoE can indicate both the performance of services, and the subjective ideas of users. Therefore, QoE is more suitable for IoT than QoS in terms of experience. Furthermore, QoE-driven ways can adaptively allocate the limited resources directly in order to enhance the perceived quality by users, and aim at reducing the waste of resources.

Many research efforts have been devoted to QoE-driven resource allocation [15], [16]. Previous studies found solutions to these QoE-driven optimization issues [15] relying on the mixing plane optimization architecture. Researchers have considered the main factors depending on the differences of planes and optimized the function of QoE-maximization jointly. Some researches also studied the approximate dynamic allocation method [16] for resource allocation based on QoE. They've got some close-to-optimal solutions of resource allocation. However, QoE model and QoE-driven optional algorithms for green solutions [17] in content-centric IoT have not received much research attentions.

Motivated by the facts mentioned above, we investigate novel QoE-driven download resource allocation algorithms for green solutions in content-centric IoT. Furthermore, a dynamic approach is proposed to calculate a more close-to-green resource allocation practically than [16]. Besides, a high-accurate allocation algorithm is proposed to manage the IoT from experience, and improve the close-to-green solution based on Deep Q-learning. A network topology is used to evaluate the influence of all kinds of factors and performances of the proposed algorithms. Although [15]–[20] present some related work about cache allocation and preliminary result, a much more suitable model and methods for QoE have been added in this paper. Some major contributions of this paper are summarized as the follows:

- In the context of content-centric IoT. In order to assess the whole quality of the whole IoT, we propose a model that aims to improve the satisfaction of the current IoT conditions based on QoE. Through a comprehensive consideration of both network cost and user experience, we strive to seek a trade-off between the cost of network resource and MOS of users.
- To improve the QoE of the content-centric IoT, through cache allocation and increasing MOS, we devise three algorithms that yield green solution for our proposed QoE-maximization problem. Resource allocation based on shortest path tree (SPT) algorithm is designed to obtain the close-to-green solution for QoE, then based on the centrality node, resource allocation based on the SPT algorithm with centrality is designed to reduce the complexity while guarantee the close-to-green solution, furthermore, resource allocation based on deep Q-learning algorithm is provided to give a closer-to-green solution than former algorithms.
- We also explore extensive factors that affect the performance of QoE such as content popularity characteristic, weight of evaluation, etc., by extensive simulations.

The reminder of this paper is organized as follows. Section II introduces the related work. A detailed model of QoE is presented in Section III, and we formulate problem of QoE-maximization in Section IV. Section V proposes two algorithms to obtain close-to-green solution. Furthermore, Section VI proposes a algorithm based on deep Q-learning to get accurate result compared to the previous two algorithms. In Section VII, some simulations are performed to analyze the QoE of whole IoT. Finally, in Section VII, we draw a conclusion of the paper.

II. RELATED WORK

This section is classified into three categories: (1) QoE; (2) green resource allocation mechanisms in content-centric IoT; and (3) major approaches for multi-cache networks.

A. QOE

The general understanding of QoE is largely the same: QoE is a novel indicator for the services where this indicator is determined by the whole quality in all environments. Thus, QoE has been applied to various scenarios. For instance, Su *et al.* [2] proposed a two-step QoE modeling way and introduced a data-driven framework for 5G networks to enhance personalized QoE and to grasp the intension of the relation between users and services. Rahman *et al.* [21] proposed a rate adaptive algorithm to improve video quality and guarantee a promised QoE by observing the available throughput and managing the playback buffer. Tran *et al.* [22] proposed a novel QoE-driven energy-aware multi-path content delivery approach for mobile phones. Qian *et al.* [23] introduced video broadcasting technologies to support a wide variety of multimedia devices interacted with video contents, reaching a heterogeneous QoE.

B. GREEN RESOURCE ALLOCATION IN CONTENT-CENTRIC IOT

A number of studies [24]–[30] have recently checked the performance of the cache in content-centric IoT. These works show simulation studies widely and offer a lot of insights about performance modeling [25]–[28], deployment motivation [24], [29], and cache nodes [30]. Homogeneous deployments are considered by all of them, where all nodes have cache with the same size. However, these uniform deployments are not the same. In contract, there are a variety of node types, which makes the network deployment strategies different for locations of suitable sized-caches. In fact, these deployments which are of homogeneority and are not the same as each other.

Therefore, we can find that different node types have their network deployments of appropriately sized-caches in the strategy of locations.

To the best of our knowledge, only a few existing studies have researched the heterogeneous strategy of green resource allocation issues [31], [32]. Zhou *et al.* [31] studied the cache allocation problem in content-centric IoT at first. The issues proposed by them are that more cache capacity should be deployed among the core nodes for green solutions rather than the edge nodes, and summarized that advantage caused by heterogeneity of sized-cache is very limited actually. By contrast, a work [32] draws the opposite conclusion later which finds that the larger caches at the edge and the more effect we have. Similarly, Gao *et al.* [33] suspected the needs of ubiquitous caching and draw a conclusion that most of the quality benefits can be obtained by caching at edge alone.

Our green allocation algorithm is different from the one shown in [16], since we select the cache location via a centrality indicator dynamically. The number of cache allocated to nodes is calculated further based on a greedy method, rather than the calculation in direct proportion to the centrality indicators of nodes as shown in [16]. In view of previous work, some studies have not considered the QoE of content-centric IoT. Our work will bring opinions on the summing-ups of previous work, as well as the green resource allocation for QoE.

C. MAJOR APPROACHES FOR MULTI-CACHE NETWORK

Typically, the Non-dynamic Algorithms (N-DAs) of content placement are used to organize the process of cache allocation, i.e., by settling the placement problem of facility. Extensive research has been done in different areas. The Un-capacitated Facility Location (UFL) and Capacitated Facility Location (CFL) [34] are two general methods, where the node capacity refers to the maximum number of users who can be provided with services simultaneously, rather than the capacity of cache.

Liu *et al.* [35] used a tree topology to formulate a standardized UFL problem of web caching and proposed the greedy method as well as dynamic programming algorithm to obtain the close-to-green solution. The minimum of the remaining traffic flow is their objective. Nikolaou *et al.* [36] also solved the problem of coordinated web caching. They used a model with a central controller to show that it could be obtained without pre-positioning and pre-fetching. Nevertheless, the methods mentioned above denote the content chunks in the network as an undivided list of table commodity. This assumption is not set up in content-centric IoT because the content is divided into chunks.

A two-step algorithm in [37] was used to resolve UFL problem of the multi-commodity with constrained capacity in trees. What is more, it offers approximate solutions for general graphs. Authors in [38] and [39] added the additional constraints of bandwidth-links in the circumstance of IPTV networks and CDNs. However, these previous works only focus on a certain constraint. It is not adaptive to a content-centric IoT, where many factors exist. For instance, most of the close-to-green solutions in previous environment assume to adopt off-path caching [38], which is dependent on pre-fetching of file. On the other hand, the related works use fixed topologies [39], which limits the performance.

What we focus in the paper is that the quality of content-centric IoT. Then, we propose a QoE model of green resource allocation and study the influence factors of QoE.

III. NETWORK MODEL AND ASSUMPTIONS

A. SYSTEM MODEL

A content-centric computing system in IoT will be modeled as an undirected graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of content-centric computing nodes, and $\mathcal{E}$ is the set of network links. In the context, every IoT user is linked to a node. Simplistically, we do not distinguish the users from the content-centric computing nodes they are linked to. We assume that Forwarding Information Entry (FIE) can be found in each content item. That is to say, in most environments, only one destination exists for each content chunk [40] from the one user's perspective.

To better understand the symbols and variables in Table 1, we use Figure 1 to illustrate the IoT model. In Figure 1, the ith content chunk (denoted by $f^i \in \mathcal{F}$, $i \in \mathcal{N}$) associates with a single node $v_s^i \in \mathcal{V}$ as its source service-node, which can be also indexed by the function $s(f^i) = v_s^i$. We define the minimum operating unit as a content chunk for cache management, which is called a entry of cache hereafter. We also assume that it is normal that all content chunks have no differences in their size. When the requested for f^i is sent from source service-node v_t^i to users. We use the term $\langle s(f^i), v_t^i\rangle = \langle v_s^i, v_t^i\rangle = \{v_s^i, \ldots, v_t^i\}$ to denote the forwarding process of data packet, where v_t^i and v_s^i are the content-centric computing nodes of requested users and source service-node.

For any intermediate node, it can meet the interests with a requested cached chunk. If node v_a^i is the first node with a cached copy of the desired content, then the data forwarding path $\{v_s^i, \ldots, v_t^i\}$ will be shorten to $\{v_a^i, v_t^i\}$ and the $b^i_{a\to t}$, $i \in \mathcal{N}$, $\{a, t\} \in \mathcal{V}$, denotes the forwarding path from v_a^i to v_t^i. The cost of network is related to the IoT conditions. Different experiences are incurred by different conditions. We denote the transmission rate of content chunk by $R^i_{a\to t}$, $i \in \mathcal{N}$, $\{a, t\} \in \mathcal{V}$. We also assume that content-centric computing nodes in the network are homogeneous and $R^i_{a\to t}$, $i \in \mathcal{N}$, $\{a, t\} \in \mathcal{V}$, is same among content-centric computing nodes under a certain condition. However, transmission rate $R^i_{a\to t}$, $i \in \mathcal{N}$, $\{a, t\} \in \mathcal{V}$, may be different under different IoT conditions. It will bring the users' distinguishing experiences. We define the experience via specific MOS [8]. Note that the network cost in our green allocation algorithm is measured by the changeable IoT conditions. The MOS of users in our algorithm is measured by $R^i_{a\to t}$, $i \in \mathcal{N}$, $\{a, t\} \in \mathcal{V}$.

B. QOE MODEL

- *The indicator of QoE level*

Aiming at achieving the total experience brought by the whole content-centric IoT, we define QoE as the quality

TABLE 1. Important notation.

G	undirected graph
$\mathcal{V}$	set of content-centric computing nodes
$\mathcal{E}$	set of physical links
$\mathcal{F}$	set of content chunks
$\mathcal{M}$	set of services
N	size of $\mathcal{F}$, $N = \lvert\mathcal{F}\rvert$
$\mathcal{N}$	$\mathcal{N} = \{1, 2, 3, \ldots, N\}$
v_s	content-centric computing node which is source service-node, $v_s \in \mathcal{V}$, $v_s^i = s(f^i)$, $i \in \mathcal{N}$
v_a^i	nearest content-centric computing nodes which can offer the content chunk f^i to requested user, $v_a^i \in \mathcal{V}$, $i \in \mathcal{N}$
f^i	ith requested content chunk, $f^i \in \mathcal{F}$, $i \in \mathcal{N}$
$s(f^i)$	an index function that returns the source service-nodes of content chunk f^i, $i \in \mathcal{N}$
$s(\mathcal{F})$	set of $s(f^i)$, $i \in \mathcal{N}$
q^i	probability that content chunk f^i is requested, $i \in \mathcal{N}$
x_a^i	binary variable indicating cache f^i on node v_a^i, $i \in \mathcal{N}$
c^i	cache entry allocated for content chunk f^i, $i \in \mathcal{N}$
$b_{a\to t}^i$	forwarding length of path of f^i from node v_a^i to node v_t^i, $i \in \mathcal{N}$
$R_{a\to t}^i$	transmission rate between adjacent nodes, $i \in \mathcal{N}$, $\{a, t\} \in \mathcal{V}$
$g(R_{a\to t}^i)$	forwarding cost between adjacent nodes once, $i \in \mathcal{N}$, $\{a, t\} \in \mathcal{V}$
ξ	a weighting parameter of the QoE between the network cost and user experience
QoE_{Net}	QoE related to Network
QoE_{Uer}	QoE related to Users
MOS_{File}	MOS related to File download service
MOS_{Video}	MOS related to Video Streaming service
MOS_{IPTV}	MOS related to IPTV service
MOS_{VoIP}	MOS related to VoIP service
p^u	probability that uth service is requested, $u \in \mathcal{N}$
ϕ	total QoE as our objective function
R_{min}	minimum transmission rate in content-centric computing nodes
R_{max}	maximum transmission rate in content-centric computing nodes
n	$n = \lvert\mathcal{V}\rvert$
c	set of cache nodes
$\mathcal{Q}$	set of values of k division point of $R_{a\to t}^i$, $i \in \mathcal{N}$, $\{a, t\} \in \mathcal{V}$
$y_{b,a}^c$	binary variable indicating node v_b^i is one of the c optimal locations in the SPT rooted at v_a^i
$Y^{a,c}$	set of $y_{b,a}^c$
Λ	state in DQN
Ω	action in DQN
$QTable$	Q-value in DQN

perceived by network and users coming from overall IoT. On the basis of this study, we will show two ways to assess QoE commonly. The first one is on the basis of the value of QoE which is connected with network. That the smaller the network cost is, the higher network experience is acquired, denoted by QoE_{Net} [41]. The second utilizes the MOS model particularly for different applications, having something with users, denoted by QoE_{Uer} [42]. The higher the MOS is, the higher the user experience becomes. The QoE of the whole IoT is affected by these two terms jointly. The detailed assessment approaches are described in the following.

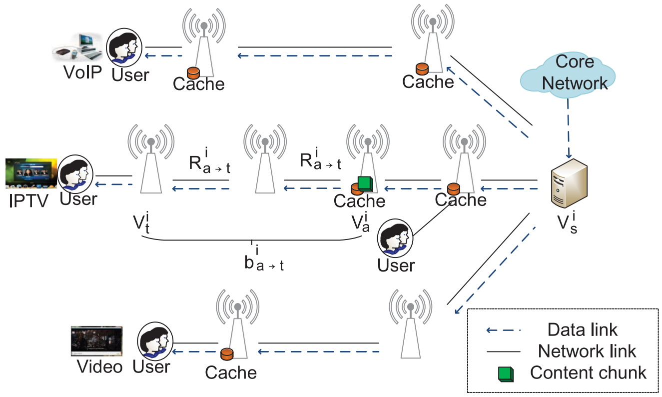


FIGURE 1. Green resource allocation for content-centric IoT.

- *QoE from the side of network*

The network traffic can be seen as the content forwarding among content-centric computing nodes and the length of the path can be defined as one term of cost during T period. When the requested content is not cached in the local node, the request will be forwarded towards to the requested node from the resource node where the content is cached. In a homogeneous network, the content chunk transmission cost between neighbor-nodes can be assumed as the same approximately. Therefore, the network cost can be represented by the cost incurred by forwarding contents, which is related to the length of path and the cost from every transmission between adjacent nodes. We assume the forwarding cost between adjacent nodes is determined by the transmission rate $R_{a\to t}^i$, $i \in \mathcal{N}$, $\{a, t\} \in \mathcal{V}$, and denoted by $g(R_{a\to t}^i)$. Therefore, we can obtain the network QoE in content-centric IoT during T period

$$QoE_{Net}(t) = \sum_{f^i \in \mathcal{F}} \sum_{v_t^i, v_a^i \in \mathcal{V}} q^i \cdot x_a^i \cdot b_{a\to t}^i \cdot g(R_{a\to t}^i), \qquad (1)$$

where x_a^i is a binary variable indicating 1 if content f^i is cached at node v_a^i, and q^i [43] is the probability of content chunk f^i requested by users.

- *QoE from the side of users*

To measure the users' subjective experiences, we utilize the widely-adopted MOS model for different applications. The function based on MOS is seen as the QoE_{Uer} assessment model in existing studies widely. In addition, the relationship between the value of MOS and $R_{a\to t}^i$, $i \in \mathcal{N}$, $\{a, t\} \in \mathcal{V}$, for three services will be detailed later in a certain condition.

File download service, Video streaming service, and VoIP service have been viewed [44] as the three most popular services for IoT users in content-centric IoT. In other words, the three services act as the most important factor for the value of MOS. Different services bring about different values of MOS described as following in different IoT conditions.

(1) *File download service.* The model of logarithmic MOS-throughput has been used in [44] for the file download service, which is not considered to be a real-time service in general. The relationship [44] between the value of MOS and

$R^i_{a\to t}$, $i \in \mathcal{N}$, $\{a,t\} \in \mathcal{V}$, is described as follows:

$$MOS_{File} = \begin{cases} 0.5, & R^i_{a\to t} < 5 \text{ kbps} \\ \alpha \lg(\beta R^i_{a\to t}), & 5 \text{ kbps} \le R^i_{a\to t} < 250 \text{ kbps} \\ 4.0, & R^i_{a\to t} \ge 250 \text{ kbps}, \end{cases} \tag{2}$$

where α and β can be obtained from the upper and lower bound of the user MOS value for file download service, and they are valued to 2.3473 and 0.2667 [44], respectively.

(2) *Video streaming service.* The quality of video streaming service developed by MOS has been influenced by both network and service parameters [45], such as packet loss rate (r_p), send bit rate (r_s), and frame rate (r_f). Therefore, the MOS function for video streaming service can be formulated as

$$MOS_{Video} = \frac{d_1 + d_2 r_f + d_3 \ln(r_s)}{1 + d_4 r_p + d_5 (r_p)^2}. \tag{3}$$

For a known video streaming service, these parameters of data rate are fixed, e.g., d_1, d_2, d_3, d_4 and d_5 are set to -0.0228, -0.0065, 0.6582, 10.0437 and 0.6865 [45], respectively for the IPTV service. Additionally, we also pay attention to the relationship between $R^i_{a\to t}$, $i \in \mathcal{N}$, $\{a,t\} \in \mathcal{V}$, and the value of MOS by fixing r_f at 15 frames per second (fps) and r_p at 0, indicating that there are not any network losses. Thus, the value of MOS for IPTV can be calculated as

$$MOS_{IPTV} = -0.0878 + 0.6582 \ln(r_s). \tag{4}$$

(3) *VoIP service.* The MOS function is measured in the following according to [45] for VoIP service

$$MOS_{VoIP} = \begin{cases} 0 & R^i_{a\to t} < 6.4 \text{ kbps} \\ 3.5, & 6.4 \text{ kbps} \le R^i_{a\to t} < 15.2 \text{ kbps} \\ 3.75, & 15.2 \text{ kbps} \le R^i_{a\to t} < 24.6 \text{ kbps} \\ 4.15, & 24.6 \text{ kbps} \le R^i_{a\to t} \le 64 \text{ kbps} \\ 4.5, & R^i_{a\to t} > 64 \text{ kbps}. \end{cases} \tag{5}$$

We define the set of services as $\mathcal{M} = \{service(File, IPTV, VoIP, \ldots)\}$, QoE_{Uer} as the total MOS utility function determined by the transmission rate $R^i_{a\to t}$, $i \in \mathcal{N}$, $\{a,t\} \in \mathcal{V}$. In practice, the probability distribution over the different services is the Zipf distribution [46], which is frequently observed in real environments. Thus, we have the following notations:

$p^1 = p(MOS_{File}) = p(MOS_1)$,
$p^2 = p(MOS_{Video}) = p(MOS_2)$,
$p^3 = p(MOS_{VoIP}) = p(MOS_3)$,
$\cdots$,
$p^u = p(MOS_{service\ u}) = p(MOS_u)$,

where the probability $p^u = p(\cdot)$ [46] presents the ratio of service-content chunks among the requested content chunks.

As a result, the QoE_{Uer} can be modeled as during T period

$$QoE_{Uer}(t) = \sum_{u\in\mathcal{N}} p^u MOS_u. \tag{6}$$

IV. PROBLEM FORMULATION

QoE is a choice as one of the most crucial performance for the quality of the whole content-centric IoT. As we all know, many factors can affect the QoE [47]. In particular, we consider QoE from two perspectives: experience of users and the network cost. In the perspective of users, we consider that the MOS indicated by the performance of a service is positively proportional to the QoE of users.

Furthermore, the MOS is also related to $R^i_{a\to t}$, $i \in \mathcal{N}$, $\{a,t\} \in \mathcal{V}$, so the $R^i_{a\to t}$, $i \in \mathcal{N}$, $\{a,t\} \in \mathcal{V}$, of each user can not be less than the each group user' minimum rate R_{min} in order to meet demands of IoT users. However, $R^i_{a\to t}$, $i \in \mathcal{N}$, $\{a,t\} \in \mathcal{V}$, can lead to huge cost when it is too large. Hence, the rate $R^i_{a\to t}$, $i \in \mathcal{N}$, $\{a,t\} \in \mathcal{V}$, can not be higher than the maximum rate the predefined R_{max}. In the perspective of network, the network cost is considered as an important factor to QoE. However, the higher the network cost is, the smaller QoE of the network becomes. Thus, the network cost is negatively proportional to QoE. We consider all the above factors in order to improve the total quality in the content-centric IoT. Finally, our object is to find the maximized long-term QoE ϕ and seek the optimal policy Ω for green solutions, so the problem is shown as follows:

$$max \ \arg\phi(\Omega) = -QoE_{Net} + \xi QoE_{Uer} \tag{7a}$$

$$\text{s.t.} \sum_{f^i\in\mathcal{F}} q^i = 1, \ \forall f^i \in \mathcal{F}, \forall i \in \mathcal{N} \tag{7b}$$

$$\sum_{u\in\mathcal{N}} p^u = 1, \ \forall u \in \mathcal{N} \tag{7c}$$

$$R_{min} \le R^i_{a\to t} \le R_{max}, \ \forall i \in \mathcal{N}, \forall\{a,t\} \in \mathcal{V} \tag{7d}$$

$$\sum_{f^i\in\mathcal{F}} \sum_{v^i_a\in\mathcal{V}} x^i_a \le c_{total}, \ \forall x^i_a \in \{0,1\}, \forall v_a \in \mathcal{V}, \tag{7e}$$

where $\xi > 0$ is a weighting parameter between network cost and user experience. Eq. (7b) shows that the sum of the probability must be 1 of content chunk requested by IoT users because it is a certain event. Similarly, the sum of the ratio must be 1 of service-content chunks among the requested content chunks in Eq. (7c). The $R^i_{a\to t}$, $i \in \mathcal{N}$, $\{a,t\} \in \mathcal{V}$, of each user can not be less than the each group user' minimum rate R_{min} in order to meet the demands of IoT users in Eq. (7d). However, the rate $R^i_{a\to t}$, $i \in \mathcal{N}$, $\{a,t\} \in \mathcal{V}$, can not be higher than the maximum rate the predefined R_{max} so as to avoid huge cost. Eq. (7e) shows that in the network the total cache space should be within the capacity c_{total} due to the performance of cache. As $R^i_{a\to t}$, $i \in \mathcal{N}$, $\{a,t\} \in \mathcal{V}$, increases, the $g(R^i_{a\to t})$, will increase as well. During T period, we can consider that the network cost caused by every transmission between adjacent nodes is proportional to the transmission

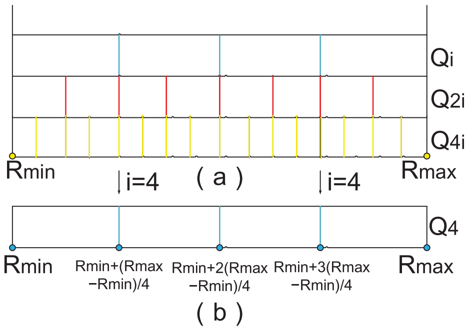


FIGURE 2. The value R_{min} and R_{max} belong to every set $\mathcal{Q}$, $\mathcal{Q}$ includes the i division point in (a). (b) contains 5 elements when $i = 4$.

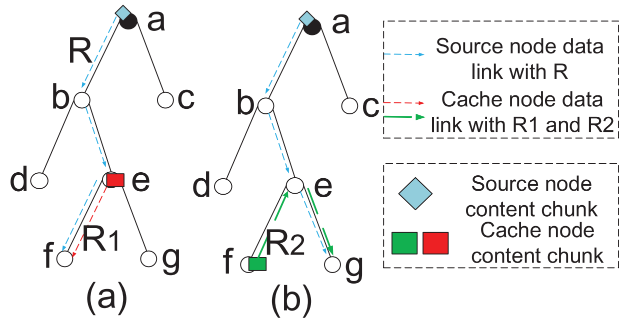


FIGURE 3. An example to illustrate the SPT, which roots from source node a. We assume that it is forwarded to destination f or g through a short path. So the content will be transmitted from node a to node b, then it will be forwarded to node e. Finally, it reaches the destination node f or g. The total cost is the consumption of forwarded times.

rate $R^i_{a\to t}$, $i \in \mathcal{N}$, $\{a, t\} \in \mathcal{V}$. Under this assumption, we can approximate $g(R^i_{a\to t})$ to $kR^i_{a\to t}$, $i \in \mathcal{N}$, $\{a, t\} \in \mathcal{V}$, where k is a coefficient factor.

V. ALGORITHMS DESIGN IN CONTENT-CENTRIC IOT

In this section, we seek optimal policy and design algorithms to solve the QoE-maximization problem as well as offer the green solution. In order to solve the problem (7a), we propose an approach in Figure 2(a) to discretize $R^i_{a\to t}, i \in \mathcal{N}$, $\{a, t\} \in \mathcal{V}$, since the second term of objective function is continuous. Since the range of $R^i_{a\to t}, i \in \mathcal{N}$, $\{a, t\} \in \mathcal{V}$ is $[R_{min}, R_{max}]$, we divide the interval into k parts, each of which is with the same size Z, and use k division points as the discrete values of $R^i_{a\to t}$, $i \in \mathcal{N}$, $\{a, t\} \in \mathcal{V}$. All the division points are included in set $\mathcal{Q}$. For example, 2(b) consists of 5 elements when $\mathcal{Q}_4 (i = 4)$, which are R_{min}, R_{max} and 3 division points. $\mathcal{Q}_4 = \{R_{min}, R_{min} + (R_{max} - R_{min})/4, R_{min} + 2(R_{max} - R_{min})/4, R_{min} + 3(R_{max} - R_{min})/4, R_{max}\}$. We can see that when $k \to \infty$, $Z \to 0$, the values of $R^i_{a\to t}$, $i \in \mathcal{N}$, $\{a, t\} \in \mathcal{V}$, are close to continuous values. Based on this, we propose the resource allocation algorithm to calculate the green solution. When the k is getting bigger, the solution is getting closer to be green. However, the computational complexity will increase accordingly.

We first denote the term $\sum_{\{v^i_t, v^i_a\} \in \mathcal{V}} q^i \cdot x^i_a \cdot b^i_{a\to t} \cdot g(R^i_{a\to t})$ as $cost^i_c$, which is the forwarding cost after allocating content f^i with cache size c^i among nodes in the network. When we use the above method, then the sub-problem is reduced to a knapsack problem [48]. What we should do is to strive the green cache allocation towards the minimum cost.

Hence, the approximate problem of green resource allocation in content-centric IoT is formulated as a general knapsack problem. For a feasible way [46] to get q^i, we assume the probability of requesting content q^i has been known, so we rewrite the objective function of sub-problem as the following:

$$\min \sum_{\{v^i_t, v^i_a\} \in \mathcal{V}} x^i_a \cdot b^i_{a\to t} \qquad (8)$$
$$\text{s.t. (7b) to (7e), and } x^i_a \in \{0, 1\}.$$

Note that Eq. (8) is our main concentration which denotes the total forwarding times of all content chunks between different adjacent nodes. Considering that content-centric IoT develops on-path caching, the network cost relies on the cache locations ($X^i = \{x^i_a, v^i_a \in \mathcal{V}\}$), and the location of the source service-nodes. For simplicity, the assumption of the path packet follows the shortest path for content f^i to source service-node $s(f^i)$ in the network. Thus, $cost^i_c$ indicated by Eq. (8) is the total network cost of nodes which are on the shortest path tree, whose root is $s(f^i)$. For this reason, it is a vital input to the close-to-green solution for the mapping of content f^i and its source service-nodes.

Figure 3 is an example to illustrate the shortest path tree, which roots from source service-node a. When the content is cached at node e, it only needs to be forwarded once in Figure 3(a); when the content is cached at node f, it only needs to be forwarded twice in Figure 3(b). We define the traffic flow is the forwarding process of content among nodes, not only including the traffic caused by content request for node itself, but also the transmission through this node caused by download nodes. The assumption is that every node generates one request content in the network per time unit. Thus, we can calculate the overall traffic flow by summing each node's traffic flow. Then, the cost is equal to the overall traffic flow for content chunk f^i with caching.

As Eq. (8) meets the formulation of knapsack problem, we can separate the problem of green cache allocation in content-centric IoT into two problems: (1) the cache allocation problem in the SPT; and (2) to resolve the whole of Eq. (8) via knapsack problem. In [49], the author optimizes the first problem as a k-means problem and solves it with complexity of $O(cn^2)$. The n denotes the overall number of nodes in the graph and c denotes the number of cache nodes. Typically, we solve the second problem by dynamic programming. A greedy method can solve the knapsack problem of different cache-contents optimally due to the provability of k-means problem in trees is liner non-decreasing concave [49]. An outline of resource allocation based on SPT (R-SPT)) is provided by Algorithm 1, which is used to obtain the close-to-green solution of resource allocation.

Algorithm 1. Resource Allocation Based on SPT

1 **begin**
2   **for** *all* $R^i_{a\to t} \in \mathcal{Q}$ **do**
3   **for** *all* $v^i_s \in s(\mathcal{F})$ **do**
4   **for** *all* $c, 0 < c < n$ **do**
5   Calculate $cost_{s,c}$, and $y^c_{b,a} S(s,c) \leftarrow y^c_{b,a}$, $v^i_b \in \mathcal{V}$
6   **end**
7   **end**
8   **for** *all* $c, 0 < c < n$ **do**
9   **for** *all* $i, 0 < i < N$ **do**
10   Calculate $cost_{a,c}$, $v^i_a = s(f^i)$ $\triangle cost^i_c \leftarrow cost^i_c - cost^i_{c-1}$
11   **end**
12   **end**
13     $c^i \leftarrow 0, \forall f^i \in \mathcal{F}$
14   **while** $\sum_{f^i \in \mathcal{F}} c^i < c_{cost}$ **do**
15       $i \leftarrow argmin_{j|f^j \in \mathcal{F}}(q^j \cdot \triangle cost^j_c)$
16       $c^i \leftarrow c^i + 1$
17   **end**
18     $\phi_R \leftarrow S(s, c, R^i_{a\to t})$
19   **end**
20   $\Omega \leftarrow argmax_{R_m \in \mathcal{Q}}(\phi_{R_m}), R_m \in \mathcal{Q}$
21   $x^i_a \leftarrow y^c_{a,s(f^i)}, v_a \in \mathcal{V}, f^i \in \mathcal{F}$
22   $X = \{x^i_a, R^i_{a\to t} | \forall v_a \in \mathcal{V}, f^i \in \mathcal{F}, R^i_{a\to t} \in \mathcal{Q}\}$
23 **end**

A. RESOURCE ALLOCATION BASED ON SPT

In R-SPT, step 1 (lines 2-7) is the discretization of $R^i_{a\to t}$, $i \in \mathcal{N}$, $\{a, t\} \in \mathcal{V}$, and calculates the cost of cache allocation in the SPT whose root is service-node v_s. Then we calculate the increased cost after allocating the cth, $c = 1, 2, \ldots, n$, cache entry of each content in step 2 (lines 8-13). In step 3 (lines 14-19), we choose the minimum cost increased greedily in $\mathcal{F}$ to allocate the space of cache. In the last step (lines 20-22), we choose the maximum of ϕ and map the state of IoT by $Y^{a,c}$ and C, where $C = \{c^i\}$, $f^i \in \mathcal{F}$, $\forall i \in \mathcal{N}$. The algorithm is worth of noting because it only needs know the distribution of content item's popularity.

The R-SPT outputs the state of IoT for close-to-green resource allocation, including the cache allocation and transmission rate $R^i_{a\to t}$, $i \in \mathcal{N}$, $\{a, t\} \in \mathcal{V}$. The cache placement across content (or nodes) can be presented by summing the columns (or rows) of X. $R^*_{a\to t}$, $\{a, t\} \in \mathcal{V}$ is the value of transmission rate in set of $\mathcal{Q}$, and then we can get the best QoE value. The triple cycle determines the complexity of the above algorithm. In step 3, the greedy method can be completed with a complexity of $O(c_{total} \log N)$ using the max-heap data structure, where N is the number of content items. In the SPT, the complexity for cache allocation is $O(sn^3)$. Therefore, the overall complexity of the problem is max $\{O(lsn^3), O(lc_{total} \log N)\}$, where $s = |s(\mathcal{F})|$ is the number of service nodes, $l = |\mathcal{Q}|$ is the number of the $R^i_{a\to t}$, $i \in \mathcal{N}$, $\{a, t\} \in \mathcal{V}$. Although the Internet-wide deployment with this method is not practical, we can explore the cache performance by referring to this in content-centric IoT.

B. RESOURCE ALLOCATION BASED ON SPT WITH CENTRALITY

Note that it is costly to obtain the overall information with R-SPT since the dispatch or deletion of content is frequent in a dynamic situation.

Based on the observation, we find that the top centrality indicator nodes overlap highly with the cache nodes, and it does not care about the location of root node. That is to say, no matter where the source service-node is located for a content, the top centrality nodes as the candidates are best choice for caching it systematically. A small core exists in every scale-free network, from which almost all other nodes are in the range of distance of $\log\log(n)$ [50].

The centrality of core constituted by nodes is higher than $n^{1/\log\log(n)}$. That proves the structure is almost meshed. In the process of SPT, one of the core nodes will be reached in few hops. Then, from the core node's neighbors, the source service-node will reach the most descendants. Taking the scale-free network with a small diameter into consideration (in the order of $log(n)/\log\log(N)$), some high fan-out nodes may flat the generated SPT. Since the length to the root node is not higher than the topology's diameter, which is relatively small in contrast to the kinds of traffic flow, the network cost after cache allocation of a node is determined by the traffic that passes through it. When the cache is allocated in the nodes, the cost of transmission can be reduced.

Apparently, the amount of traffic for SPT mainly exists in two forms of nodes: (1) the nodes with high output, (2) the other nodes which are between root nodes and high output nodes. The result is consistent with [17], which indicates that compared with the dynamic programming, the greedy method has little loss of performance. In order to find the the largest traffic nodes, we choose the cache location greedily. In the SPT, there are almost not all neighbors of core because the core nodes are fully-meshed. That is to say, caching will not influence the traffic of the others at one of these core nodes. Hence, it is appropriate to take the node with more traffic and high output as candidates when the nodes are selected for green cache allocation.

Due to the above observations, we propose the R-SPT with centrality (R-SPTC). In the algorithm, O_a is the degree of v_a, $cost'_{a,c}$ denotes the cost after cache entries are allocated on the c nodes which have the top centrality in the SPT whose root is node v. In step 1 (lines 2-4), the centrality of every node is calculated. In step 2 (lines 5-10), we calculate the cost after the top c centrality nodes are allocated with cache entries, s_c, $c = 1, 2, \ldots, n$. In step 3 (lines 11-17), the increased cost of the c^{th}, $c = 1, 2, \ldots, n$, cache entry for each content is calculated. In step 4 (line 18-19) , we choose the minimum cost greedily in $\mathcal{F}$ to allocate the cache space. Finally (line 20-24), we select the maximum ϕ and map X using O and C, where $C = \{c^i\}$, $O = \{o^j\}$, $f^i \in \mathcal{F}$, $v^j \in \mathcal{V}$, $\forall i, j \in \mathcal{N}$.

After obtaining the dynamic content allocation, we can sum all content allocation at every node as R-SPT to calculate the corresponding cache allocation. Nevertheless, compared with the R-SPT, R-SPTC explores the optimal content caching by a statistical manner. That is to say, the source service-node for each content is fixed in any IoT conditions. For this reason, the distributed cache replacement policies will beat the exact pre-fetching. The main time consuming steps of DA determine the computational complexity, which are the step 2 and step 4 in the first loop. In step 2, the complexity of calculating the cost of a SPT $cost_c^{'i}$, $c = 1, 2, \ldots, n$, is $O(n)$, $n = |\mathcal{V}|$. Therefore, the $O(n^2)$ is the computational complexity of $cost_c^{'}$, $c = 1, 2, \ldots, n$, is $O(n^2)$. Hence, the total complexity of the DA is max $\{O(ln^2), O(nlN \log(N))\}$. Since the centrality indicator is the node degree, the complexity of calculating the centrality can be not higher than $O(nN \log(N))$.

Algorithm 2. Resource Allocation Based on SPTC

begin
  for *all* v_a, $v_a \in \mathcal{V}$ **do**
    Calculate the O_a of v_a
  end
  for *all* $R \in \mathcal{Q}$ **do**
    for *all* $v_a \in s(\mathcal{V})$ **do**
      for *all* c, $0 < c < n$ **do**
        select top c of O_a calculate $cost_{a,c}^{'}$
      end
    end
    for *all* c, $0 < c < n$ **do**
      $cost_{a,c}^{'} \leftarrow \sum_{v_a \in \mathcal{V}} cost_{a,c}^{'}/n$
    end
    for *all* c, $1 < c < n$ **do**
      $\triangle cost_c^{'i} \leftarrow cost_c^{'i} - cost_{c-1}^{'i}$
    end
    $c^i \leftarrow 0, \forall f^i \in \mathcal{F}$
    while $\sum_{f^i \in \mathcal{F}} c^i < c_{cost}$ **do**
      $i \leftarrow argmin_{j|f^j \in \mathcal{F}}(q^j \cdot \triangle cost_c^{'j})$
      $c^i \leftarrow c^i + 1$
    end
    $\phi_R \leftarrow S(s, c, R_{a\to t}^i)$
  end
  $\Omega \leftarrow argmax_{R_m \in \mathcal{Q}}(\phi_{R_m}), R_m \in \mathcal{Q}$
  for *all* $f^i, f^i \in \mathcal{F}$ **do**
    $S \leftarrow$ set of top c^i centrality nodes
    for *all* v^j, $v^j \in S$ **do**
      $x_j^i \leftarrow 1$
      $R_{a\to t}^i \leftarrow R_t$
    end
  end
  $X = \{x_a^i, R_{a\to t}^i | \forall v_a \in \mathcal{V}, f^i \in \mathcal{F}, R_{a\to t}^i \in \mathcal{Q}\}$
end

VI. DEEP REINFORCEMENT LEARNING-BASED RESOURCE ALLOCATION ALGORITHM IN CONTENT-CENTRIC IOT

Although the R-SPT and R-SPTC have the low complexity, they have the low accuracy and only offer the close-to-green solution. Therefore, in this section, we present a resource allocation algorithm based on Deep Q-learning to manage the network directly from experience, improve the accuracy of QoE that refers to the value of QoE adaptively and get a closer-to-green solution than others.

A. CONSTRUCTION OF MDP IN CONTENT-CENTRIC IOT

According to previous researches, Chen and Han *et al.* [51], [52] develop respective models based on the traditional Markov Decision Processes (MDP) model, consisting of Actions A, Transition Probabilities P, States S and Rewards R, where A and R are determined by the agent, and S comes from observations. The target of MDP model is to learn preferred strategies for different states with respect to the expectation of rewards. Correspondingly, we define the IoT conditions used in our proactive model as

MDP States. The IoT states at the tth allocation interval can be specified by $S(t) = [L(t), \mathcal{Q}(t)] \in S$, where $L(t)$ is the set of conditions of cache nodes, $\mathcal{Q}(t)$ is the set of transmission rates of content chunks. S is the aggregation of IoT states. The allocation interval is the once change of state. At the beginning of each allocation interval, the Internet Service Provider (ISP) does the following preparations, such as collecting the services information, conditions of content-chunks in cache nodes and computes the transmission rates of content chunks, then reallocates the services by mapping the control policy to migration actions. Even though network costs are not revealed clearly expressed in this denotation, their costs can be figured out by the aggregation of services running on them. Therefore, the definition of IoT conditions is comprehensive.

MDP Actions. The assumption can be made that the transmission rate between adjacent nodes is the same as a service in a certain condition, while transmission rate may not be the same in another different condition. A cache node can cache several content chunks. If a content-chunk is cached in a cache node, another content-chunk will be removed from original location in this node. The process of change is denoted as an action. The action set of MDP-based model can be uniquely specified by $\mathcal{A} = \{l_1, l_2, \ldots, l_{C_{c^n}^2}\} \in A$, $\mathcal{B} = \{R_1, R_2, \ldots, R_{C_{|\mathcal{Q}|}^2}\} \in B$. where A and B are the aggregation of all possible actions, which contains approximate $C_{c^n}^2$ elements, $c = 1, 2, \ldots, n$. Under this circumstance, the dynamic programming algorithm is really a great disaster because there are so many elements in state space and action space. Actually, the action set is the same as others for every service. Using the action determination, the ISP will determine how to allocate resources and change the system conditions after allocation intervals.

Rewards. Rewards are incentives that are assigned to tasks after performing action A and B. The aim of this agent is to continuously minimize network cost and receive proper transmission rate and at the same time they also want to make the network stable, by the means of encouraging current state to maximize its final QoE ϕ. If we want to succeed in achieving this goal, what we need to do is assign rewards for actions carefully. The transmission rate $R^i_{a\to t}, i \in \mathcal{N}$, $\{a,t\} \in \mathcal{V}$, must keep balance between the minimum and maximum rate, because they need to meet the demand of their users' service and escape from he huge cost. It is clear that cost saving is partially in conflict with the above constraint conditions. For a joint consideration, we can directly define the reward function the agent receives as the goal of our problem in (7a). i.e.,

$$\begin{aligned} &max\ \phi(t) = -QoE_{Net}(t) + \xi QoE_{Uer}(t) \\ &\text{s.t. } (7b) \text{ to } (7e), \text{ and } x^i_a \in \{0,1\}. \end{aligned} \tag{9}$$

So the MDP agent optimizes a linear combination of revenue which is related to both network and users, and aims at finding a desirable tradeoff between cost and better MOS.

As for transition probability P, Han *et al.* hold the point that the transition kernel can be learnt by maximum likelihood estimation in conjunction with the sliding window scheme under the assumption that demands of every distributed task is not dependent. Nevertheless, whether the translation probability can be computed is up to the condition that the IoT state space is just right. In addition, this way can not succeed when it happens in the wholly new environment, requiring long-drawn training process again to update all related states. Moreover, the value of window size is heuristic to a certain degree. The which may lead to cause inaccurate estimations and a severe error of judgement at the key points.

In consideration of the above issues, we develop our MDP-based model with only states, actions and rewards. Compared with pre-training the transition probability, our agent learns the optimal decision and control strategy in an online fashion. Further parts will be proved in next section.

B. RESOURCE ALLOCATION ALGORITHM BASED ON DQN IN CONTENT-CENTRIC IOT

- *Description of Deep Q-learning*

Deep Reinforcement Learning has been proved to have an excellent performance when solving MDP problems [53]. In fact, Deep Q-Network (DQN) [54] is good option for solving allocation problem in networks. First, decisions in networks are highly repetitive that bring about plenty of date for training the DQN. Second, making decisions by DQN can be implemented in an online stochastic environment. After training more recent data, DQN can be more intelligent. Finally, Q-learning with deep neural networks is flexible to different circumstances as it can capture more characters. Specifically, DQN demonstrates approximation with low asymptotic approximation errors.

We give the concepts of two neural network function approximator with weights θ and θ^- respectively, called estimation networks and target networks. The presentation of two neural networks depends on the overestimation in traditional Q-learning. When updating the Q-values, the estimation error will be transmitted from one iteration to the next due to the max $Q(S^{'}, A^{'}, B^{'})$ operation [54]. To some extent, it seems like a good method to curl the overestimation through delaying the update of target network. In addition, the estimation in DQN is a regression task since the output Q-values can be any real values. Hence it can be trained by the means of reducing a sequence of loss function $L_i(\theta)$ to the minimum to accommodate the approximator. The loss function is defined as follows,

$$\begin{aligned} L_i(\theta) = E_{S,A,B\sim\rho(\cdot)}[(\phi(S,A,B) + \gamma max_{A',B'} Q(S',A',B',\theta^-) \\ - Q(S,A,B,\theta))^2], \end{aligned} \tag{10}$$

where $\rho(S,A,B)$ denotes the behaviour distribution over action A, B and state S, and i is the number of iterations. Note that the target value depends on the network weight θ^-, while the estimation is determined by current weight θ calculated by minimizing the loss function in each iteration. Thus the popular gradient descent method can be used to optimize the loss, and the corresponding differentiation of Eq. (10) with parameter θ, which denotes the current estimation weight is shown as,

$$\begin{aligned} \nabla_\theta L_i(\theta) = E_{S,A,B\sim\rho(\cdot)}[(\phi(S,A,B) + \gamma max_{A',B'} Q(S',A',B',\theta^-) \\ - Q(S,A,B,\theta))\nabla_\theta Q(S,A,B,\theta)]. \end{aligned} \tag{11}$$

where $\rho(\cdot)$ denotes the behaviour distribution over actions and states. In general, the stochastic gradient descent is often adopted to optimize the loss function on account of the large batch size.

The above works have attracted much attention among many researchers. Some deep Q-learning based algorithms are also developed recently, such as DQN with prioritized experience replay (PER) [55], and Double DQN (DDQN) [56].

Overestimation in Q learning is first studied by Thrun, who shows that each target is overestimated up to $\frac{m-1}{m+1}\gamma\epsilon$ if Q-values contain random errors uniformly distributed at $[-\epsilon, \epsilon]$. Modares *et al.* [57] argue that the estimation error of DQN increases with the increasing number of action set. Therefore the overestimation of DQN will greatly degrade the performance of Q-learning.

The remedy for this phenomenon is to change the update of DQN. Modares *et al.* shows a double DQN framework, in which two value functions are learned by different DQN. One is used to decide the action with maximum Q-value, and the other one is used to estimate the value of current state-action pair, the target value H^{DDQN}_k is shown as follows,

$$\begin{aligned} H^{DDQN}_k = \phi(S_k, A_k, B_k) + \gamma Q(S_{k+1}, \\ \arg max_{A,B} Q(S_{k+1}, A, B, \theta_k), \theta^-_k), \end{aligned} \tag{12}$$

where θ and θ^-_k are the weights of estimation network and target network, respectively. Note that these two neural networks are not fully decoupled as the target network is a periodic copy of estimation network. The overestimation in deep

Q-learning, which will damage the performance after each update, and the introduction of double neural networks can stabilize the deep Q-learning.

In a normal DQN, the neural networks will abandon inputting samples immediately after a single update. This memoryless update will result in a loss of experiences, particularly in content-centric IoT, because the frequency of allocation is not really high. Ji *et al.* [56] develop a framework for PER, which replays important transitions more frequently, so that DQN can learn from samples more efficiently than other methods. The main idea is to train the state-action pairs whose Q-values are not suitable for current approximate functions. Intuitively, we prefer to change the model repeatedly until it fits well to most samples. We than define the error ξ_k between target H_k^{DDQN} and estimation $Q(S_k, A_k, B_k)$ as,

$$\xi_k = |\,\phi(S_k, A_k, B_k) + \gamma Q(S_{k+1}, \arg max_{A,B} Q(S_{k+1}, A, B, \theta_k), \theta_k^-) - Q(S_k, A_k, B_k, \theta_k)\,|. \tag{13}$$

Consequently, the probability of sampling transition $p(k)$ can be denoted as,

$$p(k) = \frac{\xi_k^a}{\Sigma_i \xi_i^a}, \tag{14}$$

where a is a factor determining how much prioritization is used. The PER will turn to random replay if $a = 0$. During the PER, finding samples with k highest priority and updating the corresponding priority will take only $O(\leqslant n)$ time if we store the data in the unsorted sum tree. In short, DQN with PER can raise the playback frequency of important historical-data, improving learning efficiency.

- *Resource Allocation Algorithm*

In this section, based on deep Q-learning, we propose a resource allocation algorithm to obtain maximum QoE ϕ. In this algorithm, the agent [54] uses the deep neural network to calculate the correlation between the corresponding Q-values *QTable* and the different state-action pairs Λ and Ω. In fact, the construction phase of the DDQN requires the accumulation of sufficient samples through the algorithm proposed in [58] to complete the training process. In addition, this procedure can also adopt any policy. Samples are deposited in an unsorted sum tree D for PER. Based on the stored samples, DDQN is constructed (lines 1-3).

The inputs contain the states Λ and the actions Ω but the output is the corresponding Q-value *QTable* in Algorithm 3. At the start of each allocation interval t_k, the controller snatches the current IoT state, and selects an action based on the second level exploration with probability $1-\epsilon$ or chooses the action with maximum $Q(S_k, A_k, B_k)$ (lines 4-25). The selected action will be executed and the agent will get corresponding rewards QoE ϕ, with which to build a new 5-variable sample $QTable(\Lambda, \Omega, S_k, A_k, B_k)$ (lines 26-31). In addition, the DQN performs reasoning to derive $Q(S_k, A_k, B_k)$ approximately, which can maximize the Q-value initialization performance (line 32-34). The DDQN will use the PER and update the target network Q' every K steps (line 35-37).

Algorithm 3. Resource Allocation Based on DQN

1 Input: Discount parameter γ, exploration rate ϵ, ν and recovery period K
2 Initialize state table Λ, action table Ω to be empty table; estimation DQN Q with θ, target DQN Q' with $\theta^- = \theta$
3 Pretrain Q and Q' with D
4 **begin**
5   **for** *each allocation epoch* t_k **do**
6   Observe the current IoT state S_k
7   **if** $\Omega_k = \varnothing$ **then**
8   Exe. action $\Omega_k = kStateLearning(S_k)$;
9   Add Ω_k to Ω.
10   **end**
11   Generate random number φ
12   **if** $\varphi > \epsilon$ **then**
13   Generate another random number η
14   **if** $\eta > \nu$ **then**
15   /*rAS = randomActionSelection*/
16                 $A_k = rAS(S_k)$ and $B_k = rAS(S_k)$
17   **end**
18   **else**
19   /*pAS = prioritizedActionSelection*/
20                 $A_k = pAS(\Omega_k)$ and $B_k = pAS(\Omega_k)$
21   **end**
22   **end**
23   **else**
24             $A_k = max_A Q(S_k, A, \theta)$ and $B_k = max_B Q(S_k, B, \theta)$
25   **end**
26   (1) Exe. action A_k, B_k in simulator;
27   (2) Observe QoE ϕ_k and new state S';
28   (3) Exe. $H_k^{DDQN} = \phi_k + \gamma max_{A,B} Q'(S', A, B, \theta^-)$
29   and $\xi_k = |\,H_k^{DDQN} - Q(S_k, A_k, B_k)\,|$;
30   (4) Store transition $(S_k, A_k, B_k, \phi_k, S', \xi_k)$ in D;
31   (5) Exe. $(H_k^{DDQN} - Q(S_k, A, B, \theta))^2$ concerning θ.
32   **if** $k \equiv 0$ *mod* K **then**
33   Sample minibatch of K transitions from D by PER; Reset $Q' = Q$.
34   **end**
35         $updateQTable(\Lambda, \Omega, S_k, A_k, B_k)$
36   **end**
37 **end**

VII. PERFORMANCE EVALUATION

In this section, we will explore the key factors that influence the QoE in content-centric IoT. Besides, the main factors of cache allocation and transmission rate, we will also explore other factors including the content popularity, weight factor between the network cost, the experience of users as well as topology, etc. Furthermore, we will analyze the accuracy of QoE based on the proposed resource allocation algorithm.

TABLE 2. Simulation parameters.

Parameters	Value
BA model	$\gamma = 2.5, m = 2$
Computing node number	200
Service-node number	20
User number	2,000
Hidden layer 1	256 neurons
Hidden layer 2	64 neurons
Hidden layer 3	32 neurons
Hidden layer 4	8 neurons
Discount parameter	0.8
Learning rate	0.2

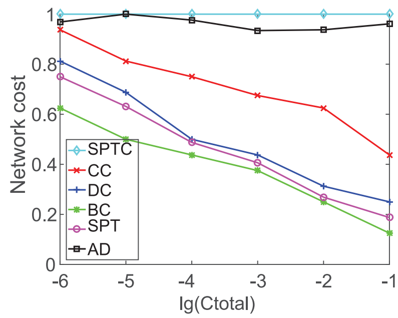


FIGURE 4. Network cost under different schemes while varying $\lg(c_{total})$.

Next, we will use resource allocation algorithm to explore the network condition, which includes the types of network and the cache allocation condition.

A. EXPERIMENT SETTINGS

We use a discrete simulator ccnSim [59] which can model the caching behavior in various graph structures. Then, we import the generated data shown in Table 2 to Matlab, in which configure the simulator consists of topology and request pattern.

In order to explore the factors, we depend on the synthetic generation that allows us change the different properties such as the clustering and degree distribution. In this paper, we mainly explore the power-law networks because of the similarity of features for Internet-like graphs. To emulate the topologies, we use the A. Barabasi (BA) model [49] with limited high centrality nodes, the model presents the impractical ideal scenario for cache, such as high request overlap. There are 200 content-centric computing nodes in generated topology. Once the topology is generated, we then attach 20 service-nodes, and share them within 2,000 users. After the objects are distributed among the source service-nodes uniformly, we randomly select their individual points of attachment. The distribution that all objects requested by each node in every time interval could be seen as Zipf distribution: $\sum_{i=1}^{N}(c/i^{\beta}) = 1,\ c = 1, 2, \ldots, n$. In addition, we normalize the global cache capacity c_{total} to be 0.01 over the total cache system.

B. IMPORTANT FACTORS OF RESOURCE ALLOCATION

When exploring the important factors which influence the QoE, our first target is cache allocation. To study the importance of cache allocation for QoE, in contrast to only considering the cache capacity, we intend to explore whether the cache allocation is important to performance. Our simulations are carried out with different allocation schemes, including resource allocation based on DQN algorithm, R-SPT, R-SPTC and a few other approaches, i.e., Degree Centrality (DC), Betweenness Centrality (BC) [16]. Least Frequently Used (LFU) [50] approach is the cache replacement policy used in all groups of simulations.

The performances of the algorithms are shown in Figure 3. The x-axis denotes the total cache capacity, which is normalized by the number of nodes multiplied by the number of content chunks. The remaining network cost is shown on the y-axis. To improve comparability, we also normalized the remaining network cost under different schemes while varying $\lg(c_{total})$ in Figure 3 as a ratio of Homogeneous Allocation (HM) [49].

The DA contributes little to the cache performance based on Closeness Centrality (CC) [16]. They provide less than 0.04 network cost reduction. In contrast, the other two indicators, i.e., DC, BC, contribute greater to reduce network cost. Note that, DC is effective when the overall budget is more than 0.01, while BC has better performance when the cache capacity is small. Regardless of the budget of overall capacity, we find that BC never allocates cache space to root nodes. In the condition where the caches are allocated to the core nodes and overall budget is small, BC method is useful. However, when there exists spare cache space which could be offered to users, it is not efficient. The R-SPT is better than all other existing algorithms. The QoE under different schemes while varying transmission rate is shown in Figure 4. We can see that when the transmission rate is very low, the QoEs of all allocation methods are low and have little difference. Therefore, the allocation method is not the main factor of QoE when the transmission rate is low. This finding is consistent with practical experience.

We also intend to validate the influence for QoE via k, which is mentioned when we design algorithms in Section V. The importance of k lies in not only that it will influence the exact result due to the proposed method, but also that it is one critical factor of the computation complexity. When the continuous $R^{i}_{a\to t}$, $i \in \mathcal{N}$, $\{a, t\} \in \mathcal{V}$, is divided into k intervals and the value of k division points is the set of $\mathcal{Q}$, the value range is one part of the original interval, so the solution of the model is also the one part of all solutions. When k becomes larger, the number of the value for division point grows. In this condition, the probability for the green solution or suboptimal solution will appear in the green solution that we can obtain when k increases. For this reason, we can say the green solution in the set will get close to the optimal one with the increasing k. On the other hand, we would consider the computation and time. By R-SPT, we find the value of k influences the outermost plane loop in algorithm. This is because when k is bigger, the computation time will become longer.

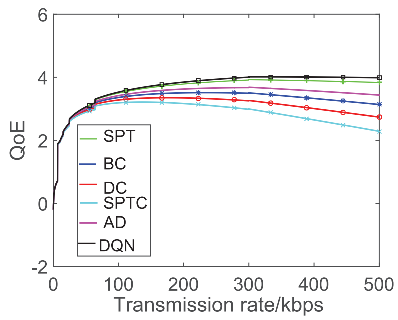


FIGURE 5. QoE under different schemes while varying transmission rate.

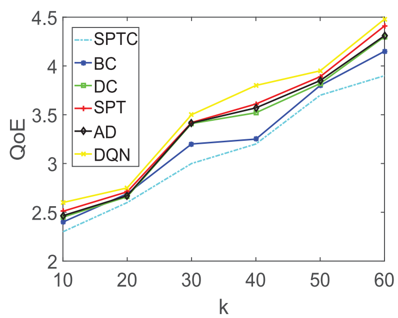


FIGURE 6. QoE performance versus k with different methods.

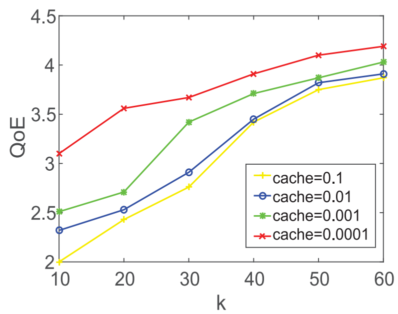


FIGURE 7. QoE performance versus k with different cache capacities.

The results shown in Figures 5 and 6 confirm our idea. The Figure 6 under different methods shows the green solution changes with the variation of k. Clearly, when k increases, QoE increases too. In particular, approximate dynamic (AD) method [16] shows the worst solution, while others show the better results. However, the QoE performance with different cache capacities in Figure 6 grows slowly when k is improved by the same step every time when $k \geq 50$. This proves that k is not the main factor of QoE again.

C. INFLUENCE OF CONTENT POPULARITY

To explore the influences of content popularity distribution, the performance under different coefficient β is presented in Figure 7. We mainly discuss the popularity of the service, which is denoted by p^i. Taking the previous model into consideration, we know that the q^i is the probability of f^i being requested by users. We accordingly believe that the p^i is macro distribution of content f^i. To change the distribution,

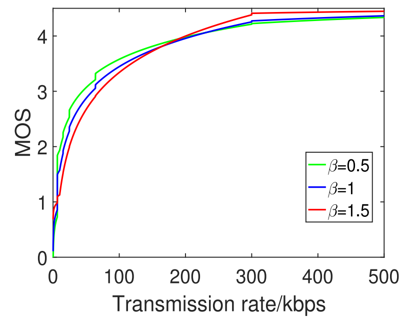


FIGURE 8. MOS under different β.

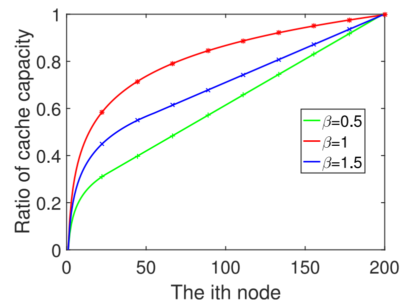


FIGURE 9. Cache capacity cumulative on nodes.

we change the parameter β. For the ith object, the requested probability is proportional to $1/i^{\beta}$. We use to BA topology to carry out simulations. The attachment order in the model is indicated by the node index on the x-axis. The index value is low when the node is core nodes and has a large centrality indicator value.

Figure 8 shows that more cache capacity will be allocated in the core when the popularity distribution is less skewed. In practical, to ensure enough interest overlap for cache hit, we should enhance cache capacity for lots of users. In this case, the higher skewed popularity distribution indicates better performance when it is served by edge caching. In contract, the popularity distribution with less skewed would perform better when it is served by core caching.

Then, when β increases, the optimal performance converges to the homogeneous allocation. More skewed the range of requested content, more effective when the caches are distributed to the edge of network. In this situation, the edge network will make the service for its own users effectively. It is different from the core caching, where the requirement of users is offered intensively. There exists other benefit of reducing the transmission cost and delay. It is suitable to use optimal cache allocation to help Internet Service Provider reduce their transmission costs.

Besides, we also study the influence of skew parameter β on MOS. We find that the MOS does not change greatly as β increases in Figure 8. In this condition, we can say that the skew parameter β is not the main factor of MOS, too. Because the QoE is determined by both network cost and MOS, we intend to analyze the impact on the QoE of skew parameter β. In Figure 9, we find that there are not significant difference for QoE under different β due to the difference of

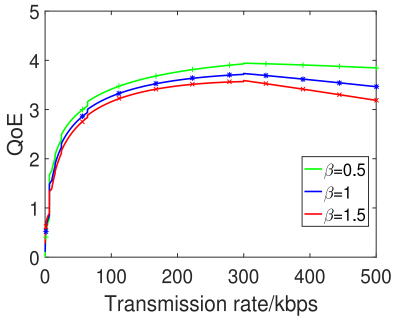


FIGURE 10. QoE performance versus transmission rate, under different β.

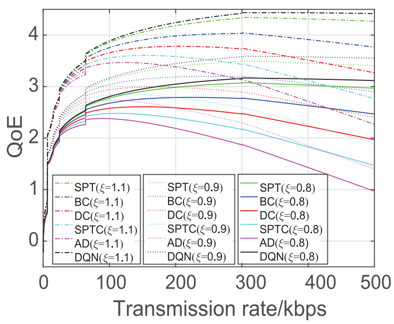


FIGURE 11. QoE performance versus transmission rate, under different ξ.

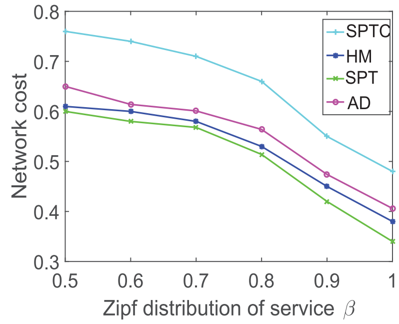


FIGURE 12. Network cost under different schemes while varying the Zipf distribution of service.

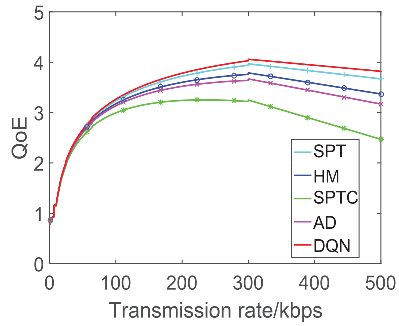


FIGURE 13. QoE under different schemes.

content popularity. The reason is that the cache allocation schemes change along with the variation of β. In addition, we explore the factor of weight between the network cost and experience of users. The results are shown in Figure 10. It is easy to find that QoE is sensitive to ξ.

D. RESOURCE ALLOCATION BASED ON R-SPTC

We also intend to validate that R-SPTC can obtain the quivalent performance without incoming huge computation cost. In Figure 11, we compare the results of R-SPTC against the homogeneous allocation (e.g., AD method [16]) and R-SPT. We change the popularity distribution to inspect the results. The superior performance for dynamic allocation is shown in Figure 12. In details, the averaged QoE of R-SPT, R-SPTC and DQN is 2.6, 2.6, and 2.8, respectively. The maximum QoE of R-SPT, R-SPTW and DQN is 3.6, 3.6, and 4, respectively. We can see that, the resource allocation based DQN algorithm generates the highest QoE value. In any popularity of skew, the result of dynamic allocation approach is close to the optimal result. The performance loss in fact is bounded by only 0.05 on average. In a worst-case scenario, the dynamic allocation will provide the great improvements. Besides, Figure 12 shows that the dynamic allocation schemes also have better QoE performance than AD method [16].

E. RESOURCE ALLOCATION BASED ON DQN

In addition, we try to validate that the green resource allocation algorithm based on DQN can obtain the better performance than other methods. In Figure 13, in contrast to the results of resource allocation based on DQN algorithm and the homogeneous allocation algorithms. We can see that the AD method exhibits the poor QoE performance. In any popularity of skew, the result of resource allocation approach based on DQN is the better QoE than others.

F. SUMMARY OF SIMULATION

Though the previous simulation results and analysis, we have the following summarizes:

- It is highly suboptimal to allocating cache space in the IoT. The benefit for using a better method to allocate resource has large impact on the QoE. When we use the R-SPT to allocate the cache resource, the result is better than the homogeneous manner.
- The transmission rate is a main factor that impacts the QoE. The set of $\mathcal{Q}$ will determine the max QoE that we can obtain. In practice, it will lead to a poor QoE when the transmission rate is too high or too low. So the ISP should select the appropriate transmission rate of service for users.
- The type of content popularity has a marginal influence on MOS. However, it will alter the optimal deployment, which can lead to different QoE. Cache should be allocated to the edge for aggregating requests when the demands are highly skewed. To our surprise, we find that there are not significant performance variations for QoE due to the differences of content popularity.
- The R-SPTC comes with limited QoE and performance penalty compared with the R-SPT. That is because dynamic allocation is independent of the content and source service-node. In the highly dynamic networks,

the dynamic allocation is suitable to the context where content is frequently published. However, resource allocation based on DQN has better QoE performance than other algorithms, which proves that the proposed DQN is a good choice to the deploy cache resource for content-centric IoT.

VIII. CONCLUSIONS

This work has studied the resource allocation in content-centric IoT and improves the QoE. We have analyzed lots of factors which impact the resource allocation and how they subsequently influence the QoE. In order to solve the problem of frequent content publishing, we propose three resource allocation algorithms. The experiments show that it is important to adopt heterogeneous resource allocation. We also find lots of factors which may impact QoE, including the content popularity and weight factor. There is not a approach that fit the all conditions of resource allocation in content-centric IoT. Instead, the ISP should decide the scheme for resource allocation based on the IoT conditions.

ACKNOWLEDGMENTS

This work is supported by NSFC (61572262, 61772286), China Postdoctoral Science Foundation (2017M610252), China Postdoctoral Science Special Foundation (2017T100297), Open Research Fund of Jiangsu Engineering Research Center of Communication and Network Technology, NJUPT, and Strategic Information and Communications R&D Promotion Programme (SCOPE No. 162302008), MIC, Japan. Parts of this paper have been published on IEEE International Conference on Communications (ICC), 2018.

REFERENCES

[1] K. Wang, and Y. Yu, "A query-matching mechanism over out-of-order event stream in IoT," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 13, no. 3/4, pp. 197–208, Jul. 2013.
[2] Z. Su, and Q. Xu, "Content distribution over content centric mobile social networks in 5G," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 66–72, Jun. 2015.
[3] K. Wang, L. Gu, S. Guo, H. Chen, V. C. M. Leung, and Y. Sun, "Crowd-Sourcing-based content-centric network: A social perspective," *IEEE Netw.*, vol. 31, no. 5, pp. 28–34, Sep. 2017.
[4] L. Zhou, M. Chen, Y. Qian, and H. Chen, "Fairness resource allocation in blind wireless multimedia communications," *IEEE Trans. Multimedia*, vol. 15, no. 4, pp. 946–956, Jun. 2013.
[5] L. Qian, Z. Cheng, Z. Fang, L. Ding, F. Yang, and W. Huang, "A QoE-driven encoder adaptation scheme for multi-user video streaming in wireless networks," *IEEE Trans. Broadcast.*, vol. 63, no. 1, pp. 20–31, Mar. 2017.
[6] Z. Deng, Y. Liu, J. Liu, X. Zhou, and S. Ci, "QoE-oriented rate allocation for multipath high-definition video streaming over heterogeneous wireless access networks," *IEEE Syst. J.*, vol. 11, no. 4, pp. 2524–2535, Dec. 2017.
[7] K. Wang, X. Qi, L. Shu, D. Deng, and J. Rodrigues, "Toward trustworthy crowdsourcing in social internet of things," *IEEE Wireless Commun. Mag.*, vol. 30, no. 5, pp. 30–36, Oct. 2016.
[8] K. Eshraghian, K. Cho, O. Kavehei, S. Kang, D. Abbott, and S. Kang, "Memristor MOS content addressable memory (MCAM): Hybrid architecture for future high performance search engines," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 19, no. 8, pp. 1407–1417, Aug. 2011.
[9] X. Zhou, K. Wang, W. Jia, and M. Guo, "Reinforcement learning-based adaptive resource management of differentiated services in geo-distributed data centers," in *Proc. IEEE/ACM Int. Symp. Quality Ser.*, 2017, pp. 1–6.
[10] R. Mok, R. Chang, and W. Li, "Detecting low-quality workers in QoE crowdtesting: A worker behavior-based approach," *IEEE Trans. Multimedia*, vol. 19, no. 3, pp. 530–543, Oct. 2017.
[11] X. Li, X. Ge, X. Wang, and V. Leung, "Energy efficiency optimization: Joint antenna subcarrier power allocation in OFDM-DASs," *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7470–7483, Nov. 2016.
[12] Z. Ye, S. Mistry, A. Bouguettaya, and H. Dong, "Long-term QoS-aware cloud service composition using multivariate time series analysis," *IEEE Trans. Serv. Comput.*, vol. 9, no. 3, pp. 382–393, May 2016.
[13] K. Wang, Y. Wang, Y. Sun, S. Guo, and J. Wu, "Green industrial Internet of Things architecture: An energy-efficient perspective," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 48–54, Dec. 2016.
[14] Y. Wang, *et al.*, "A data-driven architecture for personalized QoE management in 5G wireless networks," *IEEE Wireless Commun. Mag.*, vol. 24, no. 1, pp. 102–110, Feb. 2017.
[15] Y. Lu, M. Motani, and W. Wong, "A QoE-aware resource distribution framework incentivizing context sharing and moderate competition," *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1364–1377, Jun. 2016.
[16] Y. Wang, Z. Li, G. Tyson, S. Uhlig, and G. Xi, "Design and evaluation of the optimal cache allocation for content-centric networking," *IEEE Trans. Comput.*, vol. 65, no. 1, pp. 95–107, Jan. 2016.
[17] K. Gai, M. Qiu, H. Zhao, L. Tao, and Z. Zong, "Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing," *J. Netw. Comput. Appl.*, vol. 59, pp. 46–54, Jan. 2016.
[18] H. Zhu, Y. Cao, W. Wang, B. Liu, and T. Jiang, "QoE-aware resource allocation for adaptive device-to-device video streaming," *IEEE Netw.*, vol. 29, no. 6, pp. 6–12, Dec. 2015.
[19] G. Li, M. Dong, K. Ota, J. Wu, J. Li, and T. Ye, "Towards QoE named content-centric wireless multimedia sensor networks with mobile sinks," in *Proc. Int. Conf. Commun.*, 2017, pp. 1–6.
[20] X. Liu, *et al.*, "RFID Estimation with blocker tags," *IEEE/ACM Trans. Netw.*, vol. 25, no. 1, pp. 224–237, Feb. 2017.
[21] W. Rahman, D. Yun, and K. Chung, "A client side buffer management algorithm to improve QoE," *IEEE Trans. Consum. Electron.*, vol. 62, no. 4, pp. 371–379, Feb. 2016.
[22] H. Tran, S. Hoceini, A. Mellouk, J. Perez, and S. Zeadally, "QoE-based server selection for content distribution networks," *IEEE Trans. Comput.*, vol. 63, no. 11, pp. 2803–2815, Feb. 2014.
[23] L. Qian, Z. Chen, Z. Fang, L. Ding, F. Yang, and W. Huang, "A QoE-driven encoder adaptation scheme for multi-user video streaming in wireless networks," *IEEE Trans. Broadcast.*, vol. 63, no. 1, pp. 20–31, Mar. 2017.
[24] K. Gai, M. Qiu, and H. Zhao, "Energy-aware task assignment for mobile cyber-enabled applications in heterogeneous cloud computing," *J. Parallel Distrib. Comput.*, vol. 11, pp. 126–135, Jan. 2018.
[25] M. Afshang, H. Dhillon, and P. Chong, "Fundamentals of cluster-centric content placement in cache-enabled device-to-device networks," *IEEE Trans. Commun.*, vol. 64, no. 6, pp. 2511–2526, Jun. 2016.
[26] Y. Li, H. Xie, Y. Wen, C. Chow, and Z. Zhang, "How much to coordinate ? optimizing in-network caching in content-centric networks," *IEEE Trans. Netw. Serv. Manag.*, vol. 12, no. 3, pp. 420–434, Sep. 2015.
[27] K. Wang, J. Yu, Y. Yu, Y. Qian, D. Zeng, S. Guo, Y. Xiang, and J. Wu, "A survey on Energy Internet: Architecture, approach and emerging technologies," *IEEE Syst. J.*, vol. PP, no. 99, pp. 1–14, Jan. 2017.
[28] X. Liu, X. Xie, K. Wang, S. Guo, J. Cao, and K. Li, "Pinpointing anomaly RFID tags: Situation and opportunities," *IEEE Netw.*, vol. 31, no. 6, pp. 40–47, Nov. 2017.
[29] V. Pacifici and G. Dan, "Coordinated selfish distributed caching for peering content-centric networks," *IEEE/ACM Trans. Netw.*, vol. 24, no. 6, pp. 3690–3701, Dec. 2016.
[30] B. Zhou, Y. Cui, and M. Tao, "Stochastic content-centric multicast scheduling for cache-enabled heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 6284–6297, Sep. 2017.
[31] Z. Zhou, K. Ota, M. Dong, and C. Xu, "Energy-efficient matching for resource allocation in D2D enabled cellular networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 5256–5268, Jun. 2017.
[32] Z. Zhou, M. Dong, K. Ota, G. Wang, and L. Yang, "Energy-efficient resource allocation for D2D communications underlaying cloud-RAN-based LTE-A networks," *IEEE Internet Things J.*, vol. 3, no. 3, pp. 428–438, Jun. 2016.
[33] M. Gao, K. Wang, and L. He, "Probabilistic model checking and scheduling implementation of energy router system in energy internet for green cities," *IEEE Trans. Ind. Informat.*, vol. PP, no. 99, pp. 1–10, Jan. 2018.
[34] P. Truong and B. Mathieu, "Economic incentives for deploying local-aware ICN-based content delivery," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun.*, 2016, pp. 352–357.
[35] X. Liu, *et al.*, "Top-k queries for categorized RFID systems," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 2587–2600, Oct. 2017.

[36] S. Nikolaou, R. Renesse, and N. Schiper, “Proactive cache placement on cooperative client caches for online social networks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 4, pp. 1174–1186, Apr. 2016.

[37] C. Xu, K. Wang, and M. Guo, “Intelligent resource management in blockchain based cloud data centers,” *IEEE Cloud Comput.*, vol. 4, no. 6, pp. 50–59, Dec. 2017.

[38] D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee, and K. Ramakrishnan, “Optimal content placement for a large-scale VoD system,” *IEEE Trans. Netw.*, vol. 24, no. 4, pp. 2114–2127, Aug. 2016.

[39] J. Chakareski, “Joint source-channel rate allocation and client clustering for scalable multistream IPTV,” *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2429–2439, Aug. 2015.

[40] N. Kumar, S. Zeadally, and J. Rodrigues, “QoS-aware hierarchical web caching scheme for online video streaming applications in internet-based vehicular ad hoc networks,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 12, pp. 7892–7900, Dec. 2015.

[41] X. Liu, S. Zhang, B. Xiao, and K. Bu, “Flexible and time-efficient tag scanning with handheld readers,” *IEEE Trans. Mobile Comput.*, vol. 15, no. 4, pp. 840–854, Apr. 2016.

[42] H. Shao, H. Zhao, Y. Sun, J. Zhang, and Y. Xu, “QoE-aware downlink user-cell association in small cell networks: A transfer-matching game theoretic solution with peer effects,” *IEEE Access*, vol. 4, pp. 10029–10041, Nov. 2016.

[43] M. Venkataraman and M. Chatterjee, “Effects of internet path selection on video-QoE: Analysis and improvements,” *IEEE Trans. Netw.*, vol. 22, no. 3, pp. 689–702, Jun. 2014.

[44] J. Song, F. Yang, Y. Zhou, S. Wan, and H. Wu, “QoE evaluation of multimedia services based on audiovisual quality and user interest,” *IEEE Trans. Multimedia*, vol. 18, no. 3, pp. 444–457, Mar. 2016.

[45] X. Liu, S. Zhang, and K. Bu, “A locality-based range-free localization algorithm for anisotropic wireless sensor networks,” *Telecommun. Syst.*, vol. 62, no. 1, pp. 3–13, 2016.

[46] P. Sermpezis, and T. Spyropoulos, “Effects of content popularity on the performance of content-centric opportunistic networking: An analytical approach and applications,” *IEEE/ACM Trans. Netw.*, vol. 24, no. 6, pp. 3354–3368, Dec. 2016.

[47] K. Mitra, A. Zaslavsky, and C. Ahlund, “Context-aware QoE modelling, measurement, and prediction in mobile computing systems,” *IEEE Trans. Mobile Comput.*, vol. 14, no. 5, pp. 920–936, May 2015.

[48] R. Vaze, “Online knapsack problem and budgeted truthful bipartite matching,” in *Proc. IEEE Conf. Comput. Commun.*, 2017, pp. 1–9.

[49] O. Granmo, B. Oommen, S. Myrer, and M. Olsen, “Learning automata-based solutions to the nonlinear fractional knapsack problem with applications to optimal resource allocation,” *IEEE Trans. Syst., Man, and Cybern.*, vol. 37, no. 1, pp. 166–175, Jan. 2007.

[50] O. Cihan and M. Akar, “Fastest mixing reversible markov chains on graphs with degree proportional stationary distributions,” *IEEE Trans. Autom. Control*, vol. 60, no. 1, pp. 227–232, Jan. 2015.

[51] L. Chen, H. Shen, and K. Sapra, “Distributed autonomous virtual resource management in datacenters using finite-markov decision process,” in *Proc. ACM Symp. Cloud Comput.*, Nov. 2014, pp. 1–13.

[52] Z. Han, H. Tan, G. Chen, R. Wang, Y. Chen, and F. Lau, “Dynamic virtual machine management via approximate markov decision process,” in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun.*, 2016, pp. 1–9.

[53] L. Xiao, Y. Li, G. Han, H. Dai, and H. Vincent, “A secure mobile crowdsensing game with deep reinforcement learning,” *IEEE Trans. Inform. Forensics and Security*, vol. 13, no. 1, pp. 35–47, Jan. 2018.

[54] N. Liu, *et al.*, “A hierarchical framework of cloud resource allocation and power management using deep reinforcement learning,” in *Proc. IEEE Int. Conf. Distrib. Comput. Syst.*, 2017, pp. 372–382.

[55] H. Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double Q-learning,” in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 2094–2100.

[56] L. Ji, “Reinforcement learning for robots using neural networks,” PhD thesis, Carnegie Mellon University, Pittsburgh., Jan. 1993.

[57] H. Modares, I. Ranatunga, F. Lewis, and D. Popa, “Optimized assistive humanCrobot interaction using reinforcement learning,” *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 655–667, Mar. 2016.

[58] Q. Meng, K. Wang, B. Liu, T. Miyazaki, and X. He, “QoE-based big data analysis with deep learning in pervasive edge environment,” in *Proc. Int. Conf. Commun.*, 2018.

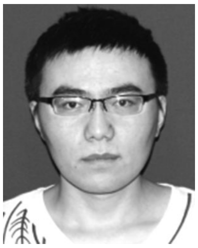
[59] Q. Wu, Z. Li, and G. Xie, “Codingcache: Multipath-aware CCN cache with network coding,” *in Proc. 5th Annu. ACM Special Interest Group Data Commun.*, 2013, pp. 41–42.

XIAOMING HE is working toward the postgraduate degree in the School of Internet of Things, Nanjing University of Posts and Telecommunications. His current research interests include Vehicle-to-Grid, big data, machine learning, content-centric network.

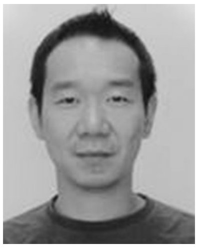
KUN WANG (M’13-SM’17) received the BEng and PhD degrees in computer science from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2004 and 2009, respectively. From 2013 to 2015, he was a postdoc fellow in Electrical Engineering Department, University of California, Los Angeles (UCLA), California. In 2016, he was a research fellow in the School of Computer Science and Engineering, the University of Aizu, Aizu-Wakamatsu City, Fukushima, Japan. He is currently a research fellow in the Department of Computing, the Hong Kong Polytechnic University, Hong Kong, China, and also a full professor in the School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing, China. He has published more than 100 papers in referred international conferences and journals. He has received Best Paper Award at IEEE GLOBECOM16. He serves as associate editor of the *IEEE Access*, editor of the *Journal of Network and Computer Applications*, the *Journal of Communications and Information Networks*, the *EAI Transactions on Industrial Networks and Intelligent Systems* and guest editors of the *IEEE Access*, the *Future Generation Computer Systems*, the *Peer-to-Peer Networking and Applications*, and the *Journal of Internet Technology*. He was the symposium chair/co-chair of IEEE IECON16, IEEE EEEIC16, IEEE WCSP16, IEEE CNCC17, etc. His current research interests include mainly in the area of big data, wireless communications and networking, smart grid, energy Internet, and information security technologies. He is a senior member of the IEEE and member of the ACM.

HUAWEI HUANG (M’16) received the PhD degree in computer science and engineering from the University of Aizu, Japan. His research interests mainly include software-defined networking (SDN), NFV, cloud computing, and IoT. He has been a visiting scholar at the Hong Kong Polytechnic University since 2017. He was a post-doctoral research fellow of JSPS between Oct. 2016 and Mar. 2018. He is also a member of the IEEE.

TOSHIAKI MIYAZAKI (SM’12) received the BE and ME degrees in applied electronic engineering from the University of Electro-Communications, Tokyo, Japan, in 1981 and 1983, respectively, and the PhD degree in electronic engineering from Tokyo Institute of Technology, Tokyo, in 1994. He is a professor with the University of Aizu, Fukushima, Japan, and the Dean of the Undergraduate School of Computer Science and Engineering. His research interests include reconfigurable hardware systems, adaptive networking technologies, and autonomous systems. Before joining the University of Aizu, he has worked for NTT for 22 years, and engaged in research on VLSI CAD systems, telecommunications-oriented FPGAs and their applications, active networks, peer-to-peer communications, and ubiquitous network environments. He was a visiting professor of the Graduate School, Niigata University, in 2004, and a Part-Time lecturer with the Tokyo University of Agriculture and Technology in 2003C2007. He is a senior member of the the IEEE, IEICE and IPSJ.



YIXUAN WANG is working toward the postgraduate degree in the School of Internet of Things, Nanjing University of Posts and Telecommunications. His current research interests include machine learning, content-centric network.



SONG GUO (M'02-SM'11) received the PhD degree in computer science from the University of Ottawa and was a professor with the University of Aizu. He is a full professor in the Department of Computing, The Hong Kong Polytechnic University. His research interests include mainly in the areas of big data, cloud computing and networking, and distributed systems with more than 400 papers published in major conferences and journals. His work was recognized by the 2016 Annual Best of Computing: Notable Books and Articles in Computing in ACM Computing Reviews. He is the recipient of the 2017 IEEE Systems Journal Annual Best Paper Award and other five Best Paper Awards from IEEE/ACM conferences. He was an associate editor of the *IEEE Transactions on Parallel and Distributed Systems* and an IEEE ComSoc Distinguished lecturer. He is now on the editorial board of the *IEEE Transactions on Emerging Topics in Computing*, the *IEEE Transactions on Sustainable Computing*, the *IEEE Transactions on Green Communications and Networking*, and the *IEEE Communications*. He also served as General, TPC and Symposium Chair for numerous IEEE conferences. He currently serves as an officer for several IEEE ComSoc Technical Committees and a director in the ComSoc Board of Governors. He is a senior member of the IEEE.