# Mobility-Aware Content Caching and User Association for Ultra-Dense Mobile Edge Computing Networks

Hui Li[1], Chuan Sun[1], Xiuhua Li[1,*], Qingyu Xiong[1], Junhao Wen[1], Xiaofei Wang[2], and Victor C. M. Leung[3, 4]
[1]School of Big Data & Software Engineering, Chongqing University, Chongqing, China
[2]TKLAN, College of Intelligence & Computing, Tianjin University, Tianjin, China
[3] College of Computer Science & Software Engineering, Shenzhen University, Shenzhen, China
[4]Department of Electrical & Computer Engineering, The University of British Columbia, Vancouver, Canada
*Corresponding author
Email:{h.li, c.sun, lixiuhua, xiong03, jhwen}@cqu.edu.cn, xiaofeiwang@tju.edu.cn, vleung@ieee.org

GLOBECOM 2020 - 2020 IEEE Global Communications Conference | 978-1-7281-8298-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/GLOBECOM42002.2020.9348257

***Abstract*—With the tremendous growth of mobile data traffic generated by various devices such as smartphones, smartpads and wearable devices, it is necessary for mobile network operators to introduce revolutionary networking techniques, thereby satisfying service requirements of mobile users. Recently, mobile edge computing (MEC) has been regarded as an effective technique to alleviate the traffic burden on backhaul networks. In this paper, we investigate the issue of mobility-aware content caching and user association for ultra-dense MEC networks by minimizing the system costs. The problem is formulated as a complex pure integer nonlinear programming, which is NP-hard. To address the original long-term optimization problem, we decompose it into a series of one-slot subproblems, and then optimize the short-term subproblem in two phases (i.e., content caching and user association). We further propose a mobility-aware online caching algorithm to achieve content caching, and a lazy re-association algorithm to determine user association based on matching theory. Trace-driven evaluation results demonstrate that the proposed framework has superior performance on reducing system costs.**

## I. Introduction

With the rapid prevalence of smart mobile devices (e.g., smartphones, smartpads and wearable devices) and the tremendous growth of mobile network services, mobile network operators (MNOs) have been facing an explosive growth of multimedia service data traffic [1]. The traditional cloud based architecture caches contents in the cloud server (CS) far away from users, and it becomes overwhelmed when satisfying massive user requests without degrading the quality of service/experience (QoS/QoE). To address this challenge, mobile edge computing (MEC) is an effective computing paradigm by sinking computation and storage resources to the edges (e.g., base stations (BSs)) of the network [2]. As a result, MEC can effectively satisfy user requests and reduce duplicated content delivery, which leads to lower backhaul burden and reduces transmission costs [3].

In terms of content caching in MEC networks, although contents can be cached at BSs that are equipped with edge servers in proximity to mobile users for lower transmission costs, the BSs are more likely to be overloaded considering that not all contents can be cached due to the limited storage resources of edge servers. Therefore, the BSs should cache the contents properly. Moreover, considering the case of ultra-dense deployment of BSs, e.g., small BSs (SBSs), it is more crucial for mobile users to associate with appropriate SBSs which not only provide sufficient radio resources but also satisfy user requests locally [4]. To achieve the full benefits of ultra-dense MEC networks, the issue of joint content caching and user association needs to be investigated.

Recently, extensive studies have been conducted on content caching or user association in MEC networks [5]–[8]. Jiang *et al.* in [5] proposed a learning-based cooperative content caching scheme in MEC networks, but only focused on the content caching and neglected the impact of user association, which may lead to the performance degradation. In [6], caching the most popular contents was considered to improve network performances, but the caching scheme is hard to capture users' real-time requests. In [7], joint user association and content placement optimization was investigated to minimize the average download delay under the constraints of the users' and SBSs' characteristics. Dai *et al.* [8] proposed an iterative approach to maximize the network utility and backhaul saving. All the studies as mentioned above, considered that users were geographically stationary in the wireless network, and user mobility was neglected. In fact, the locations of mobile users may change constantly. It is reasonable to cache contents at the BSs near the users considering their real-time location information rather than in specific servers.

As an intrinsic feature of MEC networks, there exist several studies investigating the impact of user mobility on caching scheme design. For instance, the study in [9] proposed a probabilistic caching scheme by modeling the user mobility as discrete jumps determined by the average sojourn time. In [10], a mobility-aware content placement scheme was proposed to minimize the average data load, and the user mobility was modeled as a Possion process. Chen *et al.*[11] investigated the problem of caching placement to maximize the cache hit ratio and modeled the user mobility as peer-to-peer connectivity. However, as user movements are random, the association between the BSs (or SBSs) and users may


978-1-7281-8298-8/20/$31.00 ©2020 IEEE

change frequently, resulting in re-association. For example, we consider a practical scenario as shown in Fig. 1. When the user $u_1$ moves into the overlapping area of $SBS_1$ and $SBS_2$, and can associate with $SBS_1$ or $SBS_2$ for the desired content. Considering that user $u_1$ is associated with $SBS_1$ at last moment, if the MNO pushes the requested content of user $u_1$ at the $SBS_1$ rather than $SBS_2$, the user re-association can be avoided. Therefore, it needs to properly design joint content caching and user association strategies especially when taking user mobility into consideration.

Thus, in this paper, we are motivated to investigate the issue of the joint mobility-aware content caching and user association in an ultra-dense MEC network, aiming at minimize the system costs. The main contributions of this paper can be summarized as follow:

- We integrate the issues of the analysis of user association and mobility with edge caching in an ultra-dense MEC network, for practically minimizing system costs as our goal, towards future mobile networks.
- We decompose the formulated sophisticated long-term optimization problem into a series of one-slot subproblems, and propose the low-complexity algorithms for solving the simpler short-term subproblems.
- Trace-driven evaluation results demonstrate that our proposed framework has superior performance on reducing system costs.

The remainder of this paper is organized as follows. Section II introduces the system model and formulates the optimization problem. We present our proposed framework in Section III and trace-driven evaluation results in Section IV. Finally, Section V concludes this paper.

## II. System Model and Problem Formulation

### A. Network Architecture

As illustrated in Fig. 1, we consider the case of an ultra-dense MEC system with a macro BS (MBS), $S$ densely deployed SBSs (denoted by $\mathcal{S} = \{1, 2, ..., S\}$), and $U$ mobile users (denoted by $\mathcal{U} = \{1, 2, ..., U\}$), in which each SBS is equipped with an edge server. The service areas of the SBSs may overlap and the users may be covered by multiple SBSs at the same moment. The SBSs (or edge servers) have limited caching capacities and radio resources to serve the users. The MBS does not have caching capability, but provides content delivery services for the users and SBSs. Denote the set of the MBS and all SBSs as $\mathcal{S}' = \{0\} \bigcup \mathcal{S}$. There is a library of $F$ contents in the network, denoted by $\mathcal{F} = \{1, 2, ..., F\}$, and all the contents are indivisible and available in the CS. Aiming to capture the user mobility effectively, we assume that the system operates in a slotted manner and the time period is equally divided into a series of slots, denoted by $\mathcal{T} = \{1, 2, ..., T\}$. Each time slot $t \in \mathcal{T}$ has a reasonable time span (e.g., several minutes), which is determined by users' request patterns. The SBSs and CS are cooperative to build a stable caching framework, and the MNO core can push contents at the SBSs from CS to reduce the transmission

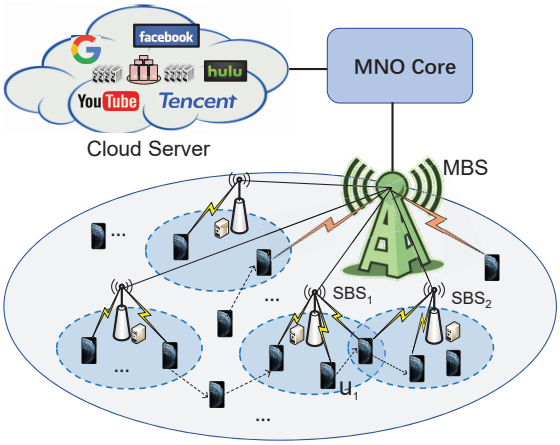


Fig. 1. Illustration of system model of ultra-dense MEC networks.

costs. The users can get the desired contents mainly via two modes, i.e., SBS association mode and MBS association mode. Specifically, the users in SBS association mode get the desired contents via the local cache of SBSs. The SBSs can collect the users' real-time request information, and determine how to effectively update the contents from the CS by considering the current caching status of contents in the considered time slot [12]. In MBS association mode, the mobile users have to acquire the contents from remote CS through the MBS. In addition, we assume that the local popularity of the contents changes slowly, and can be obtained in advance by the system learning and analysis.

### B. Content Caching and User Association Model

Unlike the traditional long-term caching in the content-centric networks, the considered MEC network allows the SBSs to cache and update the contents in a shorter period of time. Denote a binary indicator $y_{s,f}^t \in \{0, 1\}$ for whether content $f$ is cached at $SBS_s$ at time slot $t$ or not. Denote the sets $\mathcal{Y} = \{y_{s,f}^t | \forall s \in \mathcal{S}, f \in \mathcal{F}, t \in \mathcal{T}\}$ and $\mathcal{Y}^{(t)} = \{y_{s,f}^t | \forall s \in \mathcal{S}, f \in \mathcal{F}\}$ as the corresponding content caching policies. Then the storage constraint at each SBS can be expressed as

$$\sum_{f \in \mathcal{F}} y_{s,f}^t d_f \leq D_s, \forall s \in \mathcal{S}, t \in \mathcal{T}, \tag{1}$$

where $d_f$ and $D_s$ denotes the size of content $f$ and the storage capacity of the $SBS_s$, respectively.

Moreover, to get the desired content, each user associates with a suitable BS (i.e., the MBS or SBSs). At each time slot, the MNO core makes the user association policy for all users. Denote $x_{s,u}^t \in \{0, 1\}$ as the dynamic user association policy, where $x_{s,u}^t = 1$ indicates that user $u$ is associated with $SBS_s$ at time slot $t$, and $x_{s,u}^t = 0$ means opposite. Denote $x_{0,u}^t \in \{0, 1\}$ for whether user $u$ is associated with the MBS directly or not. Denote $\mathcal{X} = \{x_{s,u}^t | \forall u \in \mathcal{U}, s \in \mathcal{S}', t \in \mathcal{T}\}$ and $\mathcal{X}^{(t)} = \{x_{s,u}^t | \forall u \in \mathcal{U}, s \in \mathcal{S}'\}$ as the corresponding user association policies. As each user can get the desired content by means of at most one association mode, the user association

policy is constrained by

$$x_{0,u}^t + \sum_{s\in\mathcal{S}} x_{s,u}^t = 1, \forall u \in \mathcal{U}, t \in \mathcal{T}. \tag{2}$$

At each time slot, we assume that each SBS can serve a certain number of users based on their limited radio resources, which is expressed as

$$\sum_{u\in\varphi_s^t} x_{s,u}^t \le N_s, \forall s \in \mathcal{S}, t \in \mathcal{T}, \tag{3}$$

where $N_s$ is the maximum number of users that $\text{SBS}_s$ can serve, and $\varphi_s^t \subseteq \mathcal{U}$ denotes the set of users that are within the service coverage of $\text{SBS}_s$ at time slot $t$. We assume that MBS can serve a large number of users. Due to the page limit, the detailed resource allocation of SBSs are out of the scope of this paper and will be studied in the future work.

### C. User Mobility and Handover Cost Model

In the ultra-dense MEC network, user mobility is hard to estimate, and we model the user mobility among BSs (i.e., the SBSs and the MBS) via Markov chain [13]. We assume that the user moving path is a sequence of BSs visited and the users move from the service coverage of one BS to another BS (maybe users are static) within one adjacent time slot. Denote $M_{s',s}$ as the transition probability from $\text{BS}_{s'}$ to $\text{BS}_s$. Denote $\mathbb{L}_{U\times T}$ as the matrix of users' moving path, where each element $l_u^t \in \mathcal{S}'$ in $\mathbb{L}$ denotes the location information of user $u$ at time slot $t$. Thus, the corresponding probability of user $u$ accessing the area $\text{BS}_s$ can be calculated as $P_{u,s} = P_{u,l_u^1}^{init} \prod_{i=1}^{t-1} M_{l_u^i, l_u^{i+1}}$, here $s$ is equal to $l_u^t$. In a dynamic scenario of user movements, the user re-association cannot be neglected. When the re-association occurs, users may suffer from service interruptions and a loss of QoS. The MNO core should update the service profiles of users to follow the mobility and it will cause handover costs. Hence, the number of re-association should be minimized. The total handover costs caused by re-association for all users at time slot $t$ can be calculated as

$$\Psi_S^t(\mathcal{X}) = \sum_{s\in\mathcal{S}} \sum_{u\in\mathcal{U}} \left| x_{s,u}^t - x_{s,u}^{t-1} \right|, \forall t \in \mathcal{T}. \tag{4}$$

### D. Content Request and Transmission Cost Model

For simplicity, we assume that each user sends a content request at each time slot during the user movement, and the BSs can process the request immediately and then deliver the desired content to the user at the same time slot. Denote $\mathbb{R}_{U\times T}$ as the matrix of users' content requests and each element $r_u^t \in \mathcal{F}$ in $\mathbb{R}$ is the desired content of user $u$ at time slot $t$. Considering the uncertainty of user mobility and spatial variations of content request, the content request probability relies on not only the local popularity, but also the probability of user accessing the area. We assume that the overall content popularity follows a Zipf distribution $q_f = f^{-\beta}(\sum_{i=1}^{|\mathcal{F}|} i^{-\beta})^{-1}$, where $\beta$ is the parameter of Zipf distribution. The request probability for content $f$ of user $u$ at $\text{BS}_s$ can be calculated as $Q_{s,u}^f = q_f P_{u,s}$, satisfying $\sum_{f\in\mathcal{F}} Q_{s,u}^f = 1$, which simply captures the content preferences of the users in different areas.

In the considered MEC network, if the MNO core determines to update the contents at a SBS, the transmission costs between the CS and the SBS (i.e., CS-to-MBS-to-SBS) should not be neglected [14]. When the user gets the desired content via the local cache of SBS, the SBS can deliver the content to the user immediately and causes only SBS-to-End transmission cost. Similarly, if the user acquires the desired content from remote CS via MBS, it will cause the CS-to-MBS-to-End transmission cost. Denote the per-MB transmission cost of CS-to-MBS-to-SBS, SBS-to-End and CS-to-MBS-to-End as $\delta_C$, $\delta_D$ and $\delta_E$, respectively. Thus, based on the request probability, the total transmission costs of all users at time slot $t$ is expressed as

$$\begin{aligned}\Psi_T^t(\mathcal{X},\mathcal{Y}) = &\sum_{s\in\mathcal{S}} \sum_{f\in\mathcal{F}} \widetilde{Q_s^f} [y_{s,f}^t - y_{s,f}^{t-1}]^+ d_f \delta_C + \\ &\sum_{s\in\mathcal{S}} \sum_{u\in\mathcal{U}} Q_{s,u}^{r_u^t} x_{s,u}^t y_{s,r_u^t}^t d_{r_u^t} \delta_D + \sum_{u\in\mathcal{U}} Q_{0,u}^{r_u^t} x_{0,u}^t d_{r_u^t} \delta_E,\end{aligned} \tag{5}$$

where $[x]^+ = \max\{x, 0\}$, $\widetilde{Q_s^f} = \sum_{u\in\varphi_s^t} Q_{s,u}^f / |\varphi_s^t|$ is the average request probability. Besides, the above three additional terms denote the total transmission costs of CS-to-MBS-to-SBS, SBS-to-End, and CS-to-MBS-to-End, respectively.

### E. Problem Formulation

In the ultra-dense MEC network, we mainly consider the handover costs and transmission costs as the system costs. To achieve a cost-effective mobility-aware content caching and user association scheme, i.e., $\{\mathcal{X}, \mathcal{Y}\}$, our goal is to minimize the time-average system costs based on the constraints of mobile users and SBSs. The corresponding optimization problem can be formulated as

$$\min_{\{\mathcal{X},\mathcal{Y}\}} \frac{1}{T} \sum_{t\in\mathcal{T}} (\mathbb{Z}_1\{\Psi_T^t(\mathcal{X},\mathcal{Y})\} + \mathbb{Z}_2\{\Psi_S^t(\mathcal{X})\}) \tag{6a}$$

$$\textit{s.t.} \quad \text{the same with } (1), (2), \text{ and } (3), \tag{6b}$$

$$x_{s,u}^t, y_{s,f}^t \in \{0,1\}, \forall s \in \mathcal{S}', u \in \mathcal{U}, f \in \mathcal{F}, t \in \mathcal{T}, \tag{6c}$$

where the used $\mathbb{Z}_1\{.\}$ and $\mathbb{Z}_2\{.\}$ are linear functions with positive slopes for normalizing and eliminating the effects of different dimensions, e.g., $\mathbb{Z}_1\{x\} = \mathbb{Z}_2\{x\} = \frac{x}{\max\{x\}}$. The problem is a complex pure integer nonlinear programming (PINLP) problem and NP-hard.

## III. Proposed Framework Design

In a long-term issue, it is hard to determine the content caching and user association policy for all time slots at once. To address this challenge, we decompose the original problem in (6) into $T$ one-slot subproblems. By analyzing these short-term subproblems, we find that the partial object function $\Psi_T^t$ is a coupling term, it's hard to optimize by using exact methods. To address these short-term subproblems, we optimize them in two phases, i.e., content caching and user association, and propose the corresponding methods. The overall procedure can be shown in Fig. 2.

### A. Content Caching

As the user location changes over time, a reasonable content caching policy can reduce not only transmission costs but also

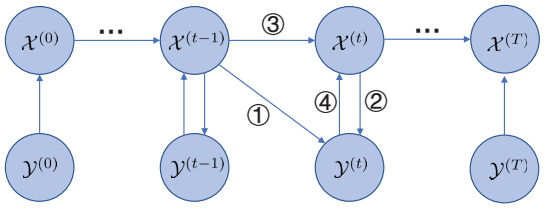


Fig. 2. Content caching and user association policy-making process.

handover costs (an example is shown in Section I). Moreover, the system costs occur at time slot $t$ which are partially related to the policy at time slot $t-1$. In this section, we focus on optimizing the $\mathcal{Y}^{(t)}$ when the $\mathcal{X}^{(t)}$ and $\mathcal{X}^{(t-1)}$ are given (i.e., ① and ② in Fig. 2). We propose a mobility-aware online content caching algorithm to find a suboptimal content caching policy $\mathcal{Y}^{(t)}$ at first, as shown in Algorithm 1. The key idea to cache the fittest content between the contents and the corresponding SBSs based on the users' real-time requests.

In detail, Algorithm 1 involves two procedures as: 1) remove outdated contents at the SBSs. At each time slot, compute the average request number for each content in $[t-\tau, t]$, and check whether it is less than the overall average request number $\overline{n}$ or the content has not been requested for $\eta$ time slot. If the condition is satisfied, the SBS will remove the content from the storage (as shown in Lines 3-7); $\tau$ is the time span of the observation, $n_f^{\upsilon}$ and $\tilde{t}_f$ denotes the total request number at time slot $\upsilon$ and the last request time for content $f$, respectively; 2) calculate the fitness between the contents and SBSs. In each iteration, as the content is indivisible, we calculate the fitness as the difference between the transmission costs of CS and SBS based on user request probability and content weight. We select the fittest content $f^* = \arg\max_{f\in\mathcal{F}}\{Fit_s^f\}$ (in Line 20), and then SBS$_s$ caches content $f^*$ until exceeding its storage size (as shown in Lines 9-22). Moreover, the complexity of Algorithm 1 is $O(|S||U||F|)$.

**Algorithm 1** Mobility-Aware Online Content Caching Algorithm.

**Input:** All users and BSs information, $\mathbb{R}$ and $\mathcal{X}^{(t-1)}$.
**Output:** $\mathcal{Y}^{(t)}$.

1: **Initialize** $U_f \leftarrow$ NULL, $\{Fit_s^f\} = \mathbf{0}_{S\times T}$, set $\mathcal{X}^{(t)}$ by associating the users in $\mathcal{U}$ with the SBS which has the largest number of remaining association.
2: **for** each $s \in \mathcal{S}$ **do**
3: **for** each content $f$ has been placed at SBS$_s$ **do**
4: **if** $t > \tau$ and $\frac{1}{\tau}\sum_{\upsilon=t-\tau}^{t} n_f^{\upsilon} < \overline{n}$ or $t - \tilde{t}_f \geqslant \eta$ **then**
5: Remove the content from SBS$_s$, $y_{s,f}^t = 0$.
6: **end if**
7: **end for**
8: update the value $\tilde{t}_f$ for all contents by using $\mathbb{R}$.
9: **for** each $f \in \mathcal{F}$ **do**
10: $U_f \leftarrow$ Find all users request content $f$ at time slot $t$.
11: **for** each $u \in U_f$ **do**
12: **if** $x_{s,u}^{t-1} == 1$ **then**
13: $Fit_s^f \leftarrow Fit_s^f + \omega Q_{s,u}^f(\delta_E - \delta_D)d_f$.
14: **else**
15: $Fit_s^f \leftarrow Fit_s^f + (1-\omega)Q_{s,u}^f(\delta_E - \delta_D)d_f$.
16: **end if**
17: **end for**
18: **end for**
19: **while** $\sum_{f\in\mathcal{F}} y_{s,f}^t d_f \leq D_s$ and $y_{s,f}^t == 0$ **do**
20: Find $f^* = \arg\max_{f\in\mathcal{F}}\{Fit_s^f\}$.
21: $y_{s,f^*}^t = 1, f \leftarrow f+1$.
22: **end while**
23: **end for**
24: **return** $\mathcal{Y}^{(t)}$.

## B. User Association

In the previous section, we determine a feasible content caching policy $\mathcal{Y}^{(t)}$. In this section, we focus on amending the user association policy $\mathcal{X}^{(t)}$. The corresponding problem can be rewritten as

$$\min_{\{\mathcal{X}^{(t)}\}} \mathbb{Z}_1\{\Psi_T^t(\mathcal{X}^{(t)}, \mathcal{Y}^{(t)})\} + \mathbb{Z}_2\{\Psi_S^t(\mathcal{X}^{(t)})\} \tag{7a}$$

$$s.t. \quad x_{0,u}^t + \sum_{s\in\mathcal{S}} x_{s,u}^t = 1, \forall u \in \mathcal{U}, t \in \mathcal{T}, \tag{7b}$$

$$\sum_{u\in\varphi_s^t} x_{s,u}^t \leq N_s, \forall s \in \mathcal{S}, t \in \mathcal{T}, \tag{7c}$$

$$x_{s,u}^t \in \{0,1\}, \forall s \in \mathcal{S}, u \in \mathcal{U}, t \in \mathcal{T}. \tag{7d}$$

The objective function in (7a) is related to a consecutive adjacent time. To minimize it, we optimize the $\mathcal{X}^{(t)}$ based on the $\mathcal{X}^{(t-1)}$ and the obtained value $\mathcal{Y}^{(t)}$ (i.e., ③ and ④ in Fig. 2). We propose a low-complexity lazy re-association algorithm base on matching theory [15], as shown in Algorithm 2. Particularly, we state the users and the BSs as two disjoint sets in a bipartite graph. Then we define the matching as a mapping from $\mathcal{S}'$ to $\mathcal{U}$. Formally, the matching between the users and the BSs can be defined as follow.

*Definition 1*: Given two disjoint sets, the set $\mathcal{S}'$ and the set $\mathcal{U}$, an one-to-many matching $\mu$ is a function from the set $\mathcal{S}' \cup \mathcal{U}$ into the set $\mathcal{S}' \cup \mathcal{U}$ such that for each $u \in \mathcal{U}$ and $s \in \mathcal{S}'$:

1) $\mu(u) \in \mathcal{S}'$ and $|\mu(u)| = 1$;
2) $\mu(s) \subseteq \mathcal{U}$ and $|\mu(s)| \leq N_s$;
3) $s = \mu(u) \Leftrightarrow u \in \mu(s)$.

Here, $|\mu(.)|$ denotes the size of the matching outcome. The first two conditions satisfy the constraints in (7b) and (7c), respectively. Moreover, we introduce swap operations which indicate that two different users exchange the associated BS with each other and the other user associates with the same BS. Besides, the swap operations are permitted if and only if the transmission costs are reduced after swapping.

Finally, Algorithm 2 involves two steps: 1) Lazy re-association. Based on the historical user association policy, when the user request at time slot $t$ can be satisfied by the SBS which is associated with at time slot $t-1$, we keep the same association (as shown in Lines 3-7); 2) Swap and update matching. Through the swap operation, the users' desired contents may be obtained from SBSs locally. Meanwhile, due

to the constraint of (7c), the users (who get the contents directly from the MBS before) may have the chance to get the contents from SBSs, which can reduce the transmission costs (as shown in Lines 9-20). In this way, the total transmission costs and handover costs can be reduced significantly. Besides, the complexity of Algorithm 2 is $O(|U|^2)$.

**Algorithm 2** Lazy Re-association Algorithm based on Matching Theory.

**Input:** All users and BSs information, $\mathcal{X}^{(t-1)}$ and $\mathcal{Y}^{(t)}$.
**Output:** $\mathcal{X}^{(t)}, \Psi_S^t$.

1: Initialize $\Psi_S^t \leftarrow 0$, $\Psi_T^t \leftarrow 0$;
2: —Step 1. [Lazy re-association]
3: **for** each $u \in \mathcal{U}$ **do**
4: **if** user $u$'s request can be satisfied by the SBS which is associated with at time slot $t-1$ **then**
5: Keep the same association, $\mathcal{U} \leftarrow \mathcal{U} \setminus u$.
6: **end if**
7: **end for**
8: —Step 2. [Swap and update matching]
9: **repeat**
10: **for** each $u \in \mathcal{U}$ **do**
11: **for** each $k \in \mathcal{U}$ and $k \neq u$ **do**
12: **if** $\mu(u) \neq \mu(k)$ **then**
13: Calculate the system costs $\Psi_T^t$ after swap.
14: **if** the system costs $\Psi_T^t$ decreases **then**
15: Update the $\mathcal{X}^{(t)}$ for user $u$ and user $k$.
16: **end if**
17: **end if**
18: **end for**
19: **end for**
20: **until** the system cost value $\Psi_T^t$ converges.
21: Calculate the $\Psi_S^t$.
22: **return** $\mathcal{X}^{(t)}, \Psi_S^t$.

## IV. Trace-driven Evaluation Results

In this section, we evaluate the effectiveness of the proposed framework with extensive simulations on a real-world scenario. The experiments are conducted on a dataset which covers a 6.2 $km^2$ central business district region in Melbourne, Australia [16]. The dataset contains the geographic information of real-world BSs and users, including longitude and latitude, which is obtained by Australian Communications and Media Authority. We consider an ultra-dense MEC network with $\mathcal{S}$ = 20 SBSs and 400 users (default) are distributed in the region. The maximum number $N_s$ of the served users for each SBS is set in the range of [20, 30]. We consider the number of popular contents as $F = 200$ and their sizes $\{d_f\}$ are randomly set in [5, 20] Mbits. The storage size of each SBS is set as the percentage $\theta$ of the total content size. We use the Random-Waypoint mobility model to generate the users' moving path, and set the $P_{u,s}^{init}$ to be the frequency of time slots that the user start walking from the BS$_s$. Likewise, we calculate the $M_{s,s'}$ as the total time slots that the user moves from BS$_s$ to BS$_{s'}$, divided by the time slots where the user within the service coverage of BS$_s$. The per-MB transmission cost is set as $\delta_C$ = 0.20/1024 \$/MB, $\delta_D$ = 0.10/1024 \$/MB and $\delta_E$ = 0.25/1024 \$/MB, respectively. We simulate 20 time slots and the interval of a time slot length is equal to 90 seconds. The algorithm parameters are set as $\tau = 5$, $\eta = 3$ and $\omega = 0.6$, respectively. The Zipf distribution parameter $\beta$ is set as 0.56. Both the function $\mathbb{Z}_1\{.\}$ and $\mathbb{Z}_2\{.\}$ use the current value to divide the maximum value, so the time-average normalized system costs are in the range of (0, 2].

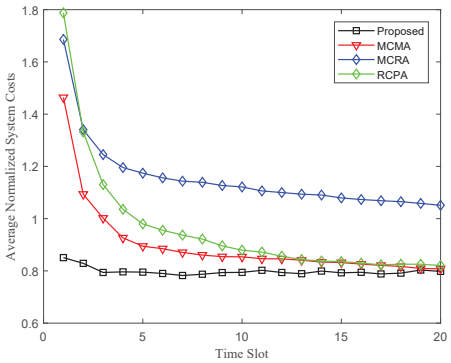

Fig. 3. Average normalized system costs in each time slot.

To evaluate our proposed framework, we consider the following three baseline schemes as: 1) Most popular Caching and Most remaining Association strategy (MCMA), where each SBS stores the most popular contents, and each user is associated with the SBS who has the largest remaining number of association; 2) Most popular Caching and Random Association strategy (MCRA), where each SBS stores the most popular contents and users are associated with SBSs randomly; 3) Random Caching and Proposed Association strategy (RCPA), where each SBS caches the contents randomly and then the proposed Algorithm 2 is applied.

Fig. 3 compares the performance of the considered schemes in terms of the average normalized system costs at each time slot. The average normalized system costs is an important metric to evaluate the long-term system costs performance. Seen from Fig. 3, the proposed scheme has lower system costs at each moment compared with the other schemes, because the proposed scheme optimizes the content caching based on the users' movement information and historical user association policy. To a certain degree, the proposed scheme is more capable to capture users' real-time requests and mobility.

In Fig. 4, we evaluate the impacts of different numbers of users in the considered schemes. It can be observed that the average normalized system costs achieved by the proposed scheme decrease gradually, while the other schemes are at a relatively high level. To explain it, as the number of users increases, the lazy re-association strategy can swap the association between users and satisfy the user requests by the SBSs rather than the CS. It proves that the proposed scheme can perform better when facing the high density of user requests.

Fig. 5 compares the performance on the average normalized system costs versus the percentage $\theta$ of the cache size of SBSs.

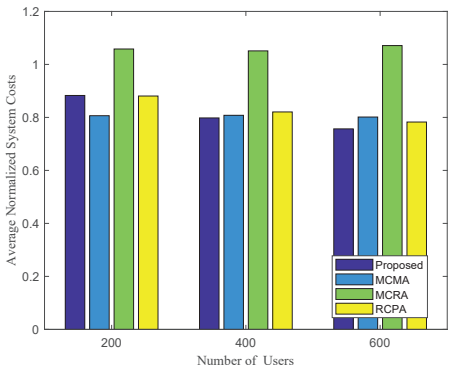


Fig. 4. Average normalized system costs versus different numbers of users.

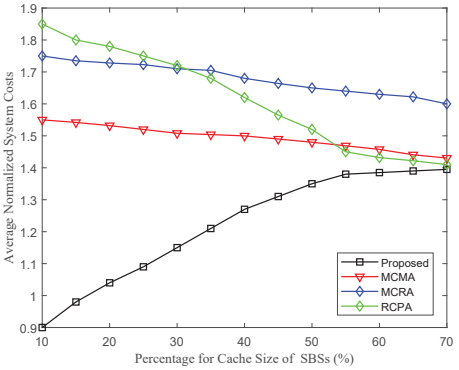


Fig. 5. Average normalized system costs versus different ratio $\theta$.

Seen from Fig. 5, the proposed scheme can achieve much better performance when the value of $\theta$ is small. Especially when the value of $\theta$ is less than 15%, its performance is nearly twice than other schemes. As the percentage $\theta$ increases, the SBSs can cache more contents and the impact of the content caching policy is becoming less. To explain it, the difference in system costs mainly comes from handover costs rather than two aspects. As a result, the performance gap between the proposed scheme and other schemes is slowly decreasing.

## V. Conclusion

In this paper, we have investigated the issue of joint mobility-aware content caching and user association to minimize the system costs for ultra-dense MEC networks by considering the users' and SBSs' characteristic constraints. To tackle the formulated complex optimization problem, we have decomposed the long-term original problem into a series of short-term one-slot subproblems. Then we have optimized these subproblems in two phases. The first phase is related to content caching, and we have proposed a mobility-aware online content caching strategy for solving it. The second phase is about user association, and we have proposed a lazy re-association algorithm based on matching theory to address it. Trace-driven evaluation results have demonstrated the superior performance of the proposed framework on reducing the system costs in the considered ultra-dense MEC network.

## Acknowledgment

This work is supported in part by National NSFC (Grants No. 61902044 and 61672117), National Key R & D Program of China (Grants No. 2018YFB2100100 and 2018YFF0214700), Chongqing Research Program of Basic Research and Frontier Technology (Grant No. cstc2019jcyj-msxmX0589), Key Research Program of Chongqing Science & Technology Commission (Grants No. CSTC2017jcyjBX0025 and CSTC2019jscx-zdztzxX0031), and Fundamental Research Funds for the Central Universities (Grant No. 2020CDJQY-A022), Chinese National Engineering Laboratory for Big Data System Computing Technology, and Canadian NSERC.

## References

[1] C. V. N. Index, "Global mobile data traffic forecast update 2014-2019 white paper," *Cisco, San Jose*, 2015.

[2] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, Aug. 2017.

[3] X. Li, X. Wang, P.-J. Wan, Z. Han, and V. C. Leung, "Hierarchical edge caching in device-to-device aided mobile networks: Modeling, optimization, and design," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 8, pp. 1768–1785, Jun. 2018.

[4] K. S. Khan and A. Jamalipour, "Coverage analysis for multi-request association model (mram) in a caching ultra-dense network," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3882–3889, Jan. 2019.

[5] W. Jiang, G. Feng, S. Qin, and Y.-C. Liang, "Learning-based cooperative content caching policy for mobile edge computing," in *Proc. IEEE ICC*, Jul. 2019, pp. 1–6.

[6] Y. Cui, F. Lai, S. Hanly, and P. Whiting, "Optimal caching and user association in cache-enabled heterogeneous wireless networks," in *Proc. IEEE GLOBECOM*, Feb. 2016, pp. 1–6.

[7] W. Teng, M. Sheng, K. Guo, and Z. Qiu, "Content placement and user association for delay minimization in small cell networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10 201–10 215, Aug. 2019.

[8] B. Dai and W. Yu, "Joint user association and content placement for cache-enabled wireless access networks," in *Proc. IEEE ICASSP*, May. 2016, pp. 3521–3525.

[9] X. Liu, J. Zhang, X. Zhang, and W. Wang, "Mobility-aware coded probabilistic caching scheme for mec-enabled small cell networks," *IEEE Access*, vol. 5, pp. 17 824–17 833, Aug. 2017.

[10] J. Song and W. Choi, "Mobility-aware content placement for device-to-device caching systems," *IEEE Trans. Wireless Commun.*, vol. 18, no. 7, pp. 3658–3668, May. 2019.

[11] M. Chen, Y. Hao, L. Hu, K. Huang, and V. K. Lau, "Green and mobility-aware caching in 5g networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 12, pp. 8347–8361, Oct. 2017.

[12] 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, Jan. 2020.

[13] I. Keshavarzian, Z. Zeinalpour-Yazdi, and A. Tadaion, "Energy-efficient mobility-aware caching algorithms for clustered small cells in ultra-dense networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6833–6846, May. 2019.

[14] X. Wu, Q. Li, V. C. Leung, and P.-C. Ching, "Joint fronthaul multicast and cooperative beamforming for cache-enabled cloud-based small cell networks: An mds codes-aided approach," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4970–4982, Aug. 2019.

[15] E. Bodine-Baron, C. Lee, A. Chong, B. Hassibi, and A. Wierman, "Peer effects and stability in matching markets," in *Proc. Springer SAGT*, vol. 68, no. 7, Oct. 2011, pp. 117–129.

[16] Q. He, G. Cui, X. Zhang, F. Chen, S. Deng, H. Jin, Y. Li, and Y. Yang, "A game-theoretical approach for user allocation in edge computing environment," *IEEE Trans. Parallel and Distributed Systems.*, vol. 31, no. 3, pp. 515–529, Sep. 2019.