IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 16, 2018, accepted November 12, 2018, date of publication January 1, 2019,
date of current version January 11, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2885571

Cache Aware User Association for Wireless Heterogeneous Networks

RIM HAW[1], S. M. AHSAN KAZMI[1,2], KYI THAR[1], MD GOLAM RABIUL ALAM[1,3], (Member, IEEE), AND CHOONG SEON HONG[1], (Senior Member, IEEE)

[1]Department of Computer Science and Engineering, Kyung Hee University, Yongin-si 17104, South Korea
[2]Network, Cyber, and Information Security Lab, Secure System and Network Engineering, Innopolis University, 420500 Tatarstan, Russia
[3]Department of Computer Science and Engineering, BRAC University, Dhaka 1212, Bangladesh

Corresponding author: Choong Seon Hong (cshong@khu.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2017R1A2A2A05000995).

ABSTRACT The proliferation of network devices and novel bandwidth hungry applications over the existing network imposes novel challenges in terms of fulfilling the users' requirements. Dense deployment of small cells is thought to be a promising solution to fulfill these requirements. However, the user association in such dense networks becomes challenging and can greatly affect the network performance, as a user in such dense deployments can be connected to any of the available base stations. Traditionally, the user association has been performed based on the signal strength, however, such an approach does not apply when taking into account novel bandwidth hungry applications. Moreover, in recent years, a successful paradigm has been proposed to handle such bandwidth hungry applications, i.e., caching at small cell base stations. In this paper, we aim to solve this joint problem of user association and content caching in a dense small cell setting. To solve this problem, we present a novel iterative scheme that uses matching theory and a learning approach to find a suboptimal solution of the joint NP hard problem. Note that the user association and cache placement are strongly coupled, i.e., the association of users at a base station will determine the cache placement at base stations and the availability of cache at base stations will force the users to change their associations. Simulation results show that the proposed scheme (i.e., cache aware user association) significantly outperforms the cache unaware scheme and achieves a performance gain of up to 31% in terms of normalized utility and saves up to twice the backhaul bandwidth. Moreover, the proposed scheme also achieves up to 82% of the utility obtained by the optimal solution.

INDEX TERMS User association, heterogeneous networks, matching games, caching, wireless small cell networks.

I. INTRODUCTION

The mobile data traffic is witnessing unprecedented growth due to the proliferation and wide acceptance of smart devices and novel bandwidth hungry applications such as multimedia streaming and mobile TV [1]–[3]. To cope with the traffic growth and fulfill these stringent novel applications requirements, dense deployment of small base stations (SBSs) is required to operate in conjunction with the existing macro base station (MBS) [5]–[7]. Enormous benefits in terms of network capacity and spectral efficiency can be achieved via dense deployment of SBSs [8], however, to reap the full benefits of dense deployment several technical challenges in terms of backhaul management and user association need to be addressed.

One promising approach for backahaul management is caching popular contents at local base stations (BSs) [2], [3]. Enabling caching at the BSs (MBS and SBSs) can significantly reduce the amount of re-downloading contents from the original content servers, which leads to lower backhaul load [3], [4]. Note that not all contents can be cached at the BS, as it has limited cache capacity compared to the amount of total contents. Therefore, each BS should store only popular contents within its limited cache storage to efficiently utilize its cache storage. In reality, it is very challenging to know whether a content is popular or not because the popularity can temporally and/or spatially vary. Thus, some assumption is typically used, in which the popularity of contents follows a distribution (e.g. a Zipf distribution) [9].

Depending on this assumption, many researchers proposed several caching models and cache decision algorithms[1] to efficiently store popular contents at wireless edge nodes [9]–[12]. In practice, this assumption can be invalid because the popularity of contents is dynamically changing depending on different factors (e.g., events, type of content, and the lifespan of the content). Therefore, a content popularity prediction scheme is needed to support to the cache decision process to be able to efficiently cache the contents.

User association in a dense BS setting [13] also becomes very crucial as now a user can be associated with a number of BSs based on its channel condition parameters, i.e., received signal strength (RSSI) [12]. Moreover, in a cache enabled setting, it is more appropriate both for the BS and the user to be associated to a specific BS which not only provides good channel conditions, but also has the user's request cached locally. This will reduce the backhaul load and enhance the caching efficiency [14]–[19]. Thus, it is of crucial importance to devise a user association strategy by taking both the channel condition and the BSs' cache into consideration. Moreover, the introduction of novel 5G applications such as enhanced mobile broadband (eMBB) and ultra reliable low latency communication (URLLC) demands that we store contents for a short period of time opposed to the traditional long term based caching approach typically used in content centric networks. Therefore, in this work, we use a small time-scale for caching contents at the BSs and thus evaluating the popularity based on short period of caching.

A. RELATED WORKS

Caching in wireless networks has received significant attention in several recent works in which the primary goal is to develop an efficient caching decision to improve the cache hit ratio. Typically, the caching decision can be categorized into two categories: i) reactive caching and ii) proactive caching. In *reactive caching*, MBSs/SBSs make the cache decision only when the request for a particular content arrives and cache the content based on its popularity [20]–[23]. In reactive caching, the caching gain is not only dependent on the content's popularity prediction, but also on the cache replacement process. The cache replacement process is responsible to replace old content with new incoming content when the cache storage of an SBS/MBS is full. Similar to the works in [9]–[12], Thar et al. [20]–[22] assumed the content's popularity followed a Zipf distribution and applied consistent hashing as a foundation of reactive caching decision to improve the cache utilization. These works ignored the time varying aspects of popular contents and assumed that the contents' popularity will remain unchanged. However, in a practical scenario, this assumption does not hold. Thus, Li *et al.* [23] proposed a caching scheme that learns the content's popularity from the dataset. However, the prediction process needs high computing resources, which are generally not available at the low cost small-cell base station.

In *proactive caching*, MBSs/SBSs pro-actively predict a content's popularity based on the user request history and cache popular content before any user request is made [24], [25]. For proactive caching, multiple MBSs/SBSs can jointly cache the popular content to maximize the caching gain, where the caching gain is reduced depending on prediction errors. Thus, proactive caching may perform worse than reactive caching when the prediction error is high. Zeydan *et al.* [24] investigated proactive content caching with popularity prediction for wireless networks, in which the authors applied bigdata analytics tool and machine learning to get efficient caching decision. Another proactive caching approach is studied in [25] for the cloud radio access networks (CRANs) in which the authors applied echo state networks (ESNs) to predict each user's content request distribution and mobility pattern to support cache decision processes. The usage of ESNs is perfectly fine in the CRANs architecture but it is not suitable for distributed low cost SBSs, which require a prediction scheme with low computational complexity.

In this work, we focus on developing a distributed scheme that can scale with network size for the joint cache placement and user association problem. Some recent works such as [14] have presented an efficient solution with the aid of McCormick envelopes and Lagrange partial relaxation for the joint caching and user association problem in heterogeneous cellular networks to minimize the access delays. Similarly, He *et al.* [15] present an efficient distributed heuristic algorithm for cache placement and user association in heterogeneous networks to minimize the power consumption of the system. Other notable works in heterogeneous cellular networks that consider the joint caching and user association problem can be found in [17]–[19]. Although these aforementioned works have significantly enhanced different heterogeneous network performance parameters such as access delays, users' quality of service and energy efficiency, they did not account for the prediction of content popularity when making the caching decision. Through the prediction of contents' popularity, a more efficient caching decision can be made that can significantly improve the network's performance [16].

Therefore, content popularity prediction plays an important role and supports the cache decision process to improve the cache hit as well as best utilize the limited cache space. Content popularity can be defined as the ratio of the number of requests for a particular content to the total number of requests from users, usually obtained for a certain region during a given period of time. Predicting the popularity of video content has been extensively studied in the recent literature [26]–[33], while few works consider how to integrate popularity forecasting into caching.

Moreover, various prediction solutions are proposed based on time series models such as the auto regressive integrated moving average [34], regression model [35] and classification models [36] without combining with the cache

[1] Algorithms to decide whether to store the new content or remove the cached content

decision process. To dynamically adapt the changing popularity of contents, [38], [43] proposed several learning methods. Blasco and Gündüz [38] investigated the trade-off between the exploration and exploitation phase of the learning algorithm, which learns the generated data. Tekin and Schaar [43] proposed a context aware popularity prediction scheme, which also exploits the similarities among users' profiles. Even though those prediction schemes are good at predicting the content's popularity, such prediction schemes require a high computation capacity, which are not suitable to be implemented at small-cell base stations.

B. CONTRIBUTIONS AND ORGANIZATIONS

In this work, we aim to address the joint user association and cache placement problem for a dense heterogeneous network. We aim to present a distributed and scalable solution for the joint problem that can enhance the overall network performance in terms of the optimal set of users (maximizing the sum rate) and maximizing the cache hit ratio. We present a novel two phase iterative approach that can efficiently address the joint problem. Initially, we consider a random placement of contents in the cache of each BS that would be broadcasted in the network. Then, based on this information, we apply the two-sided matching game to address the user association aspect of the problem in phase I. Next, based on the association, we learn and predict the contents' popularity by applying the Autoregressive Integrated Moving Average (ARIMA) [39], [40], the most common methods for time series forecasting. ARIMA technique (discussed details in Sec. III-B.1) provides us to predict the future popularity of content by examining the differences between values (content request counts) in the historical time series data. These contents' popularity is then used to make a caching decision at each BS in phase II. Based on the new caching decision, the users re-associate using the phase I game theoretic approach. This iterative process is stopped once convergence is achieved. In summary, our key contributions include the following:

- First, we formulate the joint problem of the user association and cache placement with an objective to maximize the network sum rate and cache hit ratio subject to the limited cache space and wireless resources. The formulated problem is a mixed-integer optimization problem that is challenging and requires exponential computation efforts to obtain the optimal solution.
- Second, in order to solve this joint problem, we decompose the joint problem into two sub-problems, i.e., user association and cache placement problems. A novel algorithm based on two-sided matching theory is presented to solve the combinatorial user association problem. Moreover, we also prove the stability and convergence of the proposed solution. To solve the cache placement problem, we use ARIMA to predict the content popularity and then make the caching decision.
- Finally, we iteratively solve both of these subproblems to obtain the solution of our joint problem. Moreover, we also prove that the proposed solution achieves a suboptimal solution for the joint problem.

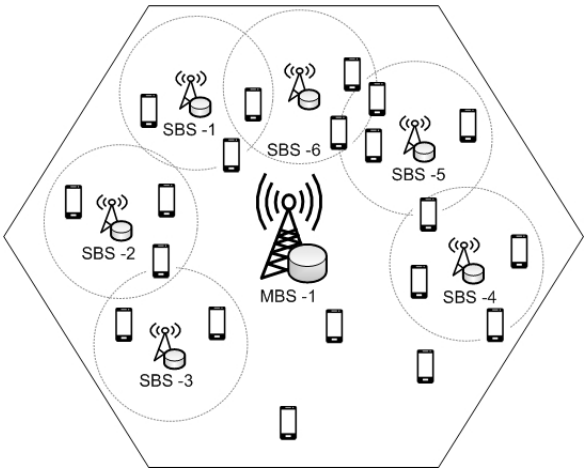


FIGURE 1. System Model.

The rest of this paper is organized as follows. Section II presents the system model and problem formulation. Section III describes in detail our solution approach, i.e., how we decompose and map the proposed optimization problem into a matching theory setting and how we apply the content prediction via ARIMA and make the caching decision. In Section IV, we present the simulation results analysis to validate the performance of our proposed solution. Finally, conclusions are drawn in Section V.

II. SYSTEM MODEL AND PROBLEM FORMULATION

We consider the downlink of a cellular network that consists of a single macrocell base station (MBS) and a set of SBSs located under its coverage, as shown in Fig. 1. We represent the set of base stations by $\mathcal{J} = \{0, 1, 2, ..., J\}$, where the index 0 represents the MBS. The set of users (UEs, i.e., macro and small cell UEs) are denoted by $\mathcal{U} = \{1, 2, ..., U\}$. In this model, the spectrum is divided into orthogonal frequency subbands $\mathcal{S}$, and each SBS j is allocated a subband s_j of multiple resources. Moreover, we assume that each subband s_j has the same cardinality, i.e., $|s_j| = \frac{|R|}{|J|}$, where R represents the total number of resources[2] owned by the operator. Furthermore, both MBS and SBSs allocate resource from their subband s_j to each associated UE u.

A. LINK MODEL AND ASSUMPTIONS

In our model, we assume that all SBSs and the MBS transmit using an equal power for every resource. However, the MBS and SBSs have their own and different power budgets. Thus, the interference power on each resource is constant such that the interference from other BSs is absorbed into the background noise σ^2. For user association optimization, we introduce a binary variable $x_{u,j}$ as follows:

$$x_{u,j} = \begin{cases} 1, & \text{if UE } u \text{ is associated to BS } j, \\ 0, & \text{otherwise.} \end{cases}$$

[2]One resource corresponds to one subcarrier or subchannel of the LTE network.

TABLE 1. Table of notations.

Notation	Description
$\mathcal{J}, j, j$	Set, number and index of base station
c_j, C	Cache capacity of BS j and total cache capacity
$\mathcal{F}, F, f$	Set, number, element of contents
$u_j(\cdot)$	Utility function or profit of base station j
$U(\cdot)$	Utility function or profit of the whole network
$x_{u,j} \in \{0,1\}$	User association variable (UE u is attached with SBS j)
$y_{j,f} \in \{0,1\}$	Cache decision variable for content f at BS j
$r^t_{j,f}$	Number of incoming requests for content f at base station j
$\pi^t_{j,f}, \widetilde{\pi}^t_{j,f}$	Actual popularity score and predicted popularity score of content f at base station j

We always set $x_{u,j} = 0$ for any UE u, which is not associated with a BS j. Then, the received signal to noise ratio (SINR) pertaining to the transmission of BS j to UE u over a resource r with transmit power P^r_j is:

$$\gamma^r_{u,j} = \frac{P^r_j g^r_{u,j}}{\sigma^2}, \tag{1}$$

where $g^r_{u,j}$ represents the channel gain between BS j and UE u. Note that in the considered model, we consider orthogonal resources at each BS, therefore, we do not consider inter-BS interference.[3] Then, the data rate of UE u associated with BS j on resource r will be given by:

$$R^r_{u,j} = W^r \log(1 + \gamma^r_{u,j}), \tag{2}$$

where W^r is the bandwidth of the resource r.

B. CACHING MODEL AND ASSUMPTIONS

In our caching model, the set of base stations $\mathcal{J}$ are equipped with a cache storage of limited capacity. Let the cache capacity at each base station be denoted as c_j. Thus, each base station can store limited contents such that the cached content size is less than its cache capacity. We denote the total cache capacity of all base stations by $C = \sum_{j\in\mathcal{J}} c_j$. Moreover, we assume the content server is located at the core of the cellular system that contains all of the content chunks represented by the set $\mathcal{F} = \{1, 2, \ldots, F\}$. Note that, if a requested content is unavailable at the BS's local cache, it needs to be provided by the content server via the backhaul link. For caching related optimization, we introduce a binary variable $y_{f,j}$ as follows:

$$y_{f,j} = \begin{cases} 1, & \text{if content } f \text{ is cached at BS } j, \\ 0, & \text{otherwise.} \end{cases}$$

Note that, if a content f is not cached at BS j, we set $y_{f,j} = 0$. Thus, the base station j can provide content f from its local cache storage when $y_{f,j} = 1$.

The arriving request for content f can be denoted as r_f. For each arriving request r_f, base station j first needs to check if the requested content f is present in its cache storage or not. The base station j provides the content f to the UE directly, if the requested content f is in its cache storage. Otherwise, the base station j retrieves the content f from the content sever. At this point, the base station j also decides whether to store the content f in its local cache based on the number of arriving request at time t (which can also be denoted as the popularity score of content f). The main goal of a caching decision process is to select the best contents among the list of contents such that the cache hit is maximized.

In order to make an efficient caching decision, the future popularity score of a content f is required for time $t + 1$ at the current decision making time t. Therefore, a prediction scheme can play a crucial role in calculating the future popularity score of a content f for the time $t+1$. Without prediction schemes, it is not possible to get the future popularity scores of the contents. Motivated by the aforementioned challenges, in this work, we propose a novel approach to predict the content's popularity score based on an autoregressive integrated moving average (ARIMA), which will be discussed in detail in Section III-B. The popularity score of content f at the base station j at time t can be denoted as π_{jf} and is calculated as [22]

$$\pi^t_{j,f} = \frac{r^t_{j,f}}{\sum_{f\in\mathcal{F}} r^t_{j,f}}, \tag{3}$$

where $r^t_{j,f}$ is the number of incoming requests for content f at base station j at time t and $\sum_{f\in\mathcal{F}} r^t_{j,f}$ is the total number of arriving requests for all contents at base station j at time t. Therefore, the predicted popularity score or future popularity score of content f becomes $\widetilde{\pi}^t_{j,f}$.

C. PROBLEM FORMULATION

Our goal is to design a mechanism that can maximize the utility of the network in such a way that each SBS j stores popular contents which can be used to serve the associated BS UEs. Therefore, we define the utility function of a BS j as follows:

$$U_j(\boldsymbol{x}, \boldsymbol{y}) = \sum_{u\in\mathcal{U}} \sum_{r\in\mathcal{R}} x_{u,j} R^r_{u,j} + \omega \sum_{f\in\mathcal{F}} y_{f,j}, \tag{4}$$

where the first and second terms denote the data rate of all associated UEs and the list of contents cached at BS j, respectively. Here, ω represents the weight parameter that characterizes the trade-off between the BS's sum rate and cached contents. In order to maximize the utility function given in (4), each BS should select a set of UEs whose achievable sum rate is higher when associated to it. Moreover, it should also consider that the contents requested by these associated UEs are mostly present in the local cache to avoid fetching the contents from the content server. Thus, our joint optimization problem involves user association and cache placement decisions. Furthermore, the cache placement involves the prediction

[3] We aim to focus on more complicated interference scenarios in our future work.

of a content's popularity score and finding best contents to be cached.

$$P: \underset{\boldsymbol{x},\boldsymbol{y}}{\text{maximize:}} \sum_{j\in\mathcal{J}} U_j(\boldsymbol{x},\boldsymbol{y}) \tag{5a}$$

$$\text{subject to:} \sum_{f\in\mathcal{F}} y_{f,j} \le c_j, \quad \forall j \in \mathcal{J}, \tag{5b}$$

$$\sum_{u\in\mathcal{U}} x_{u,j} \le |s_j|, \quad \forall j \in \mathcal{J}, \tag{5c}$$

$$\sum_{j\in\mathcal{J}} x_{u,j} \le 1, \quad \forall u \in \mathcal{U}, \tag{5d}$$

$$x_{u,j}, y_{f,j} \in \{0,1\}, \quad \forall f, u, j. \tag{5e}$$

The objective function here represents the network utility for all BSs, where the first constraint ensures that the cache capacity is not violated. The second constraint ensures that the number of UEs associated are less than the available resources at the BSs while the third constraint ensures that a UE can be only associated to a single BS. Finally the user association and cache placement variables are represented by the integer constraints. Unfortunately, the aforementioned mixed-integer optimization problem **P** is non-trivial due to the combinatorial nature of user association and cache placement decision variables [45]. Moreover, obtaining an optimal solution via exhaustive search will incur heavy computational overhead and would require a central coordinator. This approach is challenging to adopt for a practical setting with large numbers of BSs, UEs and contents. Furthermore, the cache placement decision relies on calculating the future popularity score of all contents through a prediction scheme [44]. Applying prediction schemes at a central coordinator will incur huge message exchanges, as all BSs would then be required to report to the central coordinator. Thus, we decompose the original problem into two subproblems namely the user association (UA) and cache placement (CP). Through decomposition, we can solve our problem in a distributed fashion at each BS and do not require any central controller. Our designed distributed approach will be presented in the next section.

III. JOINT USER ASSOCIATION AND CACHE PLACEMENT

In this section, we present our solution approach for the joint user association and cache placement problem **P**. In order to have a distributed solution, we decompose the joint problem into two subproblems which would be solved at each BS. The first subproblem **UA** will solve the user association (UA) problem for a given cache placement, and the second subproblem **CP** will find a solution for the cache placement (CP) problem given the associated UEs. Then, we iteratively solve these two subproblems to find a suboptimal solution of our joint problem **P**. The subproblem **UA** for user association at each BS j can be stated as follows:

$$UA: \underset{\boldsymbol{x}}{\text{maximize:}} U_j(\boldsymbol{x},\boldsymbol{y}) \tag{6a}$$

$$\text{subject to:} \sum_{u\in\mathcal{U}} x_{u,j} \le |s_j|, \quad \forall j \in \mathcal{J}, \tag{6b}$$

$$\sum_{j\in\mathcal{J}} x_{u,j} \le 1, \quad \forall u \in \mathcal{U}, \tag{6c}$$

$$x_{u,j} \in \{0,1\}, \quad \forall u, j. \tag{6d}$$

In problem **UA**, our goal is to maximize the utility by associating the optimal UEs given the set of contents cached at BS j. Note that, **UA** is still combinatorial in nature and finding a solution for a large set of BSs and UEs would be challenging [41], [42], [44]. Thus, we adopt a solution based on matching theory to solve the above problem because of its ability to tackle combinatorial problems and achieve a distributed solution [8], [44]. The benefits of matching theory come from the distributed nature of control in the system, which is crucial for designing distributed solutions. Furthermore, matching theory allows each player to define their individual utilities depending on the local information.

A. MATCHING THEORY BASED USER ASSOCIATION

In our game there are two disjoint sets of agents, the set of UEs, $\mathcal{U}$, and the set of BSs, $\mathcal{J}$. Each UE $u \in \mathcal{U}$ has a strict, transitive, and complete preference profile $\mathcal{P}_u$ of UE u defined over the BS. Note that, in this game from (6c), a UE u can only be associated with one BS. However, a BS j can accommodate a number of UEs based on its capacity or quota, i.e., (6b). Therefore, the preference profile $\mathcal{P}_j$ of BS j is defined over the set of UEs $\mathcal{U}$. Thus, our design corresponds to the *one-to-many matching* given by the tuple $(\mathcal{J}, \mathcal{U}, q_j, \succ_{\mathcal{J}}, \succ_{\mathcal{U}})$. Here, $\succ_{\mathcal{U}} \triangleq \{\succ_u\}_{u\in\mathcal{U}}$ and $\succ_{\mathcal{J}} \triangleq \{\succ_r\}_{r\in\mathcal{J}}$ represent the sets of preference relations of UEs and BSs, respectively. Formally, we define the matching game as follows:

Definition 1: A *matching* μ is defined by a function from the set $\mathcal{U} \cup \mathcal{J}$ into the set of elements of $\mathcal{U} \cup \mathcal{J}$ such that:

(i) $|\mu(u)| \le 1$ and $\mu(u) \in \mathcal{J}$,
(ii) $|\mu(j)| \le q_j$ and $\mu(j) \in 2^{\mathcal{U}} \cup \phi$,
(iii) $\mu(u) = j$ if and only if u is in $\mu(j)$,

where, q_j denotes the quota of BS j and $|\mu(.)|$ denotes the cardinality of the matching outcome $\mu(.)$. The first two conditions of Definition 1 represent constraints (6c) and (6b) in the **UA** problem, respectively, where q_j represents the total quota of BS j, i.e., $|s_j|$. Here, $\mu(.) = \phi$ means that the agent is unmatched.

1) PREFERENCES OF THE PLAYERS

In our formulated game, both sides need to rank each other using the preference profiles. Matching is performed on the basis of preference profiles that is built by both sides to rank each other. Then, potential matchings can be performed based on the local information of each player. In our game, a UE u ranks all BSs based on the following

preference function:

$$\mathcal{U}_u(j) = R^r_{u,j} + \omega \sum_f \tilde{y}_{f,j}, \quad \forall u \in \mathcal{U}. \tag{7}$$

Through (7), we can rank all BSs based on the achievable rate and the number of usable contents cached in it, where ω represents a weight parameter that quantifies the importance of cached contents at a BS. Note that $\tilde{y}_{f,j}$ represents the usable contents for a UE u. Usable contents are those contents which a UE u will request in the next time slots. Note that this information is only available at the UEs. The design of the utility given in (7) reflects that a UE will benefit more from a BS j that has more contents of interest cached in its local cache compared to other BS with similar channel conditions. Similarly, for the BSs side, each BS j also ranks the UEs according to the following preference function:

$$\mathcal{U}_j(u) = R^r_{u,j}, \quad \forall j \in \mathcal{J}. \tag{8}$$

From (8), we can rank all UEs based on their achievable rate. This utility implies that the BSs provides less utility to the UEs that have a lower achievable rate. Once the preference profiles of both sides are built, our goal is to seek a *stable matching* solution, which is a key solution concept. Note that to find a stable matching, the deferred-acceptance algorithm cannot be employed for our game [44]. In our game, a UE can be allowed a number of resources depending on its demand and channel conditions. Thus, we have to tackle the additional challenge of a dynamic quota [8]. Through a dynamic quota, a BS may allow a variable number of UEs to be associated until the constraint (6b) is not violated. Therefore, formally the blocking pair for this game can be defined as:

Definition 2: A matching μ is *stable* if there exists no blocking pair $(A', j) \in 2^U \cup J$ with $A' \neq \phi$, such that, $j \succ_u \mu(u), \forall u \in A'$ and $(A \cup A') \succ_j \mu(j), A \subseteq \mu(j)$, where $\mu(u)$ and $\mu(j)$ represent, respectively, the current matched partners of BSs and UEs.

Definition 2 is based on the following intuition [46]: a pair (A', j) will block a matching μ, if BS j is willing to accept a UE in A', possibly after rejecting some of its currently matched UEs in $\mu(j)$, i.e., $A \subseteq \mu(j)$ and all UEs $u \in \mathcal{U}$ prefer j over their current match $\mu(u)$. In the formulated game, it can be ensured that for any stable solution, no matched BS j would benefit from deviating from their associated UEs u with new UEs u'. A matching is stable if no blocking pair exists. Next, we present our novel matching based user association algorithm.

2) PROPOSED USER ASSOCIATION ALGORITHM

Next, we present a novel and stable user association algorithm presented in Alg. 1. The algorithm starts by using the local information to build the preference profiles (lines 1-2). At each iteration t, each agent first calculates its utility build of its respective preference profiles. Then, each UE u proposes to its most preferred BS j according to its preference profile P_u (line 5). A proposal also contains the demand of UE u. On receiving the proposal, each

Algorithm 1 UA via Distributed Matching Game

1: **input**: $\mathcal{P}_u, \mathcal{P}_j, \forall u, j$
2: **initialize**: $t = 0$, $\mu^{(t)} \triangleq \{\mu(u)^{(t)}, \mu(j)^{(t)}\}_{u \in \mathcal{U}, j \in \mathcal{J}} = \emptyset$, $q_j^{res(t)} = q_j^{\max}$, $\mathcal{K}_j{}^{(t)} = \emptyset$, $\mathcal{P}_u{}^{(0)} = \mathcal{P}_u$, $\mathcal{P}_j{}^{(0)} = \mathcal{P}_j, \forall u, j$
3: **repeat**
4: $t \leftarrow t + 1$
5: **for** $u \in \mathcal{U}$ with BS j as its preferred via $\mathcal{P}_u{}^{(t)}$ **do**
6: **while** $u \notin \mu(j)^{(t)}$ and $\mathcal{P}_u^{(t)} \neq \emptyset$ **do**
7: **if** $q_j^{res(t)} \geq l_u^j$, **then**
8: $\mu(j)^{(t)} \leftarrow \mu(j)^{(t)} \cup \{u\}$;
9: $q_j^{res(t)} \leftarrow q_j^{res(t)} - l_u^j$;
10: **else**
11: $\mathcal{K}'_j{}^{(t)} = \{u' \in \mu(j)^{(t)} | u \succ_j u'\}$;
12: $u_{lp} \leftarrow$ the least preferred $u' \in \mathcal{K}'_j{}^{(t)}$;
13: **while** $(\mathcal{K}'_j{}^{(t)} \neq \emptyset) \cup (q_j^{res(t)} < l_u^j)$ **do**
14: $\mu(j)^{(t)} \leftarrow \mu(j)^{(t)} \setminus \{u'\}$;
15: $\mathcal{K}'_j{}^{(t)} \leftarrow \mathcal{K}'_j{}^{(t)} \setminus \{u_{lp}\}$;
16: $q_j^{res(t)} \leftarrow q_j^{res(t)} + l_u^j$;
17: $u_{lp} \leftarrow$ the least preferred $u' \in \mathcal{K}'_j{}^{(t)}$;
18: **if** $q_j^{res(t)} \geq l_u^j$, **then**
19: $\mu(j)^{(t)} \leftarrow \mu(j)^{(t)} \cup \{u\}$;
20: $q_j^{res(t)} \leftarrow q_j^{res(t)} - l_u^j$;
21: **else**
22: $u_{lp} \leftarrow u$;
23: $\mathcal{K}_j{}^{(t)} = \{k \in \mathcal{P}_j{}^{(t)} | u_{lp} \succ_j k\} \cup \{u_{lp}\}$;
24: **for** $k \in \mathcal{K}_j{}^{(t)}$ **do**
25: $\mathcal{P}_k{}^{(t)} \leftarrow \mathcal{P}_k{}^{(t)} \setminus \{j\}$;
26: $\mathcal{P}_j{}^{(t)} \leftarrow \mathcal{P}_j{}^{(t)} \setminus \{k\}$;
27: **until** $\mu^{(t)} = \mu^{(t-1)}$
28: **output**: $\mu^{(t)}$

BS j calculates the required resources (i.e., l_u^j) to fulfill the UE's demand [46]. This can result in either of the following two cases. In this first case, a BS j may have enough resources $q_j^{res(t)}$ to accommodate the UE u. This results in a matching between the proposing UE u and BS j (lines 7-9). The second case is activated if enough resources are not available, i.e., $q_j^{res(t)} < l_u^j$ (lines 10). This means the quota of a BS j is already occupied and full. In this case, the BS j finds all of its current matched u' which have a lower ranking than the proposing UE u according to its preference profile $\mathcal{P}_j{}^{(t)}$ (lines 11-12). Each least preferred UE $u_{lp} \in \mathcal{K}'_j{}^{(t)}$ is then sequentially rejected, and the quota of the BS is updated, i.e., $q_j^{res(t)}$ until either u can be admitted or there is no additional u' to reject (lines 13-17). After rejecting all $u' \in \mathcal{K}'_j{}^{(t)}$, if BS j still has an insufficient quota to admit UE u, then u is also rejected and u is set to the least preferred u_{lp} (lines 18-22), otherwise it is accepted. All these agents (i.e., rejected UEs and BS) then update their preference profiles and enter into the next iteration (lines 23-26). Through this process, it is guaranteed that no blocking pair will exist as we remove any less preferred UEs from the matching, even

if a BS has sufficient quota to admit it, which is crucial for the matching stability of our design. In this next iteration, all rejected UEs again propose to the next preferred BSs. Once all UEs have either been accepted by a BS or rejected by all BSs, the matching process will terminate. Note that here, the matching terminates when the results of two consecutive iterations t remain unchanged (line 27). Moreover, the output $\mu^{(t)}$ of Alg. 1 can be transformed to a feasible user association vector $\boldsymbol{x}$ of problem **UA** (line 28).

Theorem 1: Alg. 1 converges to a stable allocation [8].

Proof: We prove this theorem by contradiction. Assume that Alg. 1 produces a matching μ with a blocking pair (u, j) by Definition 2. Since UE u prefers BS j over its current matched BS, i.e., $j \succ_u \mu(u)$, UE u must have proposed to BS j before its current match BS $\mu(u)$. In this case, the BS has rejected UE u due to a quota violation on j (lines 18-20). When UE u was rejected, then any less preferred UE u' was also rejected either before u (lines 13-17), or was made unable to propose because BS j is removed from UE u' preference list (lines 25-26). Thus, $u' \notin \mu(j)$, a contradiction.

Once the user association phase is over, the next step is to predict the contents popularity for the next time slot. Note that content popularity is required to make an efficient content placement decision. We will content popularity prediction and the cache placement scheme in the next subsection.

B. CONTENT'S POPULARITY PREDICTION BASED CACHING

The subproblem **CP** for cache placement at each BS j can be stated as follows:

$$CP: \quad \underset{\boldsymbol{y}}{\text{maximize:}}\ U_j(\boldsymbol{x}, \boldsymbol{y}) \tag{9a}$$

$$\text{subject to:} \sum_{f \in \mathcal{F}} y_{f,j} \leq c_j, \quad \forall j \in \mathcal{J}, \tag{9b}$$

$$y_{f,j} \in \{0, 1\}, \quad \forall f, u, j. \tag{9c}$$

The biggest challenge in solving the **CP** problem is that all contents are equally likely to be stored in the caching space. Thus, we need to predict the popularity of each content so that we can make an efficient cache placement decision such that the utility is maximized. Thus, in this section, we use a prediction scheme that can predict the content's popularity for the contents and assist each BS to make a cache placement decision, i.e., $\boldsymbol{y}$.

The overview of our prosed content's popularity prediction based caching process is shown in Fig. 2, where requests for each content are the input for the ARIMA model and the output is the collected future popularity score of each content. Then, those popularity scores are utilized by a cache decision algorithm to store the most popular contents among others. In this section, first, we introduce the content's popularity prediction design followed by the proposed cache placement design. Then, we discuss an overview of the ARIMA models and parameters selections to get the best suitable model for content popularity prediction. Next, we analyze

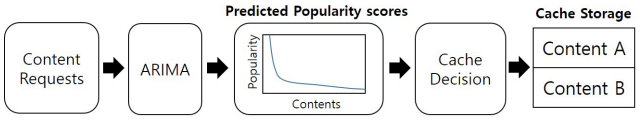


FIGURE 2. Learn, predict and cache.

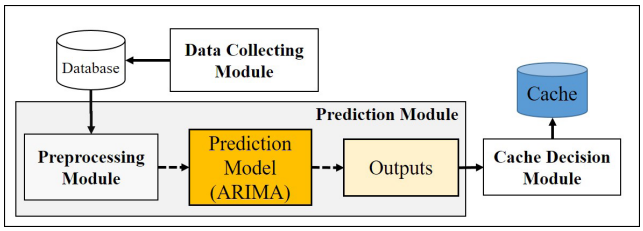


FIGURE 3. Popularity prediction and caching system design.

the ARIMA model that we choose. Finally, we integrate the ARIMA model based prediction process into cache decision process.

Fig. 3 shows the system design of the popularity prediction and caching system design, which includes: i) data collecting module, ii) database iii) preprocessing module, and iv) cache decision module. The data collecting module is responsible for collecting data such as receiving content requests at the base station, and this module keeps those data at the local database. The pre-processing module extracts data from database and feeds them into prediction module. Then, the prediction module produces the predicted future popularity score of each arriving content and feeds them into the cache decision module. Finally, the cache decision module chooses to cache contents based on future popularity scores.

1) DESCRIPTION OF ARIMA

ARIMA [39], [40] is one of the most common methods that is utilized in time series forecasting. Also, the ARIMA model can be fitted to time series data in order to predict future data points in the series. In ARIMA, there are three important parameters (p, d, q) used to determine ARIMA models. p is the auto-regressive part of the model to merge the effect of historical values into the ARIMA model. d is the integrated part of the model, which includes terms in the model that incorporate the amount of differencing (i.e., the number of historical points to subtract from the current value) to apply to the time series. q is the moving average part of the model and sets the error of the ARIMA model as a linear combination of the error values observed at previous data points in the past. ARIMA for non-seasonal usage can be denoted as ARIMA (p,d,q) and for seasonal usage can be denoted as ARIMA(P,D,Q)s. The term s is the periodicity of the time series (4 for quarterly periods, 12 for yearly periods, etc.). In the next section, we discuss the process of finding the optimal set of parameters of the ARIMA time series model for user demand prediction.

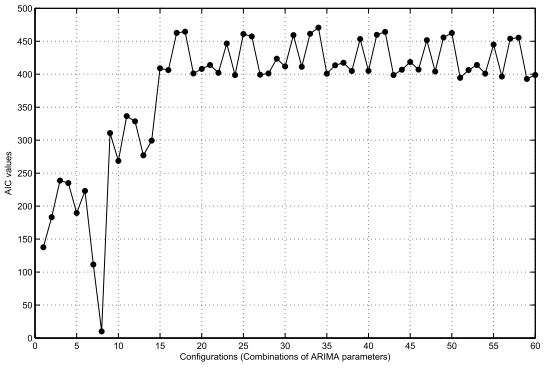


FIGURE 4. AIC values based on sets of parameter configurations.

2) PARAMETER SELECTION FOR THE ARIMA MODEL

The most challenging issue when applying the ARIMA model in any prediction problem is to choose the best parameters that optimize the Akaike Information Criterion (AIC) value [50]. Higher scores of AIC indicate that the model fits very well with the data points for the case of a large feature set. On the other hand, lower scores represent the same level of fitness for a small feature set. Therefore, we are interested in finding the model that yields the lowest AIC value. In this paper, we apply a grid search (hyper parameter optimization) to iteratively explore different combinations of parameters for model selection for the generated data. The generated data includes requests for a network of 20 users for 100 simulation runs, which is generated based on a Zipf distribution with parameter $\alpha = 1$. We ran tests using 60 combinations of ARIMA models and the results of all 60 combinations are shown in Fig. 4. Among the 60 combinations, we chose the 8th configuration (ARIMA (0,1,0)x(1,1,1,12)) because it has the minimum AIC value.

Algorithm 2 Cache Placement Decision Algorithm

1: **input**: List of $\{\widetilde{\pi}_{j,1}^{t+1}, \ldots, \widetilde{\pi}_{j,f}^{t+1}\}$;
2: At every time t, check the number of samples to predict;
3: **if** number of samples $\geqslant$ threshold **then**
4: Popularity score of each content f is predicted;
5: Construct the sorted content list based on predicted scores of contents;
6: Choose the most popular contents $\{1, 2, \ldots, f\}$ and cache contents with the condition (9b);
7: **else**
8: Choose the most popular contents $\{1, 2, \ldots, f\}$ from time $t - 1$ ensuring (9b);
9: **output**: $\boldsymbol{y}$, $\forall j$.

3) FITTING AN ARIMA TIME SERIES MODEL

Using grid search, we have identified the set of parameters (ARIMA (0,1,0)x(1,1,1,12)) that produces the best fitting model to our time series data. We then analyze the details of the ARIMA (0,1,0)x(1,1,1,12) model by feeding generated data. In Fig. 5, the Kernel density estimation (KDE) line

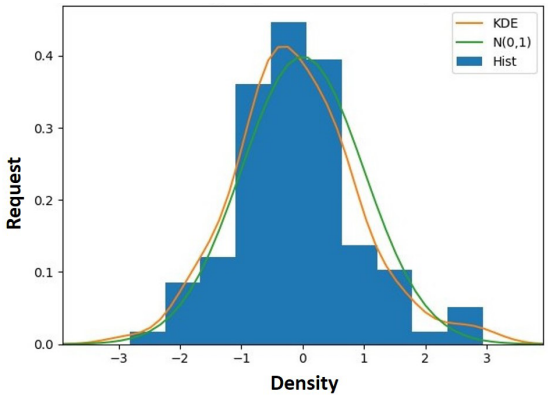


FIGURE 5. Histogram plus estimated density.

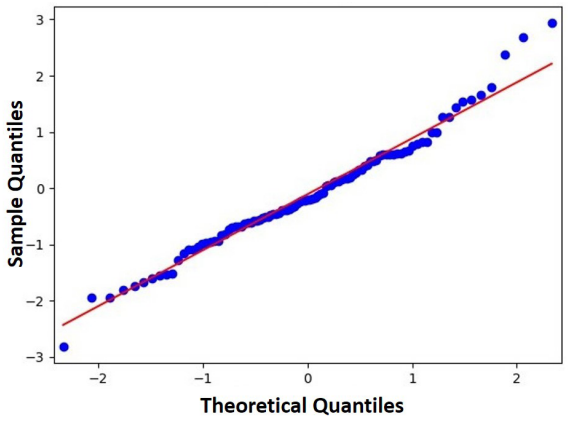


FIGURE 6. Normal quantiles.

follows closely with the normal distribution $N(0, 1)$ with mean 0 and standard deviation 1. Fig. 6 shows that the ordered distribution of residuals (blue points) follows the linear trend of the samples taken from the standard normal distribution $N(0, 1)$. Thus, this information indicates that the residuals are normally distributed. These observations led us to conclude that our model produces a satisfactory fit to forecast future values [40].

4) INTEGRATING THE ARIMA MODEL BASED PREDICTION SCHEME WITH THE CACHE DECISION PROCESS

Then, we integrate the ARIMA model based prediction with cache decision processes. Based on the ARIMA (0,1,0)x(1,1,1,12) seasonal model, we can obtain the future popularity score of each content. Then, the contents are sorted as a list based on predicted scores. Once the popularity scores are available at each BS, the cache placement algorithm utilizes the content's popularity list to choose the most popular contents, where the goal is to improve the cache hit at the base station.

Alg. 2 shows the proposed cache placement algorithm, which is run at the end of every time t at each BS j. The input of this algorithm is the list of predicted popularity scores of all requested contents at the base station. The output of this

algorithm is the decision to store the set of contents. At the initial time t, there is no predicted popularity score information. Thus, the BS j stores all content until the cache storage is full. At the end of time t, the BS receives the predicted scores from the prediction module and makes a cache decision depending on these predicted scores. In this case, the BS constructs the sorted list of the contents depending on the predicted scores. Then, it reduces the list based on the cache storage capacity and stores the contents with respect to constraint (9b).

C. JOINT USER ASSOCIATION AND CACHE PLACEMENT ALLOCATION

In this section, we discuss the overall Joint User Association and Cache Placement algorithm for our proposed problem (**P**), as shown in Alg. 3. We call it the *Cache Aware user association* (CA-UA) algorithm. In the initialization phase of the CA-UA algorithm, all BSs collect the users' channel state information (CSI) and broadcast the set of contents available at time slot $\tilde{t}$. Next, both algorithms, i.e., user association and cache placement algorithms are iteratively performed until we obtain a joint solution (i.e., a suboptimal solution). Fig. 7 shows the process diagram of the proposed joint algorithm. Note that, the joint algorithm converges when we receive the same user association for two consecutive time slots $\tilde{t}$. This indicates that the users cannot find a better BS compared to their currently associated BS in terms of good channel conditions and more usable cached contents. Formally, we state this as follows [49]:

Theorem 2: CA-UA Algorithm achieves a suboptimal solution of the original problem in (5).

Proof: This joint CA-UA algorithm is based on an alternative maximization approach. Since at each iteration ($\tilde{t}$), each subproblem (i.e., user association and cache placement) does not decrease the common objective function in a compact set, Algorithm 3 will finally converge to a sub-optimal solution of the original problem in (5).

Algorithm 3 Cache Aware User Association Algorithm (CA-UA)

1: BS obtains the CSI of all UEs in its coverage, $\forall j$;
2: Random placement of contents in each BS;
3: $\tilde{t} = 0, \boldsymbol{x}^{(\tilde{t})} = \phi, \boldsymbol{y}^{(\tilde{t})} =$ random;
4: **repeat**
5: $\quad \tilde{t} = \tilde{t} + 1$
6: $\quad \forall j$, Update the user association $\boldsymbol{x}^{(\tilde{t})}$ using Alg. 1;
7: $\quad$ Learn request pattern from time-slot, $\tilde{t}$;
8: $\quad$ Predict request pattern for time-slot, $\tilde{t} + 1$;
9: $\quad \forall j$, Update cache placement $\boldsymbol{y}^{(\tilde{t})}$ using Alg. 2;
10: **until** $\boldsymbol{x}^{(\tilde{t})} = \boldsymbol{x}^{(\tilde{t}-1)}$;

IV. NUMERICAL RESULTS

In our simulations, we consider a downlink transmission of a cellular system in which a single MBS is deployed at a fixed location, i.e., the center of the macro-cell with a radius of 500 m. Moreover, we randomly deploy five SBSs and U UEs following a homogeneous Poisson point process (PPP) under the macro-cell coverage. In our simulation, we assume a system bandwidth of 3 MHz, which is shared among all the BSs. The methodologies developed in this paper can also be applied to any value of system bandwidth. The motivation for our choice (i.e., 3 MHz) is to analyze the performance under a dense environment with peak network traffic and for the sake of simulation simplicity. Moreover, the bandwidth W of each channel and weight parameter ω are set to a normalized value of 1. Each UE u has a demand which follows a Zipf distribution. The main parameters used in our simulations are shown in Table 2 unless stated otherwise. Note that all statistical results stated, except for the real-time performance evaluation (i.e., Fig. 8), are averaged over a large number of independent runs of random locations of users, small cell base stations and resource block gains.

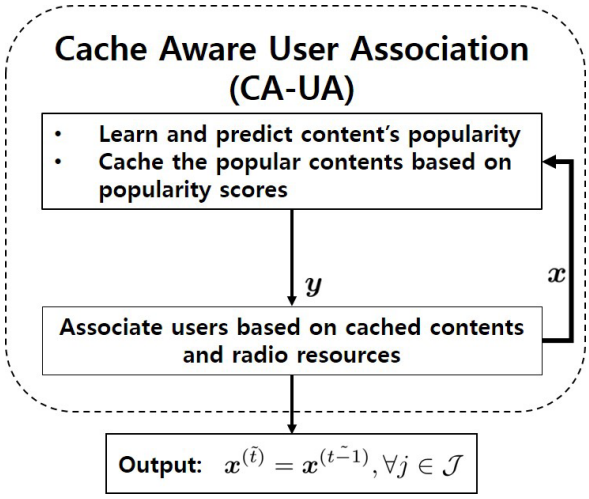


FIGURE 7. Overview process of the proposed scheme.

TABLE 2. Default simulation parameters [44].

Simulation Parameters	Values
Wireless Parameters	
Radius of MBS	500 m
Carrier frequency (f)	2 GHz
Frame Structure	Type 1 (FDD)
Transmission time interval (TTI)	1 ms
Total transmit power of MBS	46 dBm
Total transmit power of SBS	23 dBm
System bandwidth	3 MHz
Bandwidth of each RB (W)	180 kHz
Number of subcarriers per RB	12
Neighboring subcarrier spacing	15 kHz
Path loss (cellular link)	$128.1 + 37.6 \log(\mathrm{d})$, d[km]
Thermal noise for 1 Hz at 20 ˚C	-174 dBm
Caching Parameters	
Content size	1KB
No. of Content Server	1
Zipf exponent	1
Total Contents	10^3
Total Cache size	15% of Total Contents

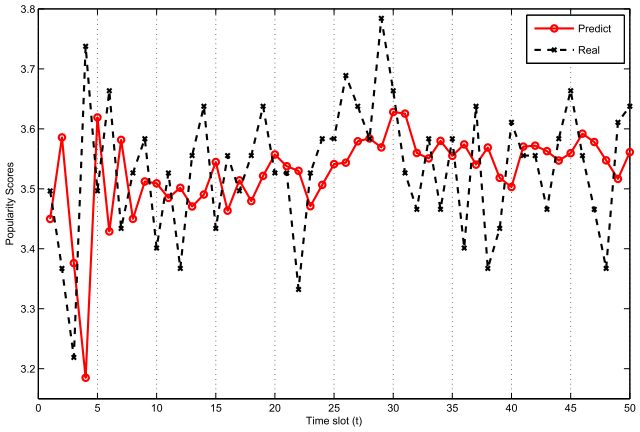


FIGURE 8. Comparison between predicted results and actual results.

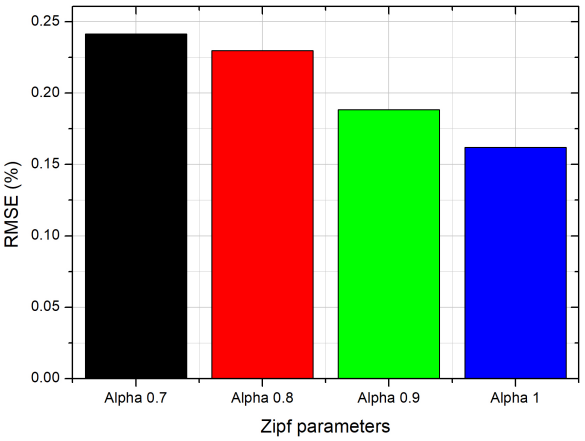


FIGURE 9. Comparison of the prediction error for different popularity profiles.

A. NUMERICAL RESULTS FOR LEARNING

In this paper, we used a homogeneous cache size for each BS in the network. Then, we considered how much cache size should be allocated for the whole autonomous system. We assigned 15% of the total contents as a cache size for all BSs $\mathcal{J}$. To generate user requests, we first considered content popularity. For the simulation, we assumed the content popularity follows a Zipf distribution, where the probability of choosing content f is given by

$$P(\alpha; s, F) = \frac{1/i^{\alpha}}{\sum_{f=1}^{F} 1/f^{\alpha}}, \tag{10}$$

where F is the number of contents, i is the rank of content f and α is the value of the exponent characterizing the distribution. To run the simulation, we considered the popularity to follow a Zipf distribution range ($\alpha = 1$).

In order to evaluate the performance of the proposed prediction scheme, we first show the comparison of one-step ahead predicted results with actual data point, in order to evaluate how much the proposed scheme deviates from the actual data points. We used the mean squared error to evaluate the deviation between the predicted and actual data points. Fig. 8 shows the comparison of prediction results (red line) and the actual data points (black line), where the actual data points are generated by utilizing a Zipf distribution with parameter α at a value of 1 for 20 users for 100 random simulation runs. It was observed that the mean squared error of the proposed scheme is close to 0.16, where we fed 30 data points as historical data and then started collecting the predicted results from the next time slot, i.e., 31 to 100. However, for clear illustration, we reduced the scale of fig. 8 from 31 to 80.

Then, we tested the prediction scheme with different content popularity profiles, where we evaluated the performance of the learning scheme under different Zipf distribution parameters, i.e., an α value from 1 to 0.9, 0.8, and 0.7. For these three popularity profiles, we also generated 100 data points for 20 users following the same process. The results shown in Fig. 9 reveal that our proposed popularity prediction scheme can make accurate predictions even when the popularity profile is changing. However, the RMSE increases as α decreases.

B. NUMERICAL RESULTS FOR THE CACHE AWARE USER ASSOCIATION ALLOCATION

In order to evaluate the performance of the Cache Aware User Association (CA-UA) scheme, first, we show the comparison in terms of normalized utility achieved by enabling the proposed CA-UA scheme under different network sizes (i.e., the number of users, U). We investigated the normalized utility under two scenarios by varying the number of resources in the system, i.e., 1.4 MHz (6 resource blocks) and 3 MHz (15 resource blocks). Second, we evaluated the average time slots required for the CA-UA scheme by varying the network size under the two different settings of system bandwidth. Third, we also investigated the cache hit ratio under the aforementioned settings. Then, we determined the cache miss percentage and the reduction of backhaul load in the network by using the proposed scheme. For comparison purposes, we compare our proposed approach with a cache unaware user association approach (CUA-UA). This approach aims to associate UEs based on the standard association approach, i.e., received signal strength indicator. Finally, we compare our solution with the optimal solution. We have calculated the optimal solution via the exhaustive search method. Note that the exhaustive search method can only be applied in a centralized manner in which all network information are assumed to be known and available at the centralized controller such as user demands for contents, user channel characteristics, cache size and etc. Furthermore, due to the combinatorial nature of our problem, we cannot apply the exhaustive search method for a large scale network. Therefore, we have taken a small scale network to compare our proposed approach (i.e., Cache Aware user association) with the optimal solution. The new network settings include a maximum of 30 users (i.e., network size) and a system bandwidth of 1.4 MHz at each BSs. We assume that the total contents in this new network setting is 100 whereas the total cache size is 5% of the total content at each BSs.

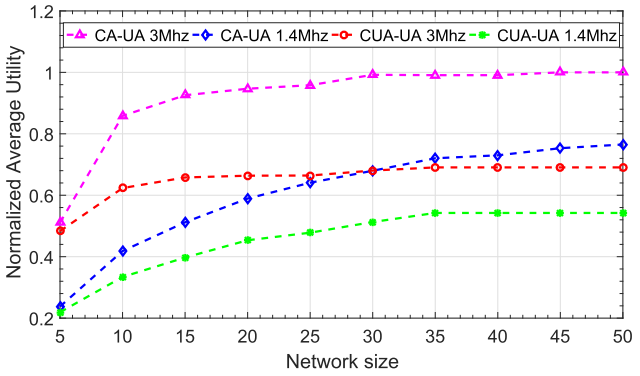


FIGURE 10. Normalized utility vs. network size for the CA-UA and CUA-UA schemes.

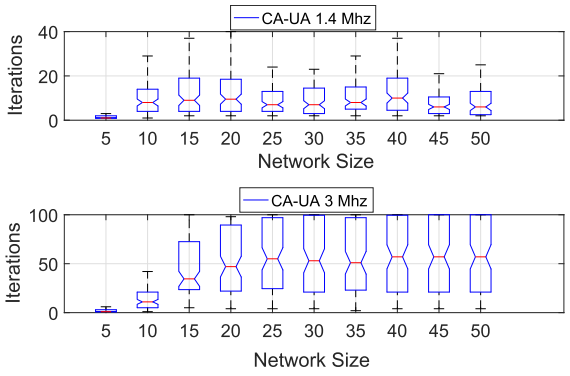


FIGURE 11. Average number of iterations vs. network size.

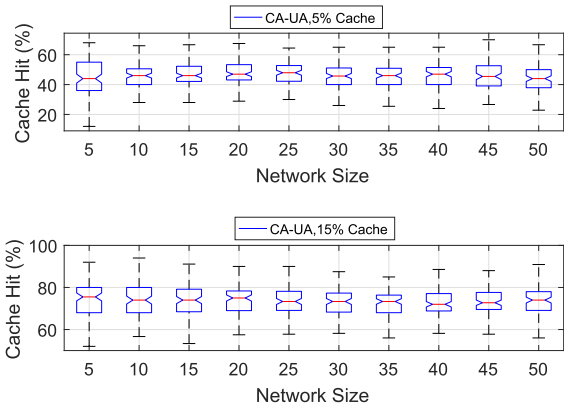


FIGURE 12. Cache hit vs. network size for different cache sizes.

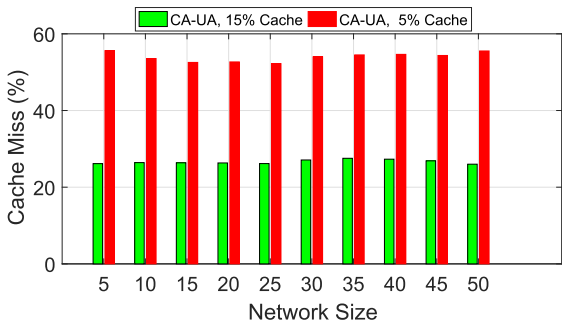


FIGURE 13. Cache miss of the CA-UA schemes for different cache sizes.

In Fig. 10, the achieved normalized utility is shown both for cache aware and cache unaware user association under two different bandwidth settings. For, this simulation, we increase the network size to evaluate the utility. First, it can be observed that as the network size becomes sufficiently large, the utility saturates for all schemes. This is because of the limited bandwidth that is occupied as the network grows and no new users can be accommodated in the network. Second, we observe that the utility of CA-UA for 3 Mhz saturates for a network size of 30 users, whereas the utility of CA-UA for 1.4 Mhz does not saturate at that point. The main reason is that we have the same cache size for both the scenarios (i.e., 5 % of the total contents). In the CA-UA 3 Mhz scenario, more users are accommodated compared to the CA-UA 1.4 Mhz case with different channel conditions. Thus, the BS considers the requests of a larger number of users when making a caching decision whereas in the other case, a smaller number of users with good channel condition are considered. However, the CA-UA 1.4 Mhz scheme still achieved 77% of normalized utility of the CA-UA 3 Mhz scheme. Finally, the CA-UA 1.4 Mhz and CA-UA 3 Mhz schemes observed a significant gain of up to 28% and 31% in terms of the normalized utility when compared with CUA-UA 1.4 Mhz and CUA-UA 3 Mhz schemes, respectively.

Fig. 11 shows the box plot for the average iterations required for the proposed CA-UA scheme to converge for two different resource settings. The box plot is generated by using 100 simulation runs with random SBS and UE locations. We can see that the convergence time (median value of the box plot) of our approach is reasonable, i.e., 15 and 50 iterations (TTI) for 1.4 and 3Mhz respectively. Moreover, it can be seen that the number of iterations for the CA-UA 3 Mhz scheme is significantly higher when compared to the CA-UA 1.4 Mhz scheme. The main reason for higher convergence time comes from the fact that more UEs are accommodated in the network, thus, the number of possible configurations for learning increases, i.e., the cache placement algorithm. Therefore, a larger number of iterations is observed. However, it can be seen that when the network size is large enough (i.e., 30 and more) the median of iterations is almost indistinguishable.

In Fig. 12, we investigate the cache hit ratio for different cache sizes. We use the box plot to analyze the cache hit in terms of the percentage. In this simulation, we fixed the system bandwidth to 3 Mhz. It was observed that the change in network size does not affect the cache hit ratio. However, when the cache size changes, a significant change in cache hit is observed. We observe upto 47% increase in the median value of the cache hit box plot when the cache size was increased from 5% to 15% for all network sizes. Fig. 13 also follows the same trend and shows the cache miss percentage of both scenarios.

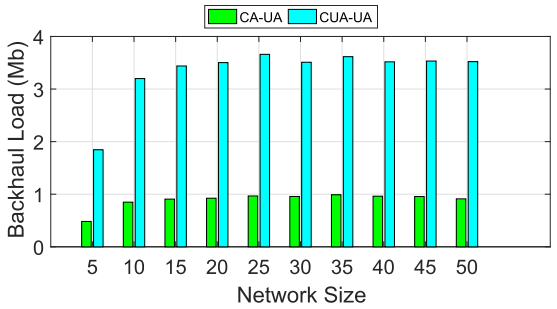


FIGURE 14. Backhaul load for CA-UA and CUA-UA schemes.

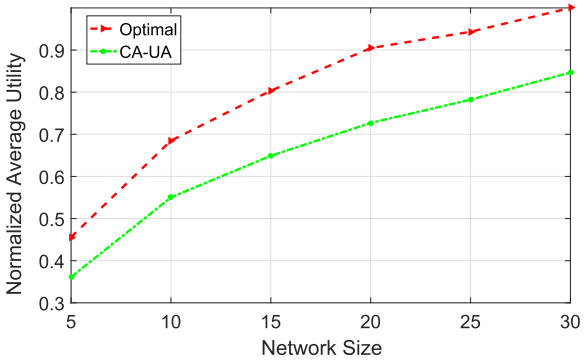


FIGURE 15. Normalized utility vs. network size for the CA-UA and the Optimal scheme.

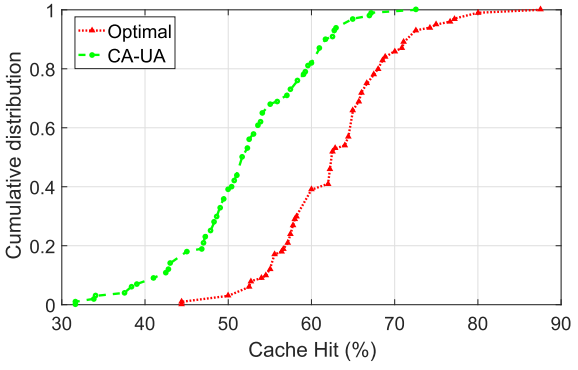


FIGURE 16. Cache hit for the CA-UA and the Optimal scheme.

Fig. 14 evaluates the backhaul load both for the CA-UA and CUA-UA schemes. In this simulation, we use the 15% cache size setting and 3 Mhz system bandwidth. It can be inferred that the CA-UA scheme (up to 1 Mb for 15 and larger network sizes) significantly reduces the backhaul load when compared to the CUA-UA scheme (up to 3.3 Mb for 15 and higher network sizes) for all network sizes. It saves up to twice the backhaul bandwidth for large network sizes.

In Fig. 15, the achieved normalized utility is shown both for optimal solution and CA-UA schemes. For, this simulation, we increase the network size to evaluate the normalized utility. It can be inferred that the proposed CA-UA scheme achieves up to 82% of the utility obtained by the optimal solution for any network size.

A similar trend is also observed in Fig. 16 in which the cumulative distribution of the cache hit is compared. In Fig. 16, we compare the cache hit observed by the optimal solution and the proposed CA-UA solution. For this simulation, we fix the network size to 30 UEs and observe the cache hit. We see that the average cache hit achieved by the CA-UA scheme is up to 83% of the optimal solution. Thus, we can state that our proposed CA-UA approach is close to the optimal solution.

V. CONCLUSION

In this work, we design a novel cache aware user association scheme for heterogeneous cellular networks. We have considered two important aspects in our design; user association and cache placement. We applied the concepts of matching theory for addressing the user association aspect and then used an autoregressive integrated moving average scheme for learning and predicting the content popularity. The results of the prediction scheme were then used in the content placement decision. The proposed cache aware user association scheme has been shown to achieve a stable, distributed, scalable and suboptimal solution for the network. Simulation results reveal that the proposed scheme significantly outperforms the cache unaware scheme in terms of the network utility and backhaul load. Moreover, we also have shown the convergence and cache hit ratio of the proposed scheme under different scenarios. As future work, we intend to enhance the proposed approach to guarantee the quality of service for the users.

REFERENCES

[1] K. Hamidouche, W. Saad, M. Debbah, J. B. Song, and C. S. Hong, "The 5G cellular backhaul management dilemma: To cache or to serve," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 4866–4879, Aug. 2017.

[2] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: Exploiting content caching and delivery techniques for 5G systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 131–139, Feb. 2014.

[3] X. Li, X. Wang, K. Li, and V. C. M. Leung, "CaaS: Caching as a service for 5G networks," *IEEE Access*, vol. 5, pp. 5982–5993, May 2017.

[4] M. J. Piran, S. M. R. Islam, and D. Y. Suh, "CASH: Content- and network-context-aware streaming over 5G HetNets," *IEEE Access*, vol. 6, pp. 46167–46178, 2018.

[5] C. S. Hong, S. M. A. Kazmi, S. Moon, and N. Van Mui, "SDN based wireless heterogeneous network management," in *Recent Advances in Electrical Engineering and Related Sciences—AETA*, 2016, pp. 3–12.

[6] S. Moon, T. LeAnh, S. M. A. Kazmi, T. Z. Oo, and C. S. Hong, "SDN based optimal user association and resource allocation in heterogeneous cognitive networks," in *Proc. 17th Asia-Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Aug. 2015, pp. 580–583.

[7] T. M. Ho *et al.*, "Network economics approach to data offloading and resource partitioning in two-tier LTE HetNets," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, Ottawa, ON, Canada, May 2015, pp. 914–917.

[8] S. M. A. Kazmi, N. H. Tran, W. Saad, L. B. Le, T. M. Ho, and C. S. Hong, "Optimized resource management in heterogeneous wireless networks," *IEEE Commun. Lett.*, vol. 20, no. 7, pp. 1397–1400, Jul. 2016.

[9] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," in *Proc. 2nd ICN Workshop Inf.-Centric Netw.*, Aug. 2012, pp. 55–60.

[10] A. Dabirmoghaddam, M. M. Barijough, and J. J. Garcia-Luna-Aceves, "Understanding optimal caching and opportunistic caching at the edge of information-centric networks," in *Proc. 1st ACM Conf. Inf.-Centric Netw.*, 2014, pp. 47–56.

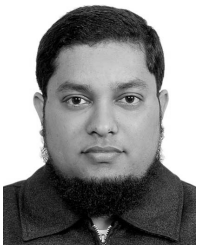
[11] D. Liu, B. Chen, C. Yang, and A. F. Molisch, "Caching at the wireless edge: Design aspects, challenges, and future directions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 22–28, Sep. 2016.

[12] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 82–89, Aug. 2014.
[13] J. Zhang, X. Zhang, Z. Yan, Y. Li, W. Wang, and Y. Zhang, "Social-aware cache information processing for 5G ultra-dense networks," in *Proc. 8th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Yangzhou, China, Oct. 2016, pp. 1–5.
[14] Y. Wang, X. Tao, X. Zhang, and G. Mao, "Joint caching placement and user association for minimizing user download delay," *IEEE Access*, vol. 4, pp. 8625–8633, 2016.
[15] S. He, H. Tian, X. Lyu, G. Nie, and S. Fan, "Distributed cache placement and user association in multicast-aided heterogeneous networks," *IEEE Access*, vol. 5, pp. 25365–25376, 2017.
[16] F. Pantisano, M. Bennis, W. Saad, and M. Debbah, "Match to cache: Joint user association and backhaul allocation in cache-aware small cell networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, London, U.K., Jun. 2015, pp. 3082–3087.
[17] H. Chen, Q. Chen, R. Chai, and D. Zhao, "Utility function optimization based joint user association and content placement in heterogeneous networks," in *Proc. 9th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Nanjing, China, Oct. 2017, pp. 1–6.
[18] G. Ren, H. Qu, J. Zhao, S. Zhao, and Z. Luan, "A distributed user association and resource allocation method in cache-enabled small cell networks," *China Commun.*, vol. 14, no. 10, pp. 95–107, Oct. 2017.
[19] M. Dehghan *et al.*, "On the complexity of optimal request routing and content caching in heterogeneous cache networks," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1635–1648, Jun. 2017.
[20] K. Thar, S. Ullah, and C. S. Hong, "Consistent hashing based cooperative caching and forwarding in content centric network," in *Proc. 16th Asia-Pacific Netw. Oper. Manage. Symp.*, Sep. 2014, pp. 1–4.
[21] K. Thar, T. Z. Oo, C. Pham, S. Ullah, D. H. Lee, and C. S. Hong, "Efficient forwarding and popularity based caching for content centric network," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2015, pp. 330–335.
[22] K. Thar, S. Ullah, R. Haw, T. LeAnh, T. Z. Oo, and C. S. Hong, "Hybrid caching and requests forwarding in information centric networking," in *Proc. 17th Asia-Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Aug. 2015, pp. 203–208.
[23] S. Li, J. Xu, M. van der Schaar, and W. Li, "Trend-aware video caching through online learning," *IEEE Trans. Multimedia*, vol. 18, no. 12, pp. 2503–2516, Dec. 2016.
[24] E. Zeydan *et al.*, "Big data caching for networking: Moving from cloud to edge," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 36–42, Sep. 2016.
[25] M. Chen, W. Saad, C. Yin, and M. Debbah, "Echo state networks for proactive caching in cloud-based radio access networks with mobile users," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3520–3535, Jun. 2017.
[26] J. Famaey, T. Wauters, and F. De Turck, "On the merits of popularity prediction in multimedia content caching," in *Proc. 12th IFIP/IEEE Int. Symp. Integr. Netw. Manage. Workshops (IM)*, May 2011, pp. 17–24.
[27] A. O. Nwana, S. Avestimehr, and T. Chen, "A latent social approach to youtube popularity prediction," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2013, pp. 3138–3144.
[28] S. He, H. Tian, and X. Lyu, "Edge popularity prediction based on social-driven propagation dynamics," *IEEE Commun. Lett.*, vol. 21, no. 5, pp. 1027–1030, May 2017.
[29] S. Ouyang, C. Li, and X. Li, "A peek into the future: Predicting the popularity of online videos," *IEEE Access*, vol. 4, pp. 3026–3033, 2016.
[30] C. Li, J. Liu, and S. Ouyang, "Characterizing and predicting the popularity of online videos," *IEEE Access*, vol. 4, pp. 1630–1641, 2016.
[31] R. Devooght and H. Bersini. (2016). "Collaborative filtering with recurrent neural networks." [Online]. Available: https://arxiv.org/abs/1608.07400
[32] F. Figueiredo, "On the prediction of popularity of trends and hits for user generated videos," in *Proc. 6th ACM Int. Conf. Web Search Data Mining*, 2013, pp. 741–746.
[33] G. Szabo and B. A. Huberman, "Predicting the popularity of online content," *Commun. ACM*, vol. 53, no. 8, pp. 80–88, 2010.
[34] S. Li, J. Xu, M. van der Schaar, and W. Li, "Popularity-driven content caching," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2016, pp. 1–9.
[35] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008, pp. 405–416.
[36] R. Kleinberg, A. Slivkins, and E. Upfal, "Multi-armed bandits in metric spaces," in *Proc. 14th Annu. ACM Symp. Theory Comput.*, 2008, pp. 681–690.
[37] J. Famaey, F. Iterbeke, T. Wauters, and F. De Turck, "Towards a predictive cache replacement strategy for multimedia content," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 219–227, Jan. 2013.
[38] P. Blasco and D. Gündüz, "Learning-based optimization of cache content in a small cell base station," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2014, pp. 1897–1903.
[39] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*. San Francisco, CA, USA: Holden-Day, 1970.
[40] W. A. Woodward, H. L. Gray, and A. C. Elliott, *Applied Time Series Analysis With R*. Boca Raton, FL, USA: CRC Press, 2016.
[41] S. M. A. Kazmi *et al.*, "Resource management in dense heterogeneous networks," in *Proc. 17th Asia-Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Busan, South Korea, Aug., 2015, pp. 440–443.
[42] S. M. A. Kazmi, N. H. Tran, T. M. Ho, D. K. Lee, and C. S. Hong, "Decentralized spectrum allocation in D2D underlying cellular networks," in *Proc. 18th Asia-Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Kanazawa, Japan, Aug. 2016, pp. 1–6.
[43] C. Tekin and M. V. D. Schaar, "Contextual online learning for multimedia content aggregation," *IEEE Trans. Multimedia*, vol. 17, no. 4, pp. 549–561, Apr. 2015.
[44] S. M. A. Kazmi *et al.*, "Mode selection and resource allocation in device-to-device communications: A matching game approach," *IEEE Trans. Mobile Comput.*, vol. 16, no. 11, pp. 3126–3141, Nov. 2017.
[45] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[46] S. M. A. Kazmi, N. H. Tran, T. M. Ho, and C. S. Hong, "Hierarchical matching game for service selection and resource purchasing in wireless network virtualization," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 121–124, Jan. 2018.
[47] S. Wang, J. Bi, and J. Wu, "Collaborative caching based on hash-routing for information-centric networking," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 535–536, 2013.
[48] S. Podlipnig and L. Böszörmenyi, "A survey of Web cache replacement strategies," *ACM Comput. Surv.*, vol. 35, no. 4, pp. 374–398, 2003.
[49] S. M. A. Kazmi, N. H. Tran, T. M. Ho, A. Manzoor, D. Niyato, and C. S. Hong, "Coordinated device-to-device communication with non-orthogonal multiple access in future wireless cellular networks," *IEEE Access*, vol. 6, pp. 39860–39875, 2018.
[50] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *Selected Papers of Hirotugu Akaike*. New York, NY, USA: Springer, 1998, pp. 199–213.

RIM HAW received the B.S. and M.S. degrees in computer engineering from Kyung Hee University, Seoul, South Korea, in 2008 and 2010, respectively. He is currently pursuing the Ph.D. degree in computer science and engineering with Kyung Hee University. He holds several national and international patents. His research interests include ambient intelligent living, advanced wireless network protocols, and P2P networks. He is a member of KIISE.

S. M. AHSAN KAZMI received the master's degree in communication system engineering from the National University of Sciences and Technology, Pakistan, in 2012, and the Ph.D. degree in computer science and engineering from Kyung Hee University, South Korea, in 2017. Since 2018, he has been with the Network, Cyber, and Information Security Lab, Secure System and Network Engineering, Innopolis University, Russia, where he is currently an Assistant Professor. His research interests include applying analytical techniques of optimization and game theory to radio resource management for future cellular networks. He received the Best KHU Thesis Award in engineering, in 2017, and several best paper awards from prestigious conferences.

KYI THAR received the bachelor's degree in computer technology from the University of Computer Studies, Yangon, Myanmar, in 2007. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Kyung Hee University, South Korea, for which he was awarded a scholarship for his graduate study, in 2012. His research interests include name-based routing, in-network caching, multimedia communication, scalable video streaming, wireless network virtualization, deep learning, and Future Internet.



MD GOLAM RABIUL ALAM received the B.S. and M.S. degrees in computer science and engineering, and information technology, and the Ph.D. degree in computer engineering from Kyung Hee University, South Korea, in 2017. He has served as a Post-Doctoral Researcher at the Computer Science and Engineering Department, Kyung Hee University, from 2017 to 2018. He is currently an Assistant Professor with the Computer Science and Engineering Department, BRAC University, Bangladesh. His research interests include healthcare informatics, mobile cloud and Edge computing, ambient intelligence, and persuasive technology. He is a member of the IES, CES, CS, SPS, CIS, ComSoc, and KIISE. He has received several best paper awards from prestigious conferences.

CHOONG SEON HONG (S'95–M'97–SM'11) received the B.S. and M.S. degrees in electronic engineering from Kyung Hee University, Seoul, South Korea, in 1983 and 1985, respectively, and the Ph.D. degree from Keio University, Minato, Japan, in 1997. In 1988, he joined Korea Telecom, where he worked on broadband networks as a member of the Technical Staff. In 1993, he joined Keio University. He worked for the Telecommunications Network Laboratory, Korea Telecom, as a Senior Member of Technical Staff and as the Director of the Networking Research Team until 1999. Since 1999, he has been a Professor with the Department of Computer Science and Engineering, Kyung Hee University. His research interests include future Internet, ad hoc networks, network management, and network security. He is a member of ACM, IEICE, IPSJ, KIISE, KICS, KIPS, and OSIA. He has served as the General Chair, the TPC Chair/Member, or as an Organizing Committee Member for international conferences such as NOMS, IM, APNOMS, E2EMON, CCNC, ADSN, ICPP, DIM, WISA, BcN, TINA, SAINT, and ICOIN. In addition, he is currently an Associate Editor of the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, the *International Journal of Network Management*, and the *Journal of Communications and Networks*, and an Associate Technical Editor of the *IEEE Communications Magazine*.