IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 January 2023, accepted 11 February 2023, date of publication 15 February 2023, date of current version 23 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3245803

RESEARCH ARTICLE

Content Caching Based on Popularity and Priority of Content Using seq2seq LSTM in ICN

MIN WOOK KANG[1] AND YUN WON CHUNG[2], (Member, IEEE)

[1]Department of Information and Telecommunication Engineering, Soongsil University, Seoul 06978, South Korea
[2]School of Electronic Engineering, Soongsil University, Seoul 06978, South Korea

Corresponding author: Yun Won Chung (ywchung@ssu.ac.kr)

This work was supported in part by the Ministry of Science and ICT (MSIT), South Korea, under the Information Technology Research Center (ITRC) Support Program supervised by the Institute for Information and Communications Technology Planning and Evaluation (IITP), under Grant IITP-2022-2017-0-01633; and in part by IITP Grant through the Korea Government, MSIT, Development of Candidate Element Technology for Intelligent 6G Mobile Core Network, under Grant 2022-0-01015.

ABSTRACT Efficient content caching is essential to address the explosive growth of multimedia contents and most works on content cache placement have been proposed mainly based on the popularity of content. Since the priority of content is also an important attribute of content, we consider both popularity and priority of content together for content caching in information-centric networking (ICN). We define weighted delivery cost of content as content delivery cost multiplied by a weighted sum of popularity and priority of the content. Then, we formulate optimized content cache placement problem to minimize weighted content delivery cost for all content requests in a hierarchical network architecture with multi-access edge computing (MEC) and software-defined networking (SDN) controller. Average quality of experience (QoE), i.e., average content delivery cost, for contents with each priority is imposed as constraint. The number of content requests is predicted based on seq2seq long short-term memory (LSTM) model in MECs and this is delivered to SDN controller. Then, SDN controller obtains predicted popularity of contents and decides content placement in MECs and core routers by solving the content cache placement optimization problem based on binary particle swarm optimization (BPSO). Performance of the proposed content caching scheme is compared with conventional popularity-based, popularity prediction-based, and popularity prediction-based optimization schemes, from the aspect of QoE satisfaction ratio, average cost, weighted average cost, total cost, and weighted total cost. Numerical results show the effectiveness of the proposed scheme at caching content with high priority efficiently, at the expense of caching content with low priority.

INDEX TERMS Caching, information-centric networking, seq2seq LSTM, BPSO, popularity, priority.

I. INTRODUCTION

Content caching is one of the most important topics nowadays due to the explosive growth of multimedia content. In content caching, contents from origin servers are cached in distributed content cache servers, and requested contents from users can be retrieved from appropriate nearby servers, instead of origin servers. Therefore, traffic overhead at origin servers is alleviated and content retrieval latency is reduced, too [1], [2].

In mobile communication networks, such as 5G, multi-access edge computing (MEC) can be used efficiently for content caching [3], [4], [5]. Since MECs are located closely with mobile users, contents can be returned to the requesting mobile users directly, if they have the requested contents, and content retrieval latency can be significantly reduced, compared to the case where contents are returned from either distant origin servers or core routers. Since the resource of MEC, i.e., buffer memory, is limited, however, it is important to select appropriate contents to cache in MECs.

Regarding content caching in MECs, popularity of content is mainly used to select contents to cache in MECs [6], [7], [8], [9], [10], [11], [12], [13]. This is because if any content is more popular, it will be requested more often than others, and thus it should be cached in MECs more actively

The associate editor coordinating the review of this manuscript and approving it for publication was Qingchun Chen.

than less popular content. Recently, the popularity of content is predicted by using deep neural network, such as long short-term memory (LSTM) [14] and the predicted popularity of content is used to decide contents to cache in MECs or edge servers appropriately [9], [10], [11], [12], [13], [15], [16], [17].

Software-defined networking (SDN) is also used for efficient content caching [7], [8]. In SDN, switch or router sends state information to SDN controller. Then, SDN controller can have global view of networks using this information and controls the operation of switch or router efficiently. In aforementioned works [7], [8], SDN controller obtains content information, such as cached content and predicted popularity, from network nodes. Then it uses this information to decide content cache placement in network nodes.

Efficient content caching and retrieval in information-centric networking (ICN) has been proposed, too [18], [19]. In ICN, content name is used as the identifier of content, instead of IP address [18], [19]. A consumer, i.e., a content requester, sends an Interest message, which has the name of the requested content, to its nearby ICN routers. The interface of the received Interest is stored in pending interest table (PIT) of the ICN routers. The Interest message is then propagated to appropriate neighbor ICN routers by using the information in forwarding information base (FIB), which is similar to forwarding table in Internet. Finally, if an ICN router which received the Interest message has the requested content in its content store (CS), Data, i.e., requested content, is returned to the requested user through the reverse path of forwarded Interest by using the stored interface information of PIT. In ICN, caching is an important issue since the operation of ICN is basically based on content caching, and leave copy everywhere (LCE) [20] and leave copy down (LCD) [21] are basic caching schemes in ICN.

In most of the works mentioned above [6], [7], [8], [9], [10], [11], [12], [13], [15], [16], [17], however, content popularity is mainly used to decide content cache placement. Since priority of content is an important attribute of content in networks, such as ICN [22], [23], content should be treated appropriately, based on the priority of content, in order to satisfy the quality of experience (QoE) of users. For example, if any content is an urgent content and should be delivered faster than other contents, it is necessary to cache such high priority contents more actively at routers which are closely located with content requesters.

In [24], [25], [26], and [27], which have been carried out by the same authors, service class of content was considered also, in addition to the popularity of content for content caching. The proportion of buffer memory is allocated to each service class statically, and contents of each service class are cached in each allocated buffer memory based on the popularity of the contents, in named data network [24] and in mobile named data network [25], respectively. In [26], the proportion of buffer memory is allocated to each service class dynamically, based on the request pattern of contents. In [27], the proportion of buffer memory is allocated to each service class by using the formula based on the weight of service class and the number of total content requests which is higher than a threshold for each service class.

In [24], [25], [26], and [27], however, although service class is considered for content caching, it is only used for the allocation of buffer memory for each service class and the popularity of content is still used for content cache placement.

In this paper, we consider both content popularity and content priority together for content cache placement in ICN, where content popularity is predicted by using seq2seq LSTM model. Then, we propose an optimal cache placement scheme by formulating optimal cache placement problem to minimize the cost of content delivery, while guaranteeing the average QoE for the content with priority, based on the predicted content popularity as well as content priority. The formulated optimization problem is solved by SDN controller by using binary particle swarm optimization (BPSO) due to the NP hard property of the formulated optimization problem.

The main novelties and contributions of our paper are summarized as follows:

- We investigate the problem of content cache placement at network nodes, such as MEC and core routers, with limited cache memory by formulating an optimization problem to place content cache appropriately, while minimizing the weighted delivery cost of content, defined as content delivery cost multiplied by a weighted sum of popularity and priority of the content. Average QoE of content with each priority, which is defined as the average delivery cost of the content with each priority, is imposed as constraint in the formulated optimization problem. The number of content requests is predicted at each MEC firstly, based on seq2seq LSTM model and this is delivered to SDN controller. Then, SDN controller obtains predicted popularity of contents, by normalizing the popularity of content at each MEC into the range [0,1] by min-max normalization. Content cache placement decision is carried out in SDN controller, based on the predicted popularity and pre-defined priority of each content by solving the optimization problem with average QoE constraint by using BPSO.
- Performance evaluation has been conducted based on synthesized data set, from the aspect of QoE satisfaction ratio, average cost, weighted average cost, total cost, and weighted total cost, considering contents with low, medium, and high priority. Simulation results show the effectiveness of the proposed scheme at caching content with high priority efficiently, at the expense of caching content with low priority.

The rest of this paper is organized as follows: Section II presents the proposed optimal content caching scheme for ICN, where the proposed cache placement procedure, content request model, content popularity prediction, content caching optimization, and algorithm of the proposed content cache placement optimization scheme are described in detail.

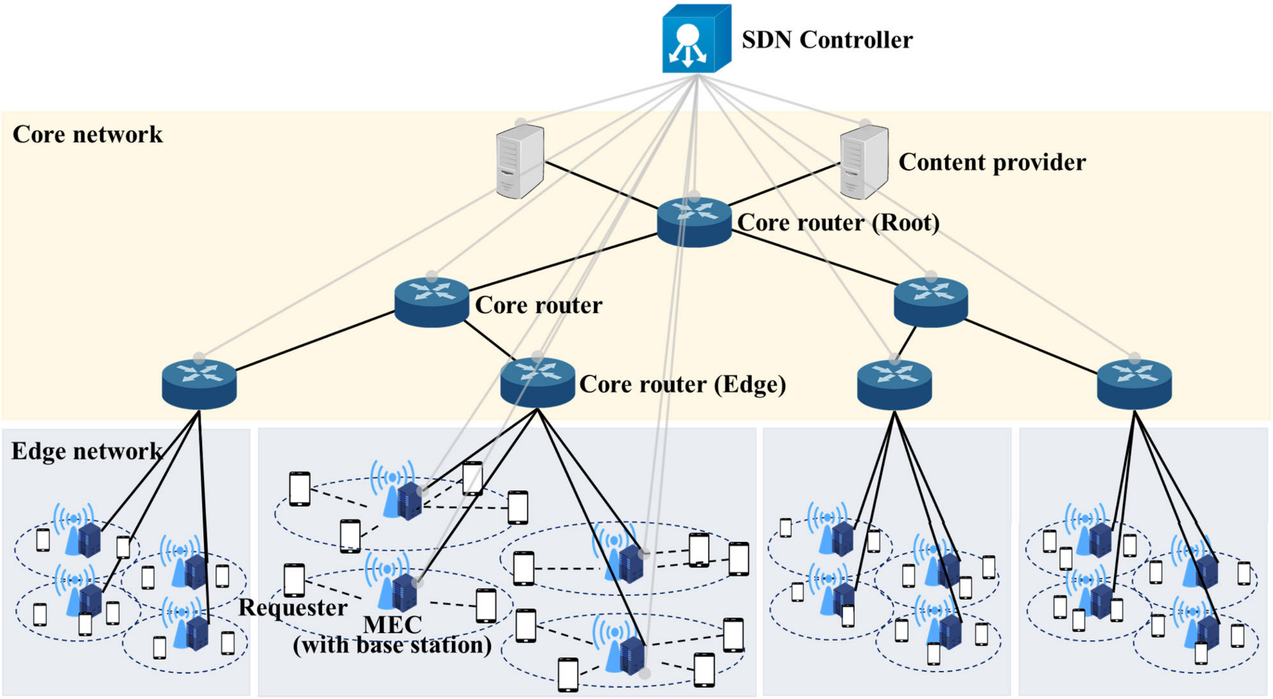


FIGURE 1. A considered hierarchical network architecture.

Section III presents numerical examples obtained by solving the optimization problem based on BPSO. Finally, Section IV summarizes this work.

II. PROPOSED OPTIMAL CONTENT CACHING SCHEME FOR ICN

A. PROPOSED CONTENT CACHING PROCEDURE

In this paper, we consider a hierarchical network architecture, as shown in Fig. 1, which consists of edge network and core network basically, similar to our previous works [28], [29]. In edge network, MECs are co-located with base stations (BSs) and they cache contents in their buffer memory. In core network, core routers are connected hierarchically in a tree topology and they also cache contents. Especially, core routers which are connected to MECs directly are called as edge routers, and core router which is located at the top of the hierarchy is called as root router. Content providers are origin servers for their generated contents. SDN controller obtains predicted content popularity from MECs and controls cache placement in MECs and core routers in a centralized manner.

Since the buffer memory of MECs and core routers is limited, it is necessary to place contents to either MECs or core routers appropriately, and this is called as a cache placement problem. Similar to previous works on cache placement [6], [7], [8], [9], [10], [11], [12], [13], [15], [16], [17], we also consider popularity of content to decide cache placement basically. That is, more popular contents should be cached more actively than less popular contents. In addition to popularity of content, however, we also consider priority of content, too. As reported in [22] and [23], priority is one of the important attributes of ICN content and thus, contents with higher priority should be cached more actively than those with lower priority. Therefore, we consider both popularity and priority together to decide cache placement.

We note that in current ICN standards, such as named data networking (NDN) [30] and content-centric networks (CCNx) [31], the priority of content is not defined in the packet format of ICN messages yet, to the best of our knowledge, although the concept of accommodating traffic specifier in the packet format of Interest message in NDN or CCNx for QoS has been proposed [32]. In ICN, the name of content can be used for the identification of content priority by including priority information in the name prefix [33]. In this paper, we assume that the priority information of content is additionally included in the Interest message and thus, network nodes know the priority of content, similar to the works in [23] and [34]. In [34], the priority of content as well as content prefix is assigned by content provider by following a global rule. It is announced by content provider and this information is recorded in routers. In [23], QoS information, i.e., QoS classifier associated with priority, is added to Interest and Data packets of ICN. Contrary to the work in [34], QoS information such as priority is pre-defined based on the type of application in vehicular network environment, and this information is set by consumer when it sends Interest packet.

Fig. 2 shows the overall procedure of the proposed optimal content caching scheme in ICN. Whenever content requests are received at each MEC, it collects the number of content requests. Based on the collected information,

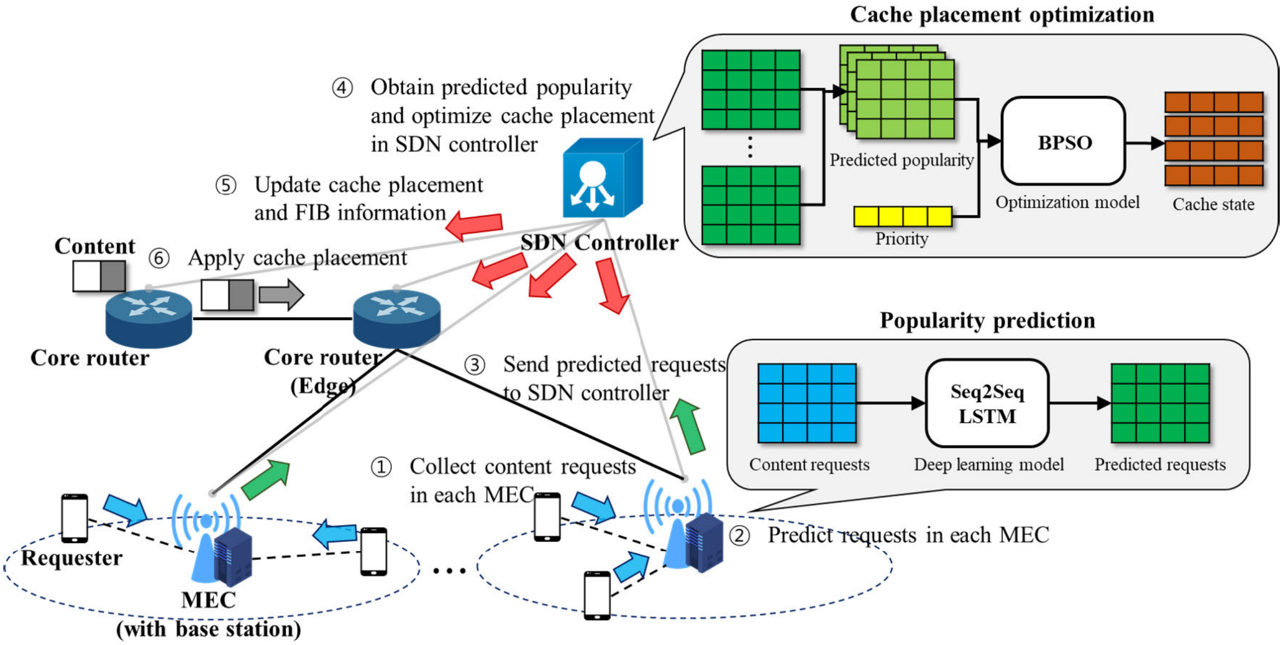

Cache placement optimization
Predicted popularity
BPSO
Optimization model
Cache state
Priority
④ Obtain predicted popularity and optimize cache placement in SDN controller
⑤ Update cache placement and FIB information
⑥ Apply cache placement
Content
SDN Controller
Core router
Core router (Edge)
Popularity prediction
Content requests
Seq2Seq LSTM
Deep learning model
Predicted requests
③ Send predicted requests to SDN controller
① Collect content requests in each MEC
② Predict requests in each MEC
Requester
MEC (with base station)

FIGURE 2. A procedure of the proposed optimal content caching scheme.

each MEC predicts the number of content requests by using seq2seq LSTM model [35], [36], which is widely used deep learning model for time-series prediction [10], [38], [39]. The predicted requests information in each MEC is sent to SDN controller. Then, SDN controller obtains predicted popularity by using predicted requests from all MECs and optimizes cache placement based on the predicted popularity and pre-defined priority of each content. Due to the NP hard property of the formulated optimization problem, BPSO [40], [41] is applied to obtain solution. Now, SDN controller has updated cache placement information, i.e., cache state, for MECs and core routers for next period. Also, SDN controller obtains FIB information for MECs and core routers for next period based on updated cache state, network topology, and content delivery cost between nodes. Updated cache placement information and FIB information are sent to MECs and core routers. Finally, content cache placement is applied in MECs and core routers, and content search is carried out by using updated FIB information. Content requests prediction in MECs is carried out periodically for each stride time window by using information during input time window. Content cache placement in SDN controller is carried out periodically for each stride time window and results in cache state during output time window.

The content cache update can be implemented by using a method, such as "fake content reques" in [38], where content is retrieved by issuing fake content request, although it is not asked by a requester originally. If fake content request is applied to our scheme, either MEC or core router can retrieve required contents by sending Interest message to content providers. In order to focus on cache placement optimization problem, we assume that contents can be cached in MECs or routers successfully in performance evaluation, after receiving content cache placement information from SDN controller, and we do not elaborate content cache placement method, since it is not the main scope of this paper. This is because since the traffic overhead for periodic content retrieval from content provider by using fake content request can be negligible, compared to that for actual content delivery by requesters. Updated FIB information in MECs or routers can be used to forward Interest message to either MEC or router with minimum delivery cost for requested content delivery, and content can be delivered to the requester by using the reverse path of the forwarded Interest. We also assume that we can obtain the content delivery cost to either MEC or router which has the minimum content delivery cost successfully in performance evaluation, without actually forwarding Interest message based on FIB information, by using received cache state information of all MECs and routers. We note that we obtain performance evaluation results in Section III, without actually implementing content cache update procedure based on fake content request and content retrieval procedure based on FIB, by utilizing cache state information efficiently for optimization.

B. CONTENT REQUESTS MODEL

We synthesize content requests received in each MEC for seven days, which consists of six days of training and one day of testing periods. The number of content requests received at MEC m at time t, $r_m^{\mathrm{real}}(t)$, is assumed to follow a Poisson distribution with parameter $\lambda_m(t)$, where the probability mass

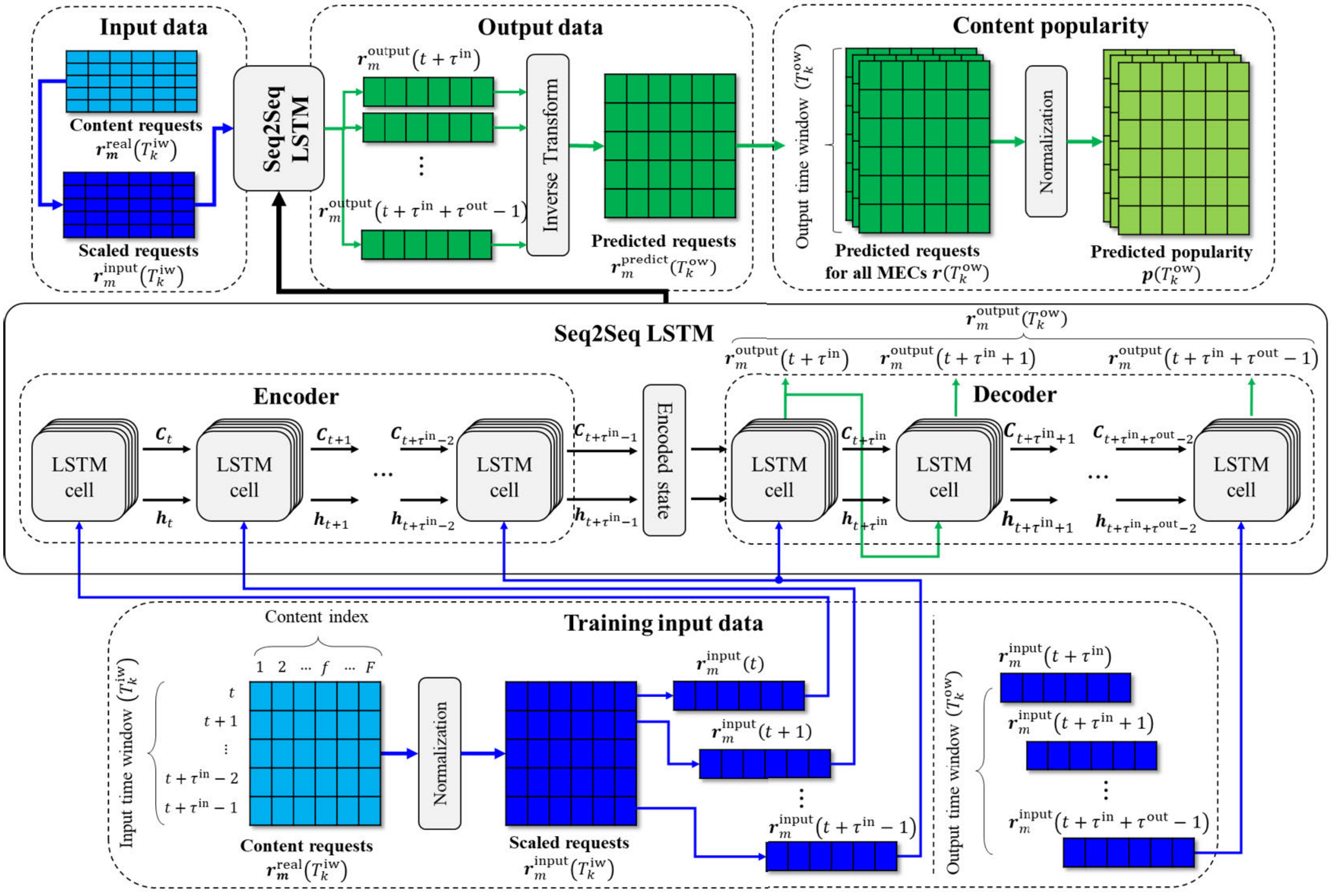

Input data
Content requests
$r_m^{\text{real}}(T_k^{\text{iw}})$
Scaled requests
$r_m^{\text{input}}(T_k^{\text{iw}})$
Seq2Seq LSTM
Output data
$r_m^{\text{output}}(t+\tau^{\text{in}})$
$r_m^{\text{output}}(t+\tau^{\text{in}}+\tau^{\text{out}}-1)$
Inverse Transform
Predicted requests
$r_m^{\text{predict}}(T_k^{\text{ow}})$
Content popularity
Output time window (T_k^{ow})
Predicted requests for all MECs $r(T_k^{\text{ow}})$
Normalization
Predicted popularity $p(T_k^{\text{ow}})$
Seq2Seq LSTM
$r_m^{\text{output}}(T_k^{\text{ow}})$
$r_m^{\text{output}}(t+\tau^{\text{in}})$
$r_m^{\text{output}}(t+\tau^{\text{in}}+1)$
$r_m^{\text{output}}(t+\tau^{\text{in}}+\tau^{\text{out}}-1)$
Encoder
LSTM cell
c_t
h_t
c_{t+1}
h_{t+1}
$c_{t+\tau^{\text{in}}-2}$
$h_{t+\tau^{\text{in}}-2}$
$c_{t+\tau^{\text{in}}-1}$
$h_{t+\tau^{\text{in}}-1}$
Encoded state
Decoder
$c_{t+\tau^{\text{in}}}$
$h_{t+\tau^{\text{in}}}$
$c_{t+\tau^{\text{in}}+1}$
$h_{t+\tau^{\text{in}}+1}$
$c_{t+\tau^{\text{in}}+\tau^{\text{out}}-2}$
$h_{t+\tau^{\text{in}}+\tau^{\text{out}}-2}$
Training input data
Content index
1 2 ⋯ f ⋯ F
Input time window (T_k^{iw})
t
$t+1$
$t+\tau^{\text{in}}-2$
$t+\tau^{\text{in}}-1$
Content requests
$r_m^{\text{real}}(T_k^{\text{iw}})$
Normalization
Scaled requests
$r_m^{\text{input}}(T_k^{\text{iw}})$
$r_m^{\text{input}}(t)$
$r_m^{\text{input}}(t+1)$
$r_m^{\text{input}}(t+\tau^{\text{in}}-1)$
Output time window (T_k^{ow})
$r_m^{\text{input}}(t+\tau^{\text{in}})$
$r_m^{\text{input}}(t+\tau^{\text{in}}+1)$
$r_m^{\text{input}}(t+\tau^{\text{in}}+\tau^{\text{out}}-1)$

FIGURE 3. Content popularity prediction process in the proposed scheme.

function (pmf) of Poisson distribution is defined, as in (1).

$$Pr(r_m^{\text{real}}(t)=k)=\frac{(\lambda_m(t))^k}{k!}e^{-\lambda_m(t)}. \tag{1}$$

In (1), $\lambda_m(t)$ is defined as a variable $\lambda_m g'_m(t)$, where λ_m is a random value obtained from a uniform distribution for each MEC m and is fixed during seven days for each MEC m. In order to have time-varying characteristics for the number of content requests received at each MEC m, $g'_m(t)$ is generated based on Gaussian distribution with mean $\mu_m = 720$ (min) and standard deviation σ_m. The probability density function (pdf) of the mentioned Gaussian distribution $g_m(t)$ for $0 \le t < 1,440$ (min) is defined, as in (2).

$$g_m(t)=\frac{1}{\sigma_m\sqrt{2\pi}}e^{-\frac{1}{2}(\frac{t-\mu_m}{\sigma_m})}, \quad (0 \le t < 1,440). \tag{2}$$

The $g'_m(t)$ is defined as $\tilde{g}_m(t-\mu'_m)$, where $\tilde{g}_m(t)$ is periodic function of $g_m(t)$ and μ'_m is a random value obtained from uniform distribution U[0, 1,440).

In this paper, we assume spatial correlation of content popularity, similar to the works in [8], [16], [38], [42], and [43], and contents' preference in MEC m is assumed to follow Zipf distribution. The pmf of content f is defined, as in (3), where ξ is skewness factor.

$$P_m^{\text{Zipf}}(f)=\frac{f^{-\xi}}{\sum\limits_{j\in\mathcal{F}} j^{-\xi}}. \tag{3}$$

By applying Zipf distribution to $r_m^{\text{real}}(t)$, we can obtain the number of content f's requests in MEC m at time t, $r_{mf}^{\text{real}}(t)$.

C. CONTENT POPULARITY PREDICTION

Fig. 3 shows a detailed process of content popularity prediction. The number of content requests in each MEC is predicted based on seq2seq LSTM model [35], [36], which is also called encoder-decoder model. Seq2seq LSTM is a variant of LSTM, where LSTM is widely used to predict time-series data, and is useful to convert one sequence to another sequence.

As shown in Fig. 3, time-series pattern of content requests in each MEC is used to predict the number of content requests based on seq2seq LSTM. More specifically, each MEC receives content requests $\boldsymbol{r}_m^{\text{real}}(T_k^{\text{iw}}) = \{r_{mf}^{\text{real}}(t)\,|\,m \in \mathcal{M}, \forall f \in \mathcal{F}, \forall t \in T_k^{\text{iw}}\}$ from requesters during input time window $T_k^{\text{iw}} = \{t\,|\,10(k-1) \le t < 10(k-1)+\tau^{\text{in}}\}$, where $k \in \mathcal{K} = \{1, 2, \ldots, K\}$ and τ^{in} is input time window size. The requests $\boldsymbol{r}_m^{\text{real}}(T_k^{\text{iw}})$ is normalized into scaled requests

$\boldsymbol{r}_m^{\text{input}}(T_k^{\text{iw}}) = \{r_{mf}^{\text{input}}(t) \,|\, m \in \mathcal{M}, \forall f \in \mathcal{F}, \forall t \in T_k^{\text{iw}}\}$ with the range [0,1] by min-max normalization. Then they are fed into LSTM cells of seq2seq LSTM encoder as input values, respectively. Encoded state, which is output values of the encoder such as hidden unit $\boldsymbol{h}_{t+\tau^{\text{in}}-1}$ and cell state $\boldsymbol{C}_{t+\tau^{\text{in}}-1}$, as well as $\boldsymbol{r}_m^{\text{input}}(t+\tau^{\text{in}}-1)$ are fed into seq2seq LSTM decoder as input values. Seq2seq LSTM decoder which partially applied teacher forcing method obtains output values, i.e., $\boldsymbol{r}_m^{\text{output}}(T_k^{\text{ow}}) = \{r_{mf}^{\text{output}}(t) \,|\, m \in \mathcal{M}, \forall f \in \mathcal{F}, \forall t \in T_k^{\text{ow}}\}$, where $T_k^{\text{ow}} = \{t \,|\, 10(k-1)+\tau^{\text{in}} \leq t < 10(k-1)+\tau^{\text{in}}+\tau^{\text{out}}\}$ is output time window and τ^{out} is output time window size. Finally, each MEC obtains predicted number of content requests $\boldsymbol{r}_m^{\text{predict}}(T_k^{\text{ow}})$ by de-normalizing $\boldsymbol{r}_m^{\text{output}}(T_k^{\text{ow}})$, which is the output of seq2seq LSTM with $\boldsymbol{r}_m^{\text{input}}(T_k^{\text{iw}})$ as input. Adaptive moment estimation (ADAM) optimizer [37] and mean square error (MSE) loss function are used in our seq2seq LSTM model. Each MEC delivers predicted number of content requests $\boldsymbol{r}_m^{\text{predict}}(T_k^{\text{ow}})$ to SDN controller and SDN controller obtains the predicted number of content requests $\boldsymbol{r}(T_k^{\text{ow}}) = \{\boldsymbol{r}_m^{\text{predict}}(T_k^{\text{ow}}) \,|\, \forall m \in \mathcal{M}\}$ for all MECs. Finally, $\boldsymbol{r}(T_k^{\text{ow}})$ is normalized into predicted popularity $\boldsymbol{p}(T_k^{\text{ow}}) = \{p_{mf}(t) \,|\, \forall m \in \mathcal{M}, \forall f \in \mathcal{F}, \forall t \in T_k^{\text{ow}}\}$ with the range [0,1] by min-max normalization.

D. CONTENT CACHING OPTIMIZATION FORMULATION

In this section, we formulate optimization problem of content cache placement, based on both popularity and priority of contents. We define binary variables x_{if}, for $i \in \mathcal{V} \cup \mathcal{N} \cup \mathcal{M}$ and $f \in \mathcal{F}$, to indicate the cache state of router i, i.e., a set of content providers $\mathcal{V}$, core routers $\mathcal{N}$, and MECs $\mathcal{M}$, as in (4). We note that we call content provider, core router, and MEC collectively as a router for notational convenience. $x_{if} = 1$ if router i caches the content f. On the other hand, $x_{if} = 0$, if the router i does not cache the content f.

$$x_{if} = \begin{cases} 1, & \text{if router } i \text{ caches content } f, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

The content delivery cost from a router which caches content f to a requester is defined as in (5).

$$c_{mf}(\boldsymbol{x}) = \sum_{i \in \mathcal{B}_m} (\gamma_{im} + \gamma_m) x_{if} \prod_{j \in \mathcal{B}'_{mi}} \left(1 - x_{jf}\right), \tag{5}$$

where $\boldsymbol{x} = \{x_{if} \,|\, \forall i \in \mathcal{V} \cup \mathcal{N} \cup \mathcal{M}, \forall f \in \mathcal{F}\}$ is a set of caching state. γ_{im} is delivery cost from router i to MEC m and $\gamma_{im} = 0$ if router i is MEC m itself. γ_m is the delivery cost from MEC m to a requester. $\mathcal{B}_m \subseteq \mathcal{V} \cup \mathcal{N} \cup \mathcal{M}$ is a set of routers with an ascending order of content delivery cost γ_{im} from an MEC serving the considered requester. $\mathcal{B}'_{mi}$, a sub-set of $\mathcal{B}_m$, is a set of routers which include less content delivery cost than router i. Since popularity and priority are two important factors for content caching placement considered in this paper, we use both of them together for cache placement decision, where a set of priority level l is defined $\mathcal{L}$, which includes low, medium and high priority. To do this, a weighted sum of popularity and priority of content f, i.e., $\alpha p_{mf}(t) + (1-\alpha) q_f$ is proposed, where α is a weight factor between the popularity of content f in MEC m, $p_{mf}(t)$ and priority of content f, $q_f \in \{q_{\text{low}}, q_{\text{medium}}, q_{\text{high}}\}$. We note that a weighted sum is widely used technique to combine the values with different units and characteristics appropriately by adjusting the values of weight, as in reference [7], where a weighted sum of latency and energy consumption, which have different units, is used for the formulation of optimization problem to minimize latency and energy consumption in the proposed caching strategy. Then, the content delivery cost considering a weighed sum of popularity and priority of content is defined, as in (6).

$$C(\boldsymbol{x}) = \sum_{t \in T_k^{\text{ow}}} \sum_{m \in \mathcal{M}} \sum_{f \in \mathcal{F}} c_{mf}(\boldsymbol{x}) \big(\alpha p_{mf}(t) + (1-\alpha) q_f\big). \tag{6}$$

With the above definition, the cost minimization problem for optimal cache placement is formulated as follows:

$$\min_{\boldsymbol{x}} \quad C(\boldsymbol{x}), \tag{7a}$$

subject to

$$x_{if} = 1, \quad \forall i \in \mathcal{V}, \quad \forall f \in \mathcal{F}, \tag{7b}$$

$$\sum_{i \in \mathcal{V} \cup \mathcal{N} \cup \mathcal{M}} x_{if} \geq 1, \quad \forall f \in \mathcal{F}, \tag{7c}$$

$$\sum_{f \in \mathcal{F}} x_{if} \leq |\mathcal{S}_i|, \quad \forall i \in \mathcal{V} \cup \mathcal{N} \cup \mathcal{M}, \tag{7d}$$

$$\frac{\sum_{m \in \mathcal{M}} \sum_{f \in \mathcal{F}_l} r_{mf}(t) c_{mf}(\boldsymbol{x})}{\sum_{m \in \mathcal{M}} \sum_{f \in \mathcal{F}_l} r_{mf}(t)} \leq Th_l, \quad \forall l \in \mathcal{L}, t \in T_k^{\text{ow}}, \tag{7e}$$

$$x_{if} \in \{0, 1\}, \quad \forall i \in \mathcal{V} \cup \mathcal{N} \cup \mathcal{M}, \quad \forall f \in \mathcal{F}. \tag{7f}$$

In the above problem, (7a) is an objective function expressed as the sum of weighted content delivery costs and (7b)-(7e) are constraints. Equation (7b) is used to guarantee that the content provider always caches contents generated by themselves. Equation (7c) is used to guarantee that the content is cached at least one router. Equation (7d) is a constraint to guarantee that the total size of cached contents at each router is less than or equal to the buffer memory of the router, where $\mathcal{S}_i$ is a set of contents that router i caches. Equation (7e) is used to guarantee that average QoE of each content with priority level l is smaller than Th_l, similar to the work in [11], where $\mathcal{F}_l$ is a set of contents which have priority level l and $Th_l \in \{Th_{\text{low}}, Th_{\text{low}}, Th_{\text{low}}\}$ is QoE threshold. QoE is defined as the content delivery cost, as in [11]. However, QoE for contents with different priorities was considered separately in this paper, which is different from the work in [11], where content priority is not considered.

To solve optimization problem with inequality constraint efficiently, we rearrange the optimization problem as penalty-based optimization problem [45], [46], as in (8),

Algorithm 1 Content Placement Optimization Procedure

```
Initialization: Content requests $\boldsymbol{r}_m^{\text{real}}(t)$, cache state $\boldsymbol{x}$, prediction process $PRD()$, optimization process $OPT()$, time $t \in \mathcal{T}$, cache placement time index $k \in \mathcal{K}$, MEC set $\mathcal{M}$, content set $\mathcal{F}$ and content priority set $\boldsymbol{q}$
for $t$ do
    for $m \in \mathcal{M}$ do
        if MEC $m$ receives request for content $f$ then
            MEC $m$ collects content requests $r_{mf}^{\text{real}}(t)$
        end if
        if $t = T_k^{ow}$ then
            $\boldsymbol{r}_m^{\text{predict}}(T_k^{\text{ow}}) = PRD(\boldsymbol{r}_m^{\text{real}}(T_k^{\text{iw}}))$
            MEC $i$ sends $\boldsymbol{r}_m^{\text{predict}}(T_k^{\text{ow}})$ to SDN controller
        end if
    end for
    if SDN controller receives $\boldsymbol{r}_m^{\text{predict}}(T_k^{\text{ow}})$ from all MECs then
        SDN controller initializes $\boldsymbol{x}$ and calculates $\boldsymbol{p}(T_k^{\text{ow}})$ by normalizing $\boldsymbol{r}(T_k^{\text{ow}})$
        Find optimal solution $\boldsymbol{x} = OPT\left(\boldsymbol{x}, \boldsymbol{p}(T_k^{\text{ow}}), \boldsymbol{q}, \boldsymbol{r}(T_k^{\text{ow}})\right)$
    end if
end for
```

where ζ is penalty factor and $E(\boldsymbol{x})$ is penalty function.

$$\begin{aligned} J(\boldsymbol{x}, \zeta) &= C(\boldsymbol{x}) + \zeta E(\boldsymbol{x}) \\ &= \sum_{t \in T_k^{\text{ow}}} \sum_{m \in \mathcal{M}} \sum_{f \in \mathcal{F}} c_{mf}(\boldsymbol{x}) \big(\alpha p_{mf}(t) + (1-\alpha) q_f\big) \\ &\quad + \frac{\zeta}{2} \Bigg(\sum_{f \in \mathcal{F}} \max \Big\{0, 1 - \sum_{i \in \mathcal{V} \cup \mathcal{N} \cup \mathcal{M}} x_{if} \Big\}^2 \\ &\quad + \sum_{i \in \mathcal{V} \cup \mathcal{N} \cup \mathcal{M}} \max \Big\{0, \sum_{f \in \mathcal{F}} x_{if} - |\mathcal{S}_i| \Big\}^2 \\ &\quad + \sum_{l \in \mathcal{L}} \max \Bigg\{0, \frac{\sum_{m \in \mathcal{M}} \sum_{f \in \mathcal{F}_l} r_{mf}(t) c_{mf}(\boldsymbol{x})}{\sum_{m \in \mathcal{M}} \sum_{f \in \mathcal{F}_l} r_{mf}(t)} - Th_l \Bigg\}^2 \Bigg). \end{aligned} \tag{8}$$

The formulated optimization problem in this paper is a mixed integer non-linear programming with inequality constraints and it is well-known that the problem is NP-hard [47], [48] and does not scale well. Therefore, we use BPSO which can solve the complex optimization problem fast to optimize (8), in order to obtain numerical results, by using similar method in [45] and [46], and a detailed algorithm of BPSO is not presented in detail due to limited space.

E. ALGORITHM OF THE PROPOSED CONTENT CACHE PLACEMENT OPTIMIZATION

The algorithm 1 shows a detailed algorithm of the proposed content cache placement optimization. An MEC m collects requests of content f for all contents from requesters at each unit time t, i.e., $r_{mf}^{\text{real}}(t)$. An MEC m predicts the number of requests for each output time window T_k^{ow}, i.e., $\boldsymbol{r}_m^{\text{predict}}(T_k^{ow})$ with seq2seq LSTM model, and it sends the predicted requests $\boldsymbol{r}_m^{\text{predict}}(T_k^{ow})$ to SDN controller. SDN controller calculates predicted popularity $\boldsymbol{p}(T_k^{ow})$ by using predicted requests $\boldsymbol{r}^{\text{predict}}(T_k^{ow})$ for all MECs. Then SDN controller decides optimal cache state $\boldsymbol{x}$ for next T_k^{ow} period by solving optimization problem based on predicted popularity $\boldsymbol{p}(T_k^{ow})$, content priority $\boldsymbol{q}$, and predicted requests $\boldsymbol{r}(T_k^{ow})$, where $\boldsymbol{q} = \{q_f \mid \forall f \in \mathcal{F}\}$.

III. NUMERICAL RESULTS

In this section, we carry out simulation with simulation settings, as in Table 1, where the values of input, output, and stride time windows are assumed to be 60, 10, and 10 minutes, respectively, to tradeoff between accuracy and complexity. In this paper, we used Nvidia GeForce RTX 3080 Ti GPU for simulation. Seq2seq LSTM-based training and testing were carried out in each MEC, with 6 days (8,640 minutes) and 1 day (1,440 minutes) datasets, respectively. For training, measured memory usage and execution time for 6 days of training dataset are 1,941 Mbytes and 35.74 seconds, respectively. The training was executed before simulation via off-line. For testing, measured memory usage and execution time for 60 minutes of testing dataset are 1,869 Mbytes and 11.6 milliseconds, respectively, where testing is executed periodically for every 10 minutes for 1 day of dataset. We note that operational overheads for the seq2seq LSTM-based prediction for testing, i.e., memory usage and execution time, are not significant, and expect that the overhead can be accommodated in commercial MEC, without much difficulty, which supports multiple GPUs with sufficient memory.

We obtain performance evaluation results, from the aspect of QoE satisfaction ratio, average cost, weighted average cost, total cost, and weighted total cost as defined in (9)-(13), where $\mathbb{1}$ in (9) is an indicator function and $\boldsymbol{x}(t)$ is a cache state at time t. QoE satisfaction ratio for content with priority level l is defined the ratio of content requests which satisfy the QoE constraint for content with priority level l. Average cost for a content with priority level l is defined as the total sum of content delivery cost for all the content requests with priority level l divided by the total number of content requests with priority level l. In order to show the effect of content priority on the delivery cost, priority of content with priority level l is multiplied to the content delivery cost for content with priority level l for the calculation of weighted average cost for a content with priority level l. Total cost at time t is defined as the total sum of content delivery cost for all the requests at time t. Weighted total cost at time t is defined as the total sum of weighted content delivery cost for all the requests at time t.

In (10) and (12), the cost of content delivery does not depend on the priority of considered content f. However, the priority of each content f is multiplied to the delivery cost of the content f in (11) and (13). It is assumed that higher value of q_f corresponds to higher priority content. This is because we want to differentiate the delivery cost of content f, depending on the priority of the content, and thus, delivery cost of high priority content is treated more importantly to satisfy the QoE of requesters.

QoE satisfaction ratio(l)

$$= \frac{\sum_{t\in\mathcal{T}}\sum_{m\in\mathcal{M}}\sum_{f\in\mathcal{F}_l} \mathbb{1}\left(c_{mf}\left(\boldsymbol{x}(t)\right) \le Th_l\right) r_{mf}^{\text{real}}(t)}{\sum_{t\in\mathcal{T}}\sum_{m\in\mathcal{M}}\sum_{f\in\mathcal{F}_l} r_{mf}^{\text{real}}(t)}. \tag{9}$$

$$\text{Average cost}(l) = \frac{\sum_{t\in\mathcal{T}}\sum_{m\in\mathcal{M}}\sum_{f\in\mathcal{F}_l} c_{mf}\left(\boldsymbol{x}(t)\right) r_{mf}^{\text{real}}(t)}{\sum_{t\in\mathcal{T}}\sum_{m\in\mathcal{M}}\sum_{f\in\mathcal{F}_l} r_{mf}^{\text{real}}(t)}. \tag{10}$$

Weighted average cost(l)

$$= \frac{\sum_{t\in\mathcal{T}}\sum_{m\in\mathcal{M}}\sum_{f\in\mathcal{F}_l} c_{mf}\left(\boldsymbol{x}(t)\right) r_{mf}^{\text{real}}(t) q_f}{\sum_{t\in\mathcal{T}}\sum_{m\in\mathcal{M}}\sum_{f\in\mathcal{F}_l} r_{mf}^{\text{real}}(t)}. \tag{11}$$

$$\text{Total cost}(t) = \sum_{m\in\mathcal{M}}\sum_{f\in\mathcal{F}} c_{mf}\left(\boldsymbol{x}(t)\right) r_{mf}^{\text{real}}(t). \tag{12}$$

$$\text{Weighted total cost}(t) = \sum_{m\in\mathcal{M}}\sum_{f\in\mathcal{F}} c_{mf}\left(\boldsymbol{x}(t)\right) r_{mf}^{\text{real}}(t) q_f. \tag{13}$$

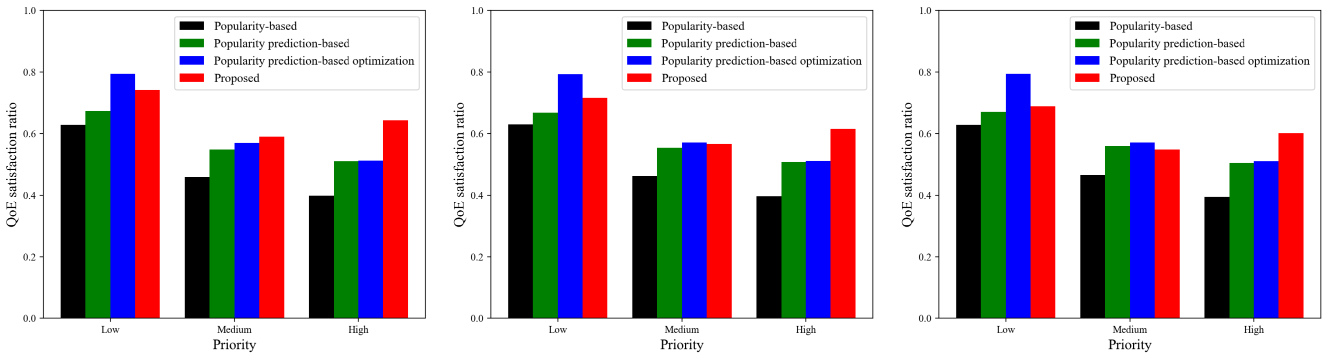
For performance comparison, we broadly classify works on content caching as 1) popularity-based caching strategy, 2) popularity prediction-based caching strategy, and 3) popularity prediction-based optimization caching strategy. In popularity-based caching strategy, each node calculates the popularity of content, primarily as the number of content request received, and if the current popularity reaches a threshold, it caches the content in its cache. In this caching strategy, there is no popularity prediction but it uses currently measured popularity. Works in [21] and [49] can be regarded as popularity-based caching strategy. In popularity prediction-based caching strategy, however, popularity is predicted based on machine learning at each node and caching is carried out at each node locally based on the predicted popularity. Works in [8], [9], and [38] fall into the category of popularity prediction-based caching strategy. In popularity prediction-based optimization caching strategy, each node predicts popularity based on machine learning but predicted popularity is collected by a centralized node, such as SDN controller, and caching decision is made by a centralized node globally by solving an optimization problem to minimize cost, latency, energy consumption, etc. Work in [7] is an example of popularity prediction-based optimization caching strategy.

In this paper, three compared schemes for performance comparison are defined as follows, where the main concepts of the mentioned caching strategies are used appropriately to define the compared schemes:

TABLE 1. Simulation settings.

Parameter		Value
Content	Number of contents	100
	Size of content	1
	q_{low}	0.25
	q_{medium}	0.5
	q_{high}	1
	popularity threshold	100
	popularity count reset	0
	N_{top}	10
Number of routers	MEC (per edge router)	4
	Edge router (per core router)	2
	Core router (per root router)	2
	Root router	1
	Content provider	1
Cache size	MEC	10
	Core router	10
	Content provider	Infinite
Delivery cost	Requester and MEC	1
	MEC and edge router	2
	Edge router and core router	4
	Core router and root router	7
	root router and content provider	11
Request model	λ_m (per day)	U[1.44×10^6, 2.88×10^6]
	μ_m (min)	720
	μ'_m (min)	U[0, 1,440)
	σ_m (min)	U[540, 900]
	ξ	0.8
Simulation time	Training (min)	8,640
	Test (min)	1,440
	Input time window (min)	60
	Output time window (min)	10
	Stride time window (min)	10
Seq2seq LSTM	Epoch	500
	Batch size	128
	Learning rate	0.001
	Hidden layer size	500
	Number of LSTM layer	1
	Teacher forcing ratio	0.6
Optimization	α	0.5
	Th_{low}	15
	Th_{medium}	12
	Th_{high}	9

- **Popularity-based scheme**: In popularity-based scheme, each node calculates popularity as the number content requests, and if the value of popularity, i.e., popularity count, reaches a popularity threshold value, the node suggests caching the content to neighbor nodes if it has the content in its cache already [49]. The neighbor nodes which receive the recommendation caches the content, and finally, the popularity count is reinitialized to a popularity count reset value to prevent flooding of the same content [49]. We assume least frequently used (LFU) cache replacement strategy [50] in this scheme.
- **Popularity prediction-based scheme**: In popularity prediction-based scheme, popularity is defined as the normalized value of the predicted number of content requests based on seq2seq LSTM model, and each router caches contents within top N_{top} popularity, similar to the work in [38].
- **Popularity prediction-based optimization scheme**: In popularity prediction-based optimization scheme,



(a) low:medium:high=0.6:0.3:0.1

(b) low:medium:high=0.5:0.3:0.2

(c) low:medium:high=0.4:0.3:0.3

FIGURE 4. QoE satisfaction ratio.

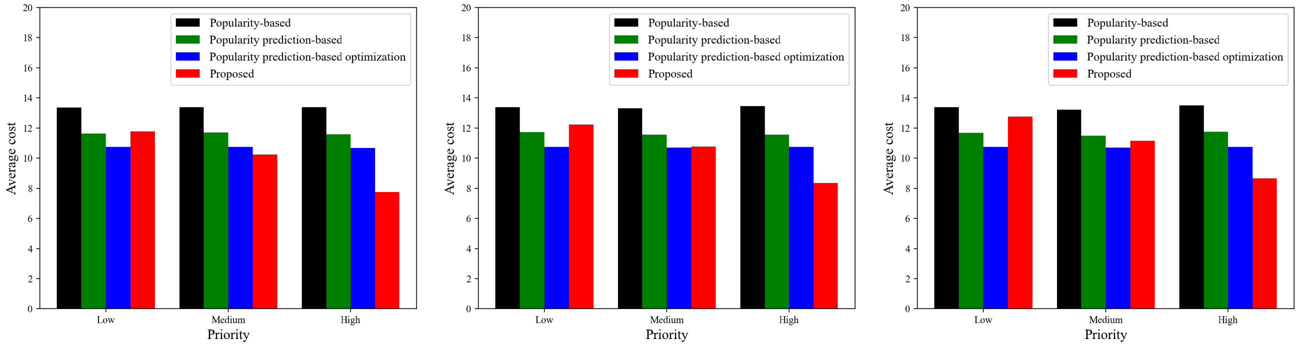
popularity is defined as the normalized value of the predicted number of content requests based on seq2seq LSTM model and predicted popularity is used for content cache optimization with objective function $C(\boldsymbol{x}) = \sum_{t\in T_k^{\mathrm{ow}}} \sum_{m\in\mathcal{M}} \sum_{f\in\mathcal{F}} c_{mf}(\boldsymbol{x})p_{mf}(t)$ and constraints (7b)-(7d), (7f). The work in [7] is similar to this scheme, in a sense that the number of content requests is predicted based on machine learning and is used for content cache optimization.

In this paper, simulation was carried out by using 100 data sets of contents with different combinations of popularity and priority. In Figs. 4-8, we consider three sets of ratios among contents with low, medium, and high priority levels to show the effect of the ratio among contents with different priority levels. Fig. 4 compares the QoE satisfaction ratio of the proposed scheme with three compared schemes. Since the popularity-based, popularity prediction-based and popularity prediction-based optimization schemes are irrelevant to the priority of content, QoE satisfaction ratios of those schemes for a given priority are not sensitive to the change of the ratio among different priority levels. However, the QoE satisfaction ratio of the proposed scheme for a given priority decreases as the ratio of content with high priority level increases for all low, medium, and high priority contents. This is because it is less likely to be able to place content with high priority in nodes which can satisfy QoE constraint, if the ratio of high priority content increases. Also, since more buffer space with less content delivery cost from requesters is used by high priority content, there is less buffer space for medium and low priority contents, which results in the decrease of QoE satisfaction ratio of medium and low priority contents. The proposed scheme has higher QoE satisfaction ratio for content with high priority level, compared to other schemes, although it has lower QoE satisfaction ratio for content with low priority level, compared to popularity prediction-based optimization scheme. The results show that the proposed scheme is very effective at satisfying QoE for content with high priority level, which is required for service differentiation in limited buffer memory environment.

Fig. 5 compares the average cost of the proposed scheme with three compared schemes. Similar to the results in Fig. 4, the average costs of popularity-based, popularity prediction-based, and popularity prediction-based optimization schemes are not sensitive to the change of the ratio among different priority levels. The average cost of the proposed scheme decreases as priority increases, since stricter QoE constraint is applied to content with higher priority, which results in less content delivery cost. The proposed scheme has lower average cost for content with high priority level than other schemes, although it has higher average cost for content with low priority level than popularity prediction-based and popularity prediction-based optimization schemes. This is because the proposed scheme has smaller average cost for content with high priority due to stricter QoE constraint. However, since the buffer memory is limited, more caching of contents with high priority level in routers which have less content delivery cost results in less caching of contents with low priority in nodes which have less content delivery cost. The performance improvement for content with high priority level is more significant when the ratio of high priority content is smaller.

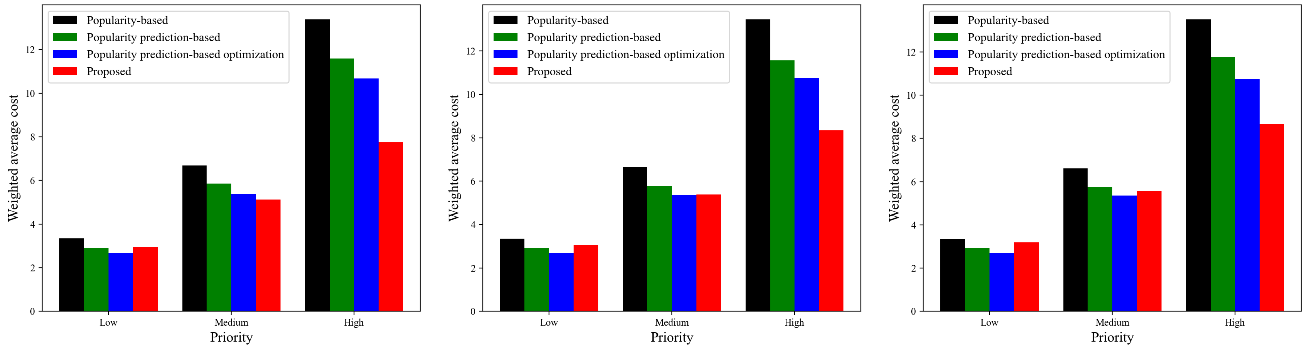
Fig. 6 compares the weighted average cost of the proposed scheme with three compared schemes, where priority value of each content is multiplied to the delivery cost of the content for each content request. The weighted average costs of four schemes increase as priority level increases but the rate of increase of the proposed scheme is less than compared schemes, since the average cost decreases as priority level of content increases, as shown in Fig. 5, in the proposed scheme. The proposed scheme has lower weighted average cost than other schemes when the priority is high. This shows that the proposed scheme is more effective for caching and delivering content with high priority than other compared schemes.

Fig. 7 compares the total cost of the proposed scheme with three compared schemes for varying simulation time, where average costs of all content requests with different



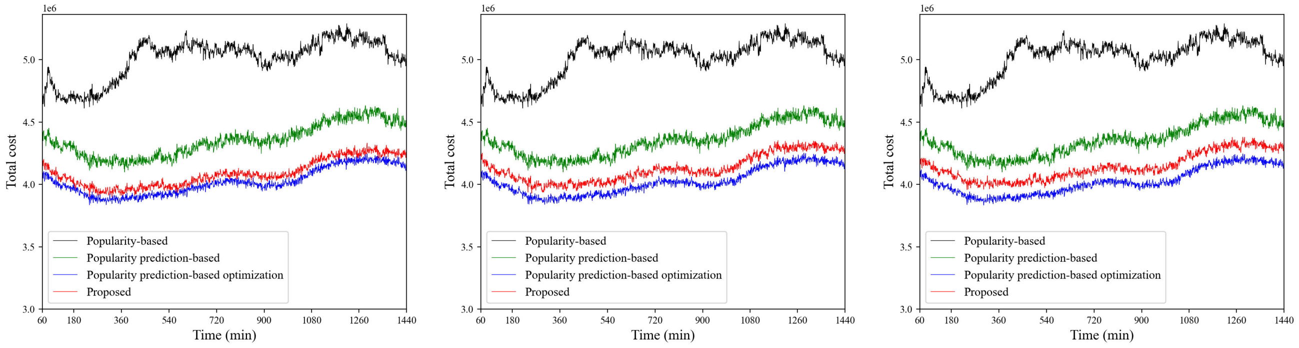
(a) low:medium:high=0.6:0.3:0.1 (b) low:medium:high=0.5:0.3:0.2 (c) low:medium:high=0.4:0.3:0.3

FIGURE 5. Average cost.



(a) low:medium:high=0.6:0.3:0.1 (b) low:medium:high=0.5:0.3:0.2 (c) low:medium:high=0.4:0.3:0.3

FIGURE 6. Weighted average cost.

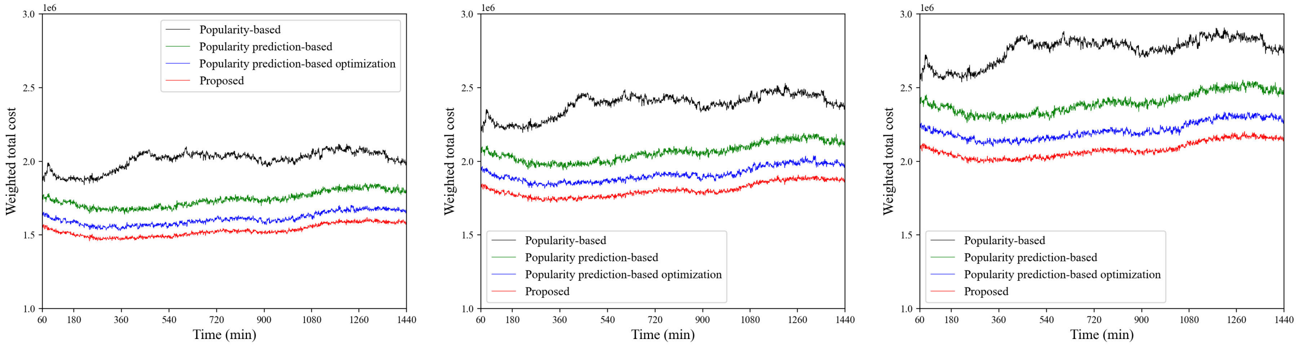


(a) low:medium:high=0.6:0.3:0.1 (b) low:medium:high=0.5:0.3:0.2 (c) low:medium:high=0.4:0.3:0.3

FIGURE 7. Total cost.

priority levels are added together. As expected from the results in Fig. 5, popularity-based scheme has the largest total cost. The proposed scheme has a little bit larger total cost than popularity prediction-based optimization scheme, since the proposed scheme consider priority in addition to popularity, which results in less efficient content caching than the scheme considering popularity only, from the aspect of content delivery cost. We note that the total cost of the proposed scheme becomes closer to that of the popularity prediction-based optimization, when the ratio of high priority is smaller, due to the less effect of high priority content on the total content delivery cost.

Fig. 8 compares the weighted total cost of the proposed scheme with three compared schemes for varying simulation time, where average costs of all content requests with different priority levels are added together and priority of



(a) low:medium:high=0.6:0.3:0.1

(b) low:medium:high=0.5:0.3:0.2

(c) low:medium:high=0.4:0.3:0.3

FIGURE 8. Weighted total cost.

content is considered for cost calculation. Popularity-based scheme has the largest weighted total cost and the proposed scheme has the smallest weighted total cost, as can be expected from the results in Fig. 6. This shows that the proposed scheme is very effective for caching and delivering contents with high priority than other compared schemes, from the aspect of weighted total cost.

IV. CONCLUSION

In this paper, we proposed a content caching network architecture, which includes edge network and core network, with SDN controller. The number of content requests was predicted in MECs based on seq2seq LSTM and it was sent to SDN controller. SDN controller obtains popularity prediction based on the received predicted requests from all MECs and decides optimal content placement by using content popularity and content priority, where QoE of content with different priority levels is constrained. A detailed optimization formulation was presented and the operation of the proposed scheme was explained in detail. The performance of the proposed scheme was compared with that of popularity-based, popularity prediction-based, and popularity prediction-based optimization schemes, from the aspect of QoE satisfaction ratio, average cost, weighted average cost, total cost, and weighted total cost, by using BPSO. The proposed scheme had higher QoE satisfaction ratio for content with high priority than other schemes. Also, although the proposed scheme had higher average cost and weighted average cost for content with low priority, it had lower average cost and weighted average cost for content with high priority. Also, the proposed scheme had smallest weighted total cost, although it had larger total cost than popularity prediction-based optimization scheme. These results showed the effectiveness of the proposed scheme at caching content with high priority efficiently, at the expense of caching content with low priority.

REFERENCES

[1] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, "A survey on low latency towards 5G: RAN, core network and caching solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3098–3130, 4th Quart., 2018.

[2] M. I. A. Zahed, I. Ahmad, D. Habibi, Q. V. Phung, M. M. Mowla, and M. Waqas, "A review on green caching strategies for next generation communication networks," *IEEE Access*, vol. 8, pp. 212709–212737, 2020.

[3] 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, 4th Quart., 2017.

[4] J. Yao, T. Han, and N. Ansari, "On mobile edge caching," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2525–2553, 3rd Quart., 2019.

[5] X. Jiang, F. R. Yu, T. Song, and V. C. M. Leung, "A survey on multi-access edge computing applied to video streaming: Some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 871–903, Mar. 2021.

[6] Y. Zeng, J. Xie, H. Jiang, G. Huang, S. Yi, N. Xiong, and J. Li, "Smart caching based on user behavior for mobile edge computing," *Inf. Sci.*, vol. 503, pp. 444–468, Nov. 2019.

[7] C. Li, Y. Zhang, Q. Sun, and Y. Luo, "Collaborative caching strategy based on optimization of latency and energy consumption in MEC," *Knowl.-Based Syst.*, vol. 233, pp. 1–18, Dec. 2021.

[8] W.-X. Liu, J. Zhang, Z.-W. Liang, L.-X. Peng, and J. Cai, "Content popularity prediction and caching for ICN: A deep learning approach with SDN," *IEEE Access*, vol. 6, pp. 5075–5089, 2018.

[9] L. Ale, N. Zhang, H. Wu, D. Chen, and T. Han, "Online proactive caching in mobile edge computing using bidirectional deep recurrent neural network," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5520–5530, Jun. 2019.

[10] J. Liang, D. Zhu, H. Liu, H. Ping, T. Li, H. Zhang, L. Geng, and Y. Liu, "Multi-head attention based popularity prediction caching in social content-centric networking with mobile edge computing," *IEEE Commun. Lett.*, vol. 25, no. 2, pp. 508–512, Oct. 2021.

[11] S. Khanal, K. Thar, and E.-N. Huh, "DCoL: Distributed collaborative learning for proactive content caching at edge networks," *IEEE Access*, vol. 9, pp. 73495–73505, 2021.

[12] T.-V. Nguyen, N.-N. Dao, V. Dat Tuong, W. Noh, and S. Cho, "User-aware and flexible proactive caching using LSTM and ensemble learning in IoT-MEC networks," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3251–3269, Mar. 2022.

[13] A. Lekharu, M. Jain, A. Sur, and A. Sarkar, "Deep learning model for content aware caching at MEC servers," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 2, pp. 1413–1425, Jun. 2022.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[15] C. Zhang, H. Pang, J. Liu, S. Tang, R. Zhang, D. Wang, and L. Sun, "Toward edge-assisted video content intelligent caching with long short-term memory learning," *IEEE Access*, vol. 7, pp. 152832–152846, 2019.

[16] H. Mou, Y. Liu, and L. Wang, "LSTM for mobility based content popularity prediction in wireless caching networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6.

[17] Y. Fu, Z. Yang, T. Q. S. Quek, and H. H. Yang, "Towards cost minimization for wireless caching networks with recommendation and uncharted Users' feature information," *IEEE Trans. Wireless Commun.*, vol. 20, no. 10, pp. 6758–6771, May 2021.

[18] R. Ullah, M. A. U. Rehman, M. A. Naeem, B.-S. Kim, and S. Mastorakis, "ICN with edge for 5G: Exploiting in-network caching in ICN-based edge computing for 5G networks," *Future Gener. Comput. Syst.*, vol. 111, pp. 159–174, Oct. 2020.

[19] O. Serhane, K. Yahyaoui, B. Nour, and H. Moungla, "A survey of ICN content naming and in-network caching in 5G and beyond networks," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4081–4104, Sep. 2021.

[20] X. Jiang, T. Zhang, and Z. Zeng, "Content clustering and popularity prediction based caching strategy in content centric networking," in *Proc. IEEE 85th Veh. Technol. Conf. (VTC Spring)*, Jun. 2017, pp. 1–5.

[21] S. Hassan, I. U. Din, A. Habbal, and N. H. Zakaria, "A popularity based caching strategy for the future internet," in *Proc. ITU Kaleidoscope, ICTs Sustain. World (ITU WT)*, Bangkok, Thailand, Nov. 2016, pp. 1–8.

[22] A. Jangam, P. Suthar, and M. Stolic, *QoS Treatments in ICN Using Disaggregated Name Components*, IETF ICNRG, document draft-anilj-icnrg-dnc-qos-icn-02, 2020.

[23] J. McCarthy, S. R. Chaudhry, P. Kuppuudaiyar, R. Loomba, and S. Clarke, "QoSA-ICN: An information-centric approach to QoS in vehicular environments," *Veh. Commun.*, vol. 30, pp. 1–19, Aug. 2021.

[24] L. V. Yovita, N. R. Syambas, and I. Y. Matheus Edward, "CAPIC: Cache based on popularity and class in named data network," in *Proc. Int. Conf. Control, Electron., Renew. Energy Commun. (ICCEREC)*, Bandung, Indonesia, Dec. 2018, pp. 24–29.

[25] L. V. Yovita, N. R. Syambas, and I. Y. M. Edward, "Cache based on popularity and class in mobile named data network," in *Proc. IEEE Asia Pacific Conf. Wireless Mobile (APWiMob)*, Bali, Indonesia, Nov. 2019, pp. 105–111.

[26] L. V. Yovita, N. R. Syambas, I. J. M. Edward, and N. Kamiyama, "Performance analysis of cache based on popularity and class in named data network," *Future Internet*, vol. 12, no. 12, p. 227, Dec. 2020.

[27] L. V. Yovita, N. R. Syambas, and I. J. M. Edward, "Weighted-CAPIC caching algorithm for priority traffic in named data network," *Future Internet*, vol. 14, no. 3, p. 84, Mar. 2022.

[28] M. W. Kang and Y. W. Chung, "A content caching optimization scheme for information-centric networking in multi-access edge computing," in *Proc. KICS Fall Conf.*, 2021, pp. 362–363.

[29] M. W. Kang and Y. W. Chung, "Content caching scheme based on LSTM in 5G MEC," in *Proc. KICS Winter Conf.*, 2022, pp. 501–502.

[30] NDN Project Team. *NDN Packet Format Specification (Version 0.3)*. [Online]. Available: https://docs.named-data.net/NDN-packet-spec/current/

[31] M. Mosko, I. Solis, and C. A. Wood, *Content-Centric Networking (CCNx) Messages in TLV Format*, IETF ICNRG, document RFC 8609, Jul. 2019. [Online]. Available: https://www.rfc-editor.org/rfc/pdfrfc/rfc8609.txt.pdf

[32] D. R. Oran, *Considerations in the Development of a QoS Architecture for CCNx-Like Information-Centric Networking Protocols*, IETF ICNRG, RFC 9064, Jun. 2021. [Online]. Available: https://www.rfc-editor.org/rfc/rfc9064.pdf

[33] I. Psaras, L. Saino, M. Arumaithurai, K. K. Ramakrishnan, and G. Pavlou, "Name-based replication priorities in disaster cases," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Toronto, ON, Canada, Apr. 2014, pp. 434–439.

[34] M. Aamir, "Content-priority based interest forwarding in content centric networks," 2014, *arXiv:1410.4987*.

[35] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, 2014, pp. 1724–1734.

[36] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process Syst.*, Montreal, QC, Canada, 2014, pp. 3104–3112.

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, 2015, pp. 1–15.

[38] A. Narayanan, S. Verma, E. Ramadan, P. Babaie, and Z.-L. Zhang, "Making content caching policies 'smart' using the deepcache framework," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 48, no. 5, pp. 64–69, Oct. 2018.

[39] C. Tirupathi, B. Hamdaoui, and A. Rayes, "HybridCache: AI-assisted cloud-RAN caching with reduced in-network content redundancy," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Taipei, Taiwan, Dec. 2020, pp. 1–6.

[40] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. Comput. Cybern. Simul.*, Orlando, FL, USA, Oct. 1997, pp. 4104–4108.

[41] Y. Valle, G. K. Venayagamoorthy, S. Mohagheghi, J. C. Hernandez, and R. G. Harley, "Particle swarm optimization: Basic concepts, variants and applications in power systems," *IEEE Trans. Evol. Comput.*, vol. 12, no. 2, pp. 171–195, Apr. 2008.

[42] H. Nakayama, S. Ata, and I. Oka, "Caching algorithm for content-oriented networks using prediction of popularity of contents," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, Ottawa, ON, Canada, May 2015, pp. 1171–1176.

[43] Z. Xiaoqiang, Z. Min, and W. Muqing, "An in-network caching scheme based on betweenness and content popularity prediction in content-centric networking," in *Proc. IEEE 27th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Valencia, Spain, Sep. 2016, pp. 1–6.

[44] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, and E. Dutkiewicz, "A novel mobile edge network architecture with joint caching-delivering and horizontal cooperation," *IEEE Trans. Mobile Comput.*, vol. 20, no. 1, pp. 19–31, Jan. 2021.

[45] K. Masuda, K. Kurihara, and E. Aiyoshi, "A penalty approach to handle inequality constraints in particle swarm optimization," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Istanbul, Turkey, Oct. 2010, pp. 2520–2525.

[46] A. R. Jordehi, "A review on constraint handling strategies in particle swarm optimisation," in *Neural Comput. Appl.*, vol. 26, pp. 1265–1275, Jan. 2015.

[47] G. De Melo, "Not quite the same: Identity constraints for the web of linked data," in *Proc. AAAI Conf. Artif. Intell.*, Bellevue, WA, USA, 2013, pp. 1092–1098.

[48] M. Gavanelli, M. Milano, S. Bragaglia, F. Chesani, E. Marengo, and P. Cagnoli, "Multi-criteria optimal planning for energy policies in CLP," in *Proc. CILC*, 2014, pp. 54–68.

[49] C. Bernardini, T. Silverston, and O. Festor, "MPC: Popularity-based caching strategy for content centric networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Budapest, Hungary, Jun. 2013, pp. 3619–3623.

[50] H. Che, Y. Tung, and Z. Wang, "Hierarchical web caching systems: Modeling, design and experimental results," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 7, pp. 1305–1314, Sep. 2002.

MIN WOOK KANG received the B.S. degree in electronic engineering from Soongsil University, in 2015, and the M.S. degree from the Department of Information and Communication, Soongsil University, in 2017, where he is currently pursuing the Ph.D. degree with the Department of Information and Communication. His research interests include delay tolerant networks (DTN), information-centric networking (ICN), 5G multi-access/mobile edge computing (MEC), and artificial intelligence/machine learning (AI/ML).

YUN WON CHUNG (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 1995, 1997, and 2001, respectively. From October 2001 to December 2002, he was a Visiting Postdoctoral Research Fellow at the Centre for Telecommunications Research, King's College London, U.K. From January 2003 to August 2005, he was with the Electronics and Telecommunications Research Institute (ETRI), Daejeon. In September 2005, he joined the Faculty of Soongsil University, Seoul, South Korea, where he is currently a Full Professor with the School of Electronic Engineering. His research interests include performance analysis of various mobile and wireless communication networks, delay/disruption-tolerant networking (DTN), information-centric networking (ICN), and artificial intelligence/machine learning (AI/ML).