Content Placement Problem in a Hierarchical Collaborative Caching method for 5G networks (CPP-HCC)

Farnaz Hassanzadeh, Ertan Onur
Department of Computer Engineering, ODTU, Ankara, Turkey
{farnaz.hassanzadeh, eronur}@metu.edu.tr

***Abstract*—The increasing demand for video streaming has imposed tremendous data rates and minimal end-to-end latency requirements on 5G mobile networks. Caching content close to the users is one of the conventional ways to meet these requirements. Subsequent requests for the same content can be supplied from the cache with minimal delay. In this paper, we present a content placement problem in a hierarchical collaborative caching (CPP-HCC) in 5G networks that can determine the location of the replica contents by solving an optimization problem. This optimization problem minimizes the latency for transferring content between entities of the network. The evaluation results confirm the efficiency of CPP-HCC compared to other benchmark methods and show that the latency and hit ratio can be improved by 83% and 62%, respectively.**

***Index Terms*—5G mobile networks, CPP, HCC, optimization problem**

I. Introduction

According to Cisco's Visual Networking Index report, mobile video traffic will account for more than 78 percent of all forms of data traffic by 2021 [1]. This dramatically increasing demand for video has imposed a considerable data rate and minimal end-to-end latency requirements on the next generation communication systems. These requirements are referred to as *gigabit experience* and zero latency in the scope of the fifth-generation (5G) networks, respectively [2].

In 5G mobile networks, users expect to access rapidly and reliably videos that require large channel capacities. Deploying services and applications closer to the end-users improves the performance of the network. To this aim, Content Delivery Networks (CDNs) will play a significant role in 5G mobile networks. Content can be easily provided from cache servers in CDNs offering globally distributed networks of caches. Caches can be used as replica servers to store the replica contents of the requests. Whenever there is a need for a subsequent request, the content can be served from nearby caches that will reduce delay.

Although using caches increase the performance of the network, efficient caching is still challenging. In the literature, caching has been verified from three different aspects: decision whether or not to cache, placement, and replacement strategies [3]. In this article, we consider the impact of the placement strategy on the performance of the network and leave the other methods as a future work. Cache placement strategy determines places for deploying the replica content and servers of a CDN near users to provide better services for users and get the maximum efficiency [4].

In 5G mobile networks, caches can be deployed locally on devices for device-to-device (D2D) communication, at small base stations (SBS), and macro base stations (MBS) of Radio Access Network (RAN) [5], in routers of transport network or in the core network (CN) as shown in Fig. 1. The requested contents are hierarchically searched from the nearest cache to the source until found.

UE caching is comprehensively investigated in [6]. End-users share their content using direct wireless communication links (D2D). Since the end-users are close to each other, the latency will reduce significantly. However, because of the limited capacity issues in UEs, the hit ratio is smaller than the other cache deployment methods. Moreover, in the 5G network, there will be some challenges while implementing D2D content sharing including the speed of mobile users, privacy, security, and resource management [6]. Due to financial issues, the storage capacity in levels of the network will be increased hierarchically. It means caches in the CN and UEs have the largest and smallest capacity, respectively. Caches in higher levels (CN and aggregation level) serve a massive number of users. As a result, the hit ratio will be increased more compared to the other low level caches. However, because of the long distance between the end-users and higher level caches, the latency will be increased. When the caches are located in UE and access networks, the backhaul links are not used and helps the bandwidth shortage problem in the 5G network [7].

Although the presented articles for the caching deployment and content placement demonstrated significant improvement in the quality of the systems, most of them consider the content caching only in one or two levels of the network (RAN or CN), and ignore the collaboration between and inside levels. Besides, a few works have discussed the hierarchical optimization problem in placement methods to find the optimal position of the caches and contents.

Therefore, this article differs by considering the collaboration

978-3-903176-31-7 © 2020 IFIP

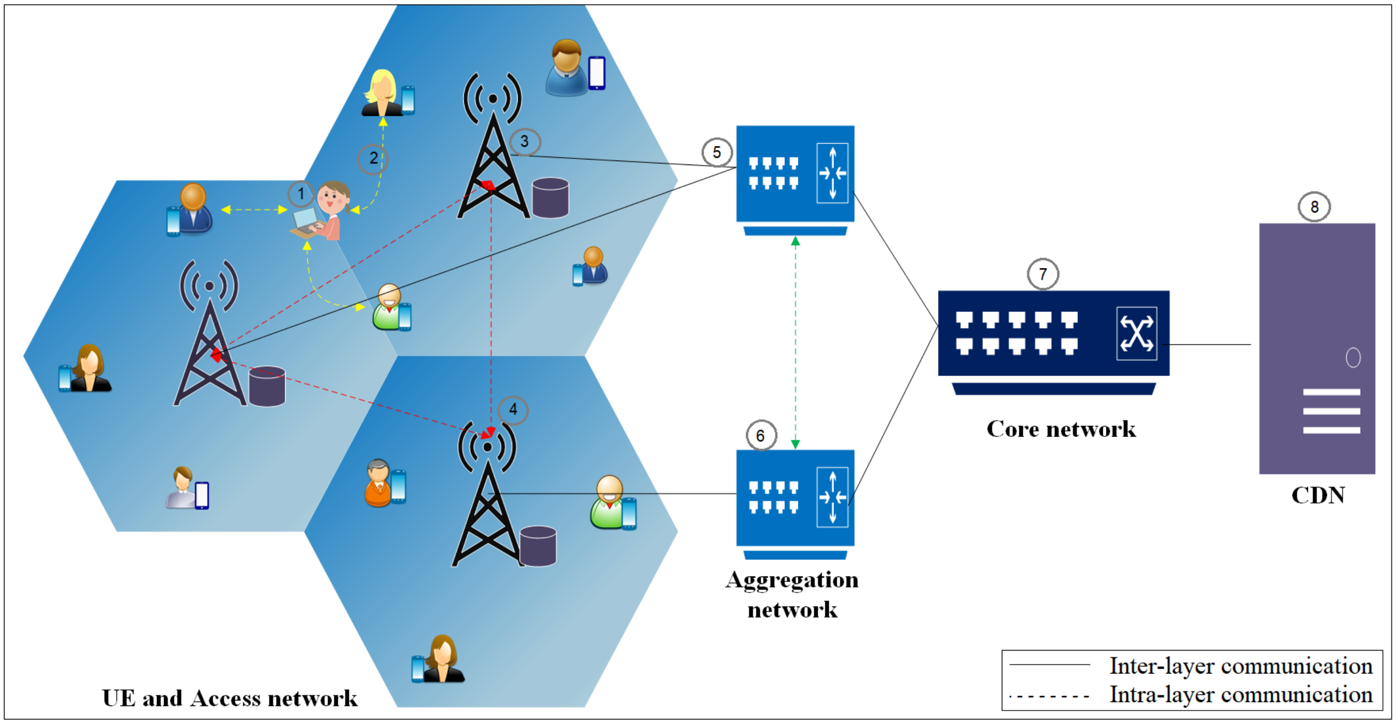

Fig. 1. 5G network architecture.

of caches in different levels of the network and presents a new hierarchical model to deploy the caches in the entities of the 5G network architecture. This proposed method not only considers inter-level communications but also brings up intra-level connections between tiers, which increases the hit ratio and decreases the delay of the whole network as we show in this paper. To reach this issue, we formulate an optimization problem for minimizing the cost of deploying content in 5G networks.

In comparison to the related work, our main contributions in this paper are as follows:

- We design a hierarchical collaborative caching (HCC) in 5G network architecture that consists of four levels, including user equipment (UE), access, aggregation, and core network (CN) in section II.
- We formulate the content placement problem (CPP) and show that the optimal solution to this problem for minimizing the cost of the locating replica content in the network is NP-hard in section II.
- Given the output of CPP into HCC as an input, we develop an efficient caching method in 5G networks (CPP-HCC) in section III.
- We evaluate our proposed method using OMNet++ simulator and show that the performance of the network will improve compared to the other reference methods named leave copy everywhere (LCE) and leave copy down (LCD) caching methods [8] in section III. In LCE, retrieved content will be stored in every cache it passes, while in LCD, it will be stored just in the cache that is the direct successor of the cache, which generates a hit.
- At last, we discuss related work and collaborative caching strategies in literature in Section IV and conclude the paper in Section V.

II. Hierarchical Collaborative Caching (HCC)

The generic 5G architecture is illustrated in Fig. 1. It consists of four main levels named user equipment (UE), radio access networks (RANs) known as access, aggregator, and core network (CN). The replica servers should be located in these levels in a way to get maximum performance of the network and hence improve the quality of services (QoS) of the users. In this paper, we focus on the replication of the contents of the origin server, not the replica servers, since they are located in 5G components introduced as caches.
After locating the caches, content providers decide which contents should be placed in which replica servers. As depicted in Fig. 1, different levels of the network are collaborating hierarchically. Moreover, entities of each level are collaborating among each other. Therefore, the proposed method not only considers inter-level communications but also it brings up intra-level communications.
When using cooperative usage of the levels, a response to the user can go through different caches. HCC prefetches the contents from the nearest cache each time users make requests. If the subsequent requests of the user have already been cached, it serves them directly without prefetching from the server. Therefore, the time for retrieving content will decrease.

A. Content Placement Problem (CPP)

One of the most critical problems in developing a caching mechanism is determining the location of the replica contents. Whenever they are located near to the users, it can be served to the users as fast as possible. On the other hand, because of the limited storage capacity of the caches, a few numbers of users can benefit from them. In contrast, when they are located in the aggregation level and CN, although it can be served to a large number of users, retrieval time will increase. In our system model, we consider a hybrid system of caches where UEs, BS, routers in the aggregation level, and CN are equipped with storage capacities.

When using a hybrid model, a response to the user can go through different caches. Therefore, we determine in which cache, the content should be stored in the reply path to the user. To this aim, a planning model is proposed that decides where to place the portion of the content in the caches with a minimum cost of implementation (CPP). Our optimization model is a function that minimizes the transferring cost of the content. The detailed description of CPP is provided as follows.

B. Mathematical Model

The nomenclature used in this paper is presented in Table I. We hypothesize that all the assignments of BSs to routers in the aggregation level and CN are fixed. Let U, I, J and K be the total number of UEs, BSs in access, routers in aggregation and caches in the CN, respectively. Decision variables y and x indicate whether content m is stored in located caches in related levels or not. If it is cached, the variable is 1, otherwise 0. Besides, the decision variable a indicates whether user u is assigned to BS i or not. The cost for transmitting data between CDN and cache in the CN is shown as T_k (we mention the latency of the network as cost), the cost for transmitting data between aggregation and CN as T_{jk}, the cost for transmitting data between caches in the aggregation level as $T_{jj'}$, the cost for transmitting data between access and aggregation level as T_{ij}, the cost for transmitting data between caches in the access level as $T_{ii'}$, the cost for transmitting data between UEs and access level as T_{ui}, and the cost for transmitting data between caches in UEs are defined as $T_{uu'}$. In our system model, we assume that $T_k > T_{jk} > T_{jj'}$, $T_{ij} > T_{ii'}$ and $T_{ui} > T_{uu'}$.
According to the framework illustrated in Fig. 1, whenever a user requests for content m, at first looks at his local UE cache. If m was there, it could be served directly, without any cost. If not, checks the other nearby UE's cache using device-to-device communication. If so, the cost for retrieving content m will be $T_{uu'}$. If m cannot be found at any UE, the request will send to the access level to the assigned BS. If it holds m, serves it to the user, and the cost will be $(T_{uu'} + T_{ui})$. If not, it looks for other BSs in the access level. If m was in one of nearby BSs, the cost would be $(T_{uu'} + T_{ui} + T_{ii'})$. Again if m cannot be satisfied at any nearby BS, the request goes to the aggregation and CN levels and does the same. At first, an entity checks corresponding cache, then the neighboring caches and calculate the cost of retrieving m. At last, if m cannot be found at any caches in whole tiers, it needs to be fetched from CDN.
We should mention that the video types are not considered in this article, and video sizes are emphasized. According to the mentioned scenario, we first formulate the total transmitting cost of contents to evaluate their complexities. After that, we propose a solution to this problem. We formulate the total transmitting cost of contents as

$$
\begin{aligned}
O = & \sum_{u\in U}\sum_{i\in I}\sum_{j\in J}\sum_{k\in K}\sum_{m\in M}\{\\
& (1-y_u^m)\, min\,\{y_{u'}^m . (T_{uu'})\}\\
& + a_u^i\,(1-x_i^m)\,(T_{uu'}+T_{ui})\\
& + (1-x_i^m)\,(1-y_i^m)\, min\,\{y_{i'}^m\,(T_{uu'}+T_{ui}+T_{ii'})\}\\
& + (1-x_i^m)\left(1-x_j^m\right)(T_{uu'}+T_{ui}+T_{ii'}+T_{ij})\\
& + (1-x_i^m)\left(1-x_j^m\right)\left(1-y_j^m\right)\\
& \quad min\,\{y_{j'}^m\,(T_{uu'}+T_{ui}+T_{ii'}+T_{ij}+T_{jj'})\}\\
& + y_k^m\,(1-x_i^m)\left(1-x_j^m\right)(1-x_k^m)\\
& \quad (T_{uu'}+T_{ui}+T_{ii'}+T_{ij}+T_{jj'}+T_{jk})\\
& + (1-x_i^m)\left(1-x_j^m\right)(1-x_k^m)\,(1-y_k^m)\\
& \quad (T_{uu'}+T_{ui}+T_{ii'}+T_{ij}+T_{jj'}+T_{jk}+T_k)\ \}\,.
\end{aligned} \tag{1}
$$

Our objective is to minimize the latency for the retrieval of the content. Therefore, the optimization problem considering an initial a_u^i assignment is formulated as below to find decision variables,

$$
\begin{aligned}
\underset{a_u^i,\, y_u^m,\, y_i^m,\, y_j^m,\, y_k^m}{\text{minimize}} \quad & O, && && \text{(2a)}\\
\text{subject to} \quad & \sum_{m\in M} y_u^m S_m \le C_u && \forall u \in U, && \text{(2b)}\\
& \sum_{m\in M} y_i^m S_m \le C_i && \forall i \in I, && \text{(2c)}\\
& \sum_{m\in M} y_j^m S_m \le C_j && \forall j \in J, && \text{(2d)}\\
& \sum_{m\in M} y_k^m S_m \le C_k && \forall k \in K, && \text{(2e)}\\
& \sum_{i\in I} a_u^i = 1 && \forall u \in U, && \text{(2f)}\\
& a_u^i, y_u^m, y_i^m, y_j^m, y_k^m \in \{0,1\} && && \text{(2g)}
\end{aligned}
$$

Each of the constraints in (2b) to (2e) states that the total size of the content stored at each cache should not exceed the maximum capacity of the caches in levels, (2f) indicates that each UE can be assigned to just one BS and (2g) shows decision variables. This problem is a binary integer linear program with a search space complexity of $O(2^{U\times I} + 2^{U\times M} + 2^{I\times M} + 2^{J\times M} + 2^{K\times M})$. By using Knapsack problem we can describe our optimization problem where content m is placed in cache levels. Considering each cache as a knapsack with its limited capacity proves that this problem is an NP-hard problem. Since in large-scale problems, finding solutions are not feasible, we divide the proposed problem to subproblems to get practical solutions. As user requests are routed to upper levels when the lower levels cannot satisfy them, we can divide the optimization problem into level based subproblems and solve each subproblem separately. Therefore, the problem O can be written as

$$O = O_{UE} + O_{Access} + O_{aggregate} + O_{CN} + O_{CDN}. \quad (3)$$

such that O_{UE} is formulated as:

$$\sum_{u \in U} \sum_{m \in M} \left(1 - y_u^m\right) min \left\{ y_{u'}^m . \left(T_{uu'}\right) \right\}. \quad (4)$$

O_{Access} is defined as

$$\begin{aligned} &\sum_{u \in U} \sum_{i \in I} \sum_{m \in M} a_u^i \left(1 - x_i^m\right) \left(T_{uu'} + T_{ui}\right) \\ &+ \left(1 - x_i^m\right) \left(1 - y_i^m\right) min \left\{ y_{i'}^m \left(T_{uu'} + T_{ui} + T_{ii'}\right) \right\}. \end{aligned} \quad (5)$$

where x_i^m is defined as follows and can be obtain from O_{UE} .

$$x_i^m = \begin{cases} 1 & \sum_{u \in U} y_u^m > 0 \\ 0 & otherwise. \end{cases} \quad (6)$$

$O_{aggregate}$ is accessed from

$$\begin{aligned} &\sum_{j \in J} \sum_{m \in M} \left(1 - x_i^m\right) \left(1 - x_j^m\right) \left(T_{uu'} + T_{ui} + T_{ii'} + T_{ij}\right) \\ &+ \left(1 - x_i^m\right) \left(1 - x_j^m\right) \left(1 - y_j^m\right) \\ &min \left\{ y_{j'}^m \left(T_{uu'} + T_{ui} + T_{ii'} + T_{ij} + T_{jj'}\right) \right\}. \end{aligned} \quad (7)$$

where x_j^m is defined as follows and can be obtained from O_{Access} .

$$x_j^m = \begin{cases} 1 & \sum_{i \in I} y_i^m > 0 \\ 0 & otherwise. \end{cases} \quad (8)$$

O_{CN} as:

$$\begin{aligned} &\sum_{m \in M} y_k^m \left(1 - x_i^m\right) \left(1 - x_j^m\right) \left(1 - x_k^m\right) \\ &\left(T_{uu'} + T_{ui} + T_{ii'} + T_{ij} + T_{jj'} + T_{jk}\right). \end{aligned} \quad (9)$$

where x_k^m is defined as follows and can be obtained from $O_{aggregate}$.

$$x_k^m = \begin{cases} 1 & \sum_{j \in J} y_j^m > 0 \\ 0 & otherwise. \end{cases} \quad (10)$$

and at last O_{CDN} could be written as:

$$\begin{aligned} &\left(1 - x_i^m\right) \left(1 - x_j^m\right) \left(1 - x_k^m\right) \left(1 - y_k^m\right) \\ &\left(T_{uu'} + T_{ui} + T_{ii'} + T_{ij} + T_{jj'} + T_{jk} + T_k\right). \end{aligned} \quad (11)$$

By solving the CPP, optimal locations of the caches will be found. Therefore, by deploying the replica contents in the mentioned places, the performance of the network will be improved by decreasing the content retrieval time.

III. Experimental Evaluations

In this section, we evaluate the CPP-HCC method in two different simulation environments. As mentioned in the previous sections, the locations of the caches will be found by solving CPP, and then the results can be used as input to the HCC network. CPP is solved in a CPLEX optimization environment. Then, HCC is simulated using OMNeT++ simulator. The parameters of the network are shown in Table I.

According to Table I, the network topology consists of 1000 UEs, three BSs, two routers in the aggregation level, and one CN. The whole number of videos is 10,000, which are randomly requested by users and each video size is between 2 to 9 GB [10] [11]. The latency cost between caches in each level is randomly and uniformly generated, and the cache size is considered as the capacity ratio of the total capacities of the caches to the size of entire contents [9]. Here, the capacity ratio is 50%, and the cache sizes in UEs, BSs, routers, and CN are considered as 10%, 20%, 30% and 40% of the total cache sizes, respectively.

Knowing the location of the contents in our network structure by solving CPP, we define server nodes as caches and client nodes as UEs. Our dataset, including randomized content size, user IDs, and the requested time for each file, was created in MySQL schema. Using two frameworks of INET and SimuLTE in OMNet++ simulator [12] [13], essential communications and links were made between users and eNodeBs in the access level, routers and switches in aggregation and CN level. SimuLTE and INET frameworks provide one-to-many D2D and wireless communications, respectively.

According to Fig. 1, an example scenario was provided here. When a user requests content, if it was a server node, directly got the content (step 1). Else is a client node and sends the request by generating a multicast message at its application level to the neighbors towards a multicast IP address. All neighbor nodes are subscribed to the multicast IP address group. If one of the subscribed UEs performs as a server node, transmits the content to the client UE (step 2). When the content was not found in nearby UEs, the packet will be sent to assigned eNodeB (step 3). Again if the related eNodeB performs as a server, retrieves the content towards download link. If not, sends it to adjacent eNodeBs (step 4). The same tasks were done through steps 5 to 7. Finally, if the content was not in any storage capacity, it will be retrieved from servers in CDN (step 8).

In the rest of this paper, we verify the impact of the different caches sizes in the performance of the CPP-HCC and compare this method with other well-known caching strategies like leave copy everywhere (LCE) and leave copy down (LCD) caching methods [8]. In LCE, retrieved content will be stored in every cache it passes, while in LCD, it will be stored just in the cache that is the direct successor of the cache, which generates a hit.

A. Impact of the cache size in CPP-HCC latency

At first, we verify the five different caching methods including (1) no caching in the network, (2) UE caching, (3) UE

TABLE I
LIST OF SYMBOLS.

Symbol	Explanation	Unit	Value	Ref
u, u'	Index of UEs where $u \neq u'$		1000	
i, i'	Index of BSs in access level where $i \neq i'$		3	
j, j'	Index of routers in aggregation level where $j \neq j'$		2	
k, k'	Index of routers in core network level where $k \neq k'$		1	
m	Index of content from set M		$10,000$	
C_u	Maximum capacity of UEs	GB	10%	[9]
C_i	Maximum capacity of BSs in access level	GB	20%	[9]
C_j	Maximum capacity of caches in aggregation level	GB	30%	[9]
C_k	Maximum capacity of cache in core network	GB	40%	[9]
S_m	Size of content m	GB	$[2, 9]$	[10] [11]
T_k	Cost for transmitting data between CDN and cache in core network	ms	$[150, 200]ms$	[9]
T_{jk}	Cost for transmitting data between aggregation and core network	ms	$[100, 150]ms$	[9]
$T_{jj'}$	Cost for transmitting data between caches in aggregation level	ms	$[80, 100]ms$	[9]
T_{ij}	Cost for transmitting data between access and aggregation level	ms	$[50, 80]ms$	[9]
$T_{ii'}$	Cost for transmitting data between caches in access level	ms	$[30, 50]ms$	[9]
T_{ui}	Cost for transmitting data between UEs and access level	ms	$[10, 30]ms$	[9]
$T_{uu'}$	Cost for transmitting data between caches in UEs	ms	$[5, 10]ms$	[9]
y_u^m	Decision variable showing content m is cached at UE u or not			
y_i^m	Decision variable showing content m is cached at BS i or not			
y_j^m	Decision variable showing content m is cached at cache j in aggregation level or not			
y_k^m	Decision variable showing content m is cached in core network level or not			
x_i^m	Decision variable showing content m is stored at the whole lower level caches in UEs or not			
x_j^m	Decision variable showing content m is stored at the whole lower level caches in a cell or not			
x_k^m	Decision variable showing content m is stored at the whole lower level caches in aggregation or not			
a_u^i	Decision variable showing user u is assigned to BS i or not			

plus BS caching, (4) UE plus BS plus router caching, and (5) UE plus BS plus router plus CN caching. Fig. 2 illustrates the comparison of these five methods. It compares the latency cost of these strategies versus different cache sizes. According to this figure, as we expect, latency is decreased by increasing cache sizes, and the fifth method that includes caches in every level of the network reduces more in comparison to the others. For instance, with no cache in the network (1), the average latency is around 205 ms. The collaborative method (CPP-HCC) that uses caches in every level of the topology (5), reduces the latency by 49% and 83% when the cache sizes are 10% and 30%, respectively.

Secondly, we compare CPP-HCC with LCE and LCD caching in Fig. 3. It is obvious a good policy for decreasing the latency is the one that brings contents closer to the users. In LCD, whenever the location of the requested content is found, the same content will be dropped at the first neighbor of the cache, and LCE drops the content at every caches it passes. As depicted in Fig. 3, LCD method decreases the latency more than LCE, since LCE caches were filled faster than LCD caches with high redundant contents. Besides, CPP-HCC method performs better than LCD. Little redundancy of the contents and inter-level communication make CPP-HCC performs better and decrease latency by 45% and 74% when the cache sizes are 10 and 30 percent.

These results show that CPP-HCC outperforms in reducing latency to the other presented methods because of inter-level and intra-level communications between caches.

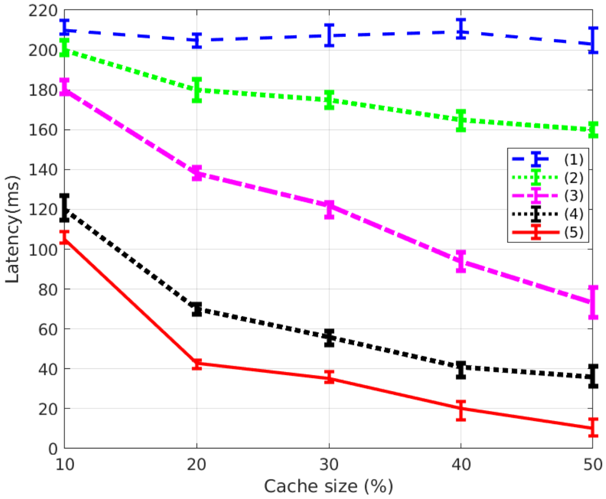

Fig. 2. Latency cost versus cache size.

B. Impact of the cache size in the CPP-HCC hit ratio

Fig. 4 shows the impact of the cache sizes on the hit ratio. Hit ratio is the probability that the requested content will be found in cache locations. As we expected, by increasing the cache sizes, the hit ratio rises for all of the methods. However, CPP-HCC exhibited up a 62% hit ratio whereas LCE has never provided a hit ratio of more than 46%. One of the obstacles that lower the hit ratio, is content redundancy. When the cache size is smaller, LCE makes caches filled faster. Therefore, there will not be enough space for locating new requests and hit

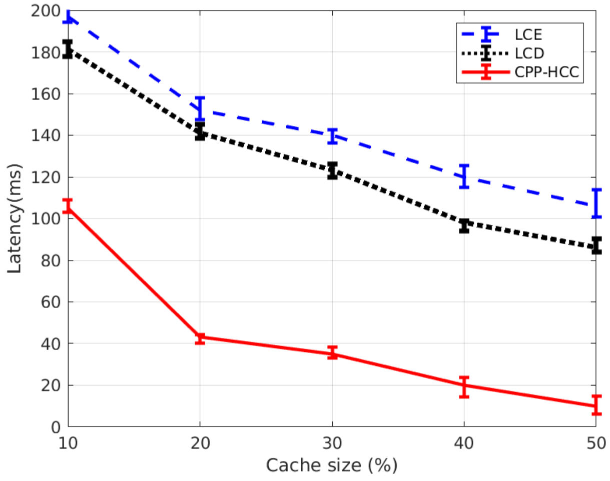

Fig. 3. Latency cost versus cache size.

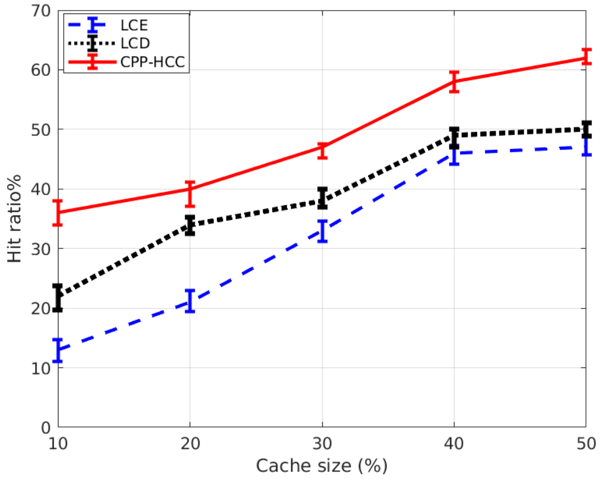

Fig. 4. Hit Ratio versus cache size.

ratio decreases dramatically. However, by increasing the cache sizes, the differences between presented methods will be closer to each other. The reason for that closeness is that in large size caches, contents could be located for a long time without worrying in capacities of the caches.

IV. Related Work

Deployment of the caches and contents have been largely discussed in mobile networks. [14] and [15] provide a comprehensive study of the locations of the caches for a CDN. The authors investigated different placement strategies and categorized them based on their characteristics. A hierarchical collaborative caching was presented in [9]. The authors offered a genetic CPP in access and CN levels to maximize the latency saving of the whole network. [16] proposes a content admission model in a 5G network that is resilient to link cuts by combining the core with edge data centers. CPP in metro networks was formulated in [10]. They claim by powering on and off the cache nodes, energy efficiency will be decreased. [11] investigates the best location of the caches from two different aspects, performance and deployment cost. The results proved if the caches were located in both access and metro segments of the network, there would be a balance between performance and cost. An in-network video caching was presented for Long Term Evolution (LTE) network [17]. The caches are located in evolved packet core (EPC), and the authors provided an optimization problem with the aim of minimizing the aggregate latency. MS caching was introduced in [18]. The caches are placed in SBS and UEs. They designed a computation offloading scheme that makes a balance trade-off between SBS and D2D. [19] provides a hierarchical collaborative caching for LTE networks. They showed if the caches were located in both RAN and EPC, the performance of the network will be increased. The proposal in [20] contains three different caching methods, including caching independently, cooperating in a cluster of MECs, and collaboratively working in the entire network. It reduces energy consumption and average service latency. In [21], MEC servers collaborate to maximize resource utilization, but they waste backhaul bandwidth. The presented articles consider the content caching only in one or two levels of the network (RAN or CN), and ignore the collaboration between levels. Besides, the hierarchical optimization problem in placement methods to find the optimal position of the caches and contents was rarely investigated. Therefore, our work differs by considering the collaboration of caches in different levels of the network and presents a new hierarchical model to deploy the caches in the entities of the 5G network architecture.

V. Conclusion

In this paper, a content placement problem in a hierarchical collaborative caching (CPP-HCC) for 5G networks was presented. First, CPP was verified. We defined an optimization problem to find the optimal locations of the storage entities. This optimization problem minimized the latency for transferring content between inter-level and intra-levels communications. After solving the CPP and locating the caches in a real network model, we evaluate the performance of the network in a hierarchical collaborating manner. The evaluation results confirm the efficiency of CPP-HCC compared to other benchmark methods and show that the latency and hit ratio can be improved by 83% and 62%, respectively. As a future work, we will consider the decision and replacement methods in HCC for 5G networks.

References

[1] Cisco Systems 2017, “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 20162021 White Paper,” *Cisco*, pp. 2016–2021, 2017.

[2] Nokia, “5G Use Cases and Requirements,” *White paper*, pp. 1–16, 2014.

[3] X. Zhou, Z. Zou, R. Song, Y. Wang, and Z. Yu, “Cooperative caching strategies for mobile peer-to-peer networks: a survey,” in *Information Science and Applications (ICISA) 2016*. Springer, 2016, pp. 279–287.

[4] M. Kimmerlin, J. Costa-Requena, and J. Manner, “Caching using software-defined networking in LTE networks,” *2014 IEEE International Conference on Advanced Networks and Telecommunication Systems, ANTS 2014*, pp. 1–6, 2014.

[5] 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 Communications Surveys and Tutorials*, no. c, 2018.

[6] D. Prerna, R. Tekchandani, and N. Kumar, “Device-to-device content caching techniques in 5g: A taxonomy, solutions, and challenges,” *Computer Communications*, vol. 153, pp. 48–84, 2020.

[7] 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 Communications Magazine*, vol. 52, no. 2, pp. 131–139, 2014.

[8] N. Laoutaris, S. Syntila, and I. Stavrakakis, “Meta algorithms for hierarchical web caches,” in *IEEE International Conference on Performance, Computing, and Communications, 2004*. IEEE, 2004, pp. 445–452.

[9] Q. Tang, R. Xie, T. Huang, and Y. Liu, “Hierarchical collaborative caching in 5g networks,” *IET Communications*, vol. 12, no. 18, pp. 2357–2365, 2018.

[10] O. Ayoub, F. Musumeci, M. Tornatore, and A. Pattavina, “Energy-efficient video-on-demand content caching and distribution in metro area networks,” *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 1, pp. 159–169, 2018.

[11] ——, “Techno-economic evaluation of cdn deployments in metropolitan area networks,” *International Conference on Networking and Network Applications (NaNA)*, pp. 314–319, 2017.

[12] G. Nardini, A. Virdis, and G. Stea, “Simulating device-to-device communications in omnet++ with simulte: scenarios and configurations,” *arXiv preprint arXiv:1609.05173*, 2016.

[13] A. Varga, “Omnet++,” in *Modeling and tools for network simulation*. Springer, 2010, pp. 35–59.

[14] J. Sahoo, M. A. Salahuddin, R. Glitho, H. Elbiaze, and W. Ajib, “A survey on replica server placement algorithms for content delivery networks,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1002–1026, 2016.

[15] M. A. Salahuddin, J. Sahoo, R. Glitho, H. Elbiaze, and W. Ajib, “A survey on content placement algorithms for cloud-based content delivery networks,” *IEEE Access*, vol. 6, pp. 91–114, 2017.

[16] C. Natalino, A. de Sousa, L. Wosinska, and M. Furdek, “Content placement in 5g-enabled edge/core datacenter networks resilient to link cut attacks,” *Networks*, 2020.

[17] J. Zhu, J. He, H. Zhou, and B. Zhao, “Epcache: In-network video caching for lte core networks,” in *2013 International Conference on Wireless Communications and Signal Processing*. IEEE, 2013, pp. 1–6.

[18] M. Chen, Y. Hao, M. Qiu, J. Song, D. Wu, and I. Humar, “Mobility-aware caching and computation offloading in 5g ultra-dense cellular networks,” *Sensors*, vol. 16, no. 7, p. 974, 2016.

[19] S. Ren, T. Lin, W. An, Y. Li, Y. Zhang, and Z. Xu, “Collaborative epc and ran caching algorithms for lte mobile networks,” in *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2015, pp. 1–6.

[20] D. Ren, X. Gui, W. Lu, J. An, H. Dai, X. Liang, and I. Engineering, “GHCC: Grouping-Based and Hierarchical Collaborative Caching for Mobile Edge Computing,” pp. 1–6, 2018.

[21] A. Ndikumana, S. Ullah, T. LeAnh, N. H. Tran, and C. S. Hong, “Collaborative cache allocation and computation offloading in mobile edge computing,” *19th Asia-Pacific Network Operations and Management Symposium: Managing a World of Things, APNOMS 2017*, pp. 366–369, 2017.