Data-Driven Computing and Caching in 5G Networks: Architecture and Delay Analysis

Min Chen, Yongfeng Qian, Yixue Hao, Yong Li, and Jeungeun Song

Abstract

Recently, there has been increasing interest of deploying computation-intensive and rich-media applications on mobile devices, and ultra-low latency has become an important requirement to achieve high user QoE. However, conventional mobile communication systems are incapable of providing considerable communication and computation resources to support low latency. Although 5G is expected to effectively increase communication capacity, it is difficult to achieve ultra-low end-to-end delay for the ever growing number of cognitive applications. To address this issue, this article first proposes a novel network architecture using a resource cognitive engine and data engine. The resource cognitive intelligence, based on the learning of network contexts, is aimed at a global view of computing, caching, and communication resources in the network. The data cognitive intelligence, based on data analytics, is critical for the provisioning of personalized and smart services toward specific domains. Then we introduce an optimal caching strategy for the small-cell cloud and the macro-cell cloud. Experimental results demonstrate the effectiveness of the proposed caching strategy, and its latency is lower than that of the two conventional approaches, that is, the popular caching strategy and the greedy caching strategy.

Introduction

Ultra-low latency and high reliability represent the most important factors for improving the user experience in communication systems. Although fifth generation (5G) mobile networks effectively enhance the performance of mobile networks, it is still a major challenge to reduce the latency of network services. This is due to the increasing requirements of the novel mobile applications and services on timely content delivery or real-time user interaction [1]. To overcome this issue, various caching techniques have been proposed, and it is proven that caching is an effective approach for improving users' quality of experience (QoE) [2].

Considering the computing resource of the edge cloud, recently edge computing, fog clouds, cloudlets, and mobile clouds are utilized for offloading computation and/or content at the network edges to lower latency [3]. For instance, an effective offloading strategy in the edge cloud for massive users is reported in Chen *et al.* [4], which significantly lowers the computation latency and energy consumption. Zhang *et al.* [5] discussed predictive offloading in the edge cloud. In this article, based on the deployment's location in a 5G network, edge clouds are classified into two categories: small-cell cloud (or fog cloud, mobile cloud, or cloudlet) and macrocell cloud (or edge cloud).

The small cell cloud is formed by the storage and computing resources deployed in small cells. Due to the limited hardware resources, the small cell cloud may not provide adequate computational and caching services. However, users can directly access the small cell cloud via a one-hop wireless channel, and therefore, the latency in terms of communication is often low [6]. The macrocell cloud contains the storage and computing resources deployed in the macro base station. Thus, its storage and computation capacity is higher than that of small cell cloud. Although the macrocell cloud provides considerable caching and computational services and guarantees shorter latency of data processing, it still suffers from limited storage and computing resources when handling complex tasks [7].

On the other hand, due to the intrinsic network dynamics, it is difficult to accurately recognize and control all the available resources on the network edges. Fortunately, assisted by software defined networking (SDN), the control layer is separated from the infrastructure layer. Based on the information collected in the infrastructure layer, the control layer is able to provide resource scheduling with a global view [8].

With the incorporation of the caching in the macrocell and small-cell clouds, this article introduces an innovative cognitive SDN architecture for further lowering the latency of domain-oriented applications. The main cognitive intelligence of the proposed architecture is the integration of a data cognitive engine and a resource cognitive engine. As shown in Fig. 1, for a user-centric healthcare service, we assume that there are three users, Alice, Bob, and Eva, within the coverage of 5G networks. The body signals of each user are collected by wearable devices (e.g., smart clothing). Then the data are sent to edge clouds or the core network for cloud analytics [9]. Without the use of a data cognitive engine, the base station cannot know the priority of healthcare users, and will give each user an average resource allocation in terms of computing, caching, and communications. Once a user with a chronic disease suffers from an emerging situation, doctors, a medical center, and/or immediate family members need to contact the user with

Digital Object Identifier:
10.1109/MWC.2018.1700216

Min Chen, Yongfeng Qian (corresponding author), Yixue Hao, and Jeungeun Song are with Huazhong University of Science and Technology; Yong Li is with with Tsinghua University.

1536-1284/18/$25.00 © 2018 IEEE

ultra-low latency. If the critical moment to handle the medical emergency passes, his/her life might be endangered. However, it is hard for the base station to know this user-centric situation. Especially when massive numbers of mobile devices are covered by the base station, the situation would deteriorate with a large transmission delay.

With the introduction of a data cognitive engine, the healthcare big data cloud produces disease diagnosis results based on a user's healthcare data and medical records. Given Fig. 1 as an example, three categories of users are classified in the cloud, that is, a high risk user with the highest priority, a low risk user with medium priority, and a healthy user with the lowest priority. In this example, Alice has a high disease risk, so the data engine notifies the SDN controller to assign resources to her with the highest priority. That is, the SDN controller will evoke the resource cognitive engine to secure differentiated and high-quality services to Alice. Thus, through the joint usage of these two cognitive engines, we can optimize the allocation of resources to achieve ultra-low latency and differentiated services with a certain constraint on computing, storage, and bandwidth. In order to further decrease the service delay, an optimal caching strategy is also deployed in the cognitive SDN engine with the cooperative caching design on small-cell clouds and the macrocell cloud. In summary, the main contributions of this article are listed as follows.

We propose a novel SDN architecture in 5G networks, which includes a resource cognitive engine and a data cognitive engine. The resource cognitive engine achieves the required ultra-low latency by situation cognition and optimally allocating the resources in the network. The data cognitive engine achieves the required intelligence by analyzing domain-oriented big data.

Based on the proposed architecture, we implement caching latency analysis and present the optimal caching strategy at small-cell clouds and the macrocell cloud. The presented caching policy effectively reduces end-to-end latency for delay-sensitive applications. Experimental results show that the proposed optimal caching strategy exhibits better performance in terms of end-to-end latency compared to traditional caching policies.

The rest of the article is organized as follows. The following section presents various ultra-low-latency scenarios. Following that, the proposed SDN architecture is introduced in detail. Then we provide the latency analysis for the small-cell clouds and macrocell cloud in 5G networks. Then the experimental results and discussion of them are given. Finally, we conclude this article.

Ultra-Low-Latency Scenarios in 5G

The ultra-low-latency services in 5G technology can be either push-based or request-based [10]. The push-based services are supported by the content provider [11]. In contrast, the request-based services are initiated by the users [12]. Generally, push-based service corresponds to content caching service, while request-based service is more personalized with specific task descriptions.

Push-based services include the following three representative categories, that is, emotion-aware computing, cognitive mobile gaming, and content delivery-oriented augumented reality (AR) gaming. Emotion-aware computing recognizes the user's emotions through sensory data, such as the facial expression and heart rate, and then provides emotional care services to the user. Emotion-aware computing requires real-time emotion detection, and therefore, low latency is essential to reach the requirement of a user's QoE. Cognitive mobile gaming analyzes the user's context in real time and pushes the corresponding gaming contents. Hence, ultra-low-latency transmission is indispensable to ensure that the pushing service meets the user's demand. Content-delivery-based AR gaming is more time-sensitive, and hence, outdated pushing is not acceptable and brings ill effects for the content provider to pursue high profits through content provisioning due to the unpleasant user's QoE.

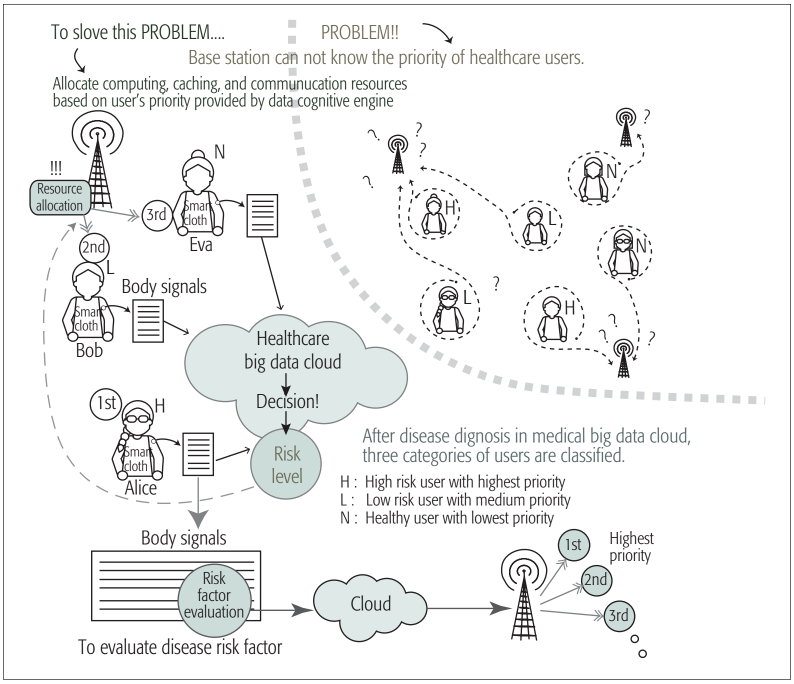
To slove this PROBLEM....
PROBLEM!!
Base station can not know the priority of healthcare users.
Allocate computing, caching, and communication resources based on user's priority provided by data cognitive engine
!!!
Resource allocation
3rd
Smart cloth
N
Eva
2nd
L
Smart cloth
Bob
Body signals
Healthcare big data cloud
Decision!
1st
H
Smart cloth
Alice
Risk level
After disease dignosis in medical big data cloud, three categories of users are classified.
H : High risk user with highest priority
L : Low risk user with medium priority
N : Healthy user with lowest priority
H
L
N
N
L
H
?
Body signals
Risk factor evaluation
To evaluate disease risk factor
Cloud
1st
2nd
3rd
Highest priority

FIGURE 1. Your life is our priority: towards high quality of user-centric healthcare services powered by data cognitive engine.

The request-based service emphasizes the interactions between users and networks. It can be classified into five categories: the tactile internet, remote surgery, mobile healthcare, assisted driving and transport services, and industry automation. In the tactile Internet, the user's tactile information is sensed and analyzed for providing the corresponding services to the user. In remote surgery, the patient's health and safety can be ensured subject to low latency and highly reliable communications. Mobile healthcare aims to collect the user's physiological information and provide health analysis. For the patient with the higher disease risk factor, the mobile healthcare system should provide medical services prior to other users. Assisted driving and transport services can provide timely and valid traffic information. However, it requires ultra-low latency, especially for high-speed vehicles. Industrial automation also requires ultra-low latency to meet the requirements of large-scale industrial manufacturing and provide smart industry.

There are relations between content caching and computation offloading. Taking video transcoding [13] as an example, the video transcoding usually corresponds to a kind of computing service. However, when the transcoding result of the video has already been cached in a small-cell cloud, it can be transformed into a kind of caching service.

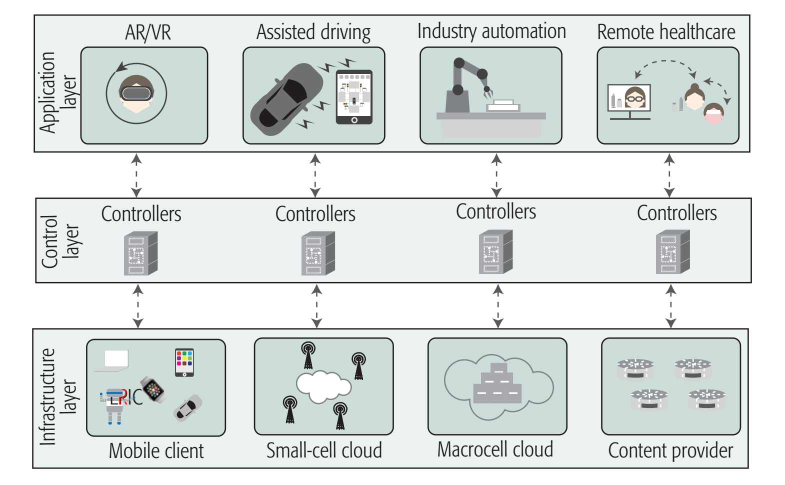

FIGURE 2. SDN-based hierarchical network architecture.

Since caching for such computing service can also reduce latency, in this article, we mainly focus on the caching problem.

Architecture of SDN-Based Mobile Network with Resource and Data Cognitive Engines

The requirements of the emerging mobile services and applications represented by AR/virtual reality (VR) on communication latency are significantly high. This article introduces a network architecture based on SDN that reduces the latency of the mobile network, where the caching technology is utilized to effectively reduce the latency.

SDN-Based Hierarchical Network

We present a network architecture based on SDN that meets the low latency requirement of the 5G network. It effectively employs the storage and computing resources on the network edge to reduce the latency. Furthermore, the presented SDN conducts resource and data cognition for the small-cell cloud and macrocell cloud. Figure 2 illustrates the SDN-based low-latency mobile network architecture, which is formed by three layers: the physical, control, and application layers.

The infrastructure layer is formed by the user terminals (including the sensors, cognitive devices, and mobile devices), wireless access network, small-cell cloud, macrocell cloud, core network, and remote cloud. A user terminal is the requester of the services, whereas the wireless access network and the core network are responsible for the communication access and transmission. The small-cell cloud and macrocell cloud provide caching for appropriate contents to meet the user's requirements. The caching strategy determines what contents to cache and where the contents are cached. The remote cloud represents the database of the content provider, and is responsible for storage or computation of the data and provides services to the user.

The control layer consists of the main controller and the subordinate controller. The subordinate controller hands out the control instructions separately to the user terminal, wireless access network, small-cell cloud, macro-cell cloud, core network, and remote cloud. Moreover, it receives infrastructure information fed back by the bottom layer and collects the data. The main controller is responsible for coordinating and allocating the resources throughout the whole network. It also transmits the instructions to the subordinate controller and receives the resources information at the bottom layer in real time to realize more globally optimized scheduling.

The application layer is formed by various user applications, for example, the tactile Internet, remote surgery, emotion-aware computing, and cognitive cloud gaming. After a user chooses an application, the main controller converts the user requirements into instructions and transmits them to the subordinate controllers at all levels. This conducts the corresponding deployment and resource allocation for the bottom layer.

Resource and Data Cognitive Engines

To sum up, meeting the requirements of the mobile applications and services with low end-to-end latency requires unified allocation and treatment of the communication, storage, and computing resources. This yields a comprehensive arrangement of the resources. For instance, reducing the end-to-end latency of the video content and the reduction of fronthaul traffic load can be achieved by arranging the popular contents at the location as close as possible to terminal user devices.

Based on our framework, in order to meet the requirements for ultra-low latency and improve the resource management, we propose the resource cognitive engine and the data cognitive engine, as shown in Fig. 3. The resource cognitive engine can achieve resource cognition (e.g., the dynamic storage and computing resources in the small-cell cloud or macrocell cloud), based on the learning of network contexts, and then achieve ultra-low latency and energy efficiency in 5G networks. The data cognitive engine can achieve cognition of big data, and then realize intelligence associated with the user requirements (i.e., the caching and computing problems) .Through the resource cognitive engine and data cognitive engine, the network resource can be flexibly and efficiently divided, thus achieving ultra-low latency.

In this article, in order to allocate resources more rationally, we deploy a hierarchical SDN, that is, the management on the data at the small-cell cloud level is mainly provided by the controller at the small-cell level, and centralized control in the macrocell cloud is mainly achieved by the controller at the macrocell cloud level. The controller on the remote cloud is responsible for intensively managing data on the remote cloud, optimizing and utilizing the global information. In the next section, we analyze how to implement content caching based on the architecture above to reduce latency further.

Caching-Based Latency Optimization Strategy

To reach the minimum latency, this section considers caching using the small-cell cloud and the macrocell cloud, which reduces the response time after the user requests the service. Figure 4 depicts delay components under the caching architecture. If the service content requested by a user is cached on the small-cell cloud, the small-cell cloud will transmit that content to the user, that is, feedback content cached at the small-cell cloud (SC-feedback). Here, $T_{SC\text{-}U}$ represents the latency when the small-cell cloud transmits the content to that user, where the latency in this case is denoted by $T = T_{SC\text{-}U}$. If the content required by a user has not cached on the small-cell cloud, the request can be sent to the

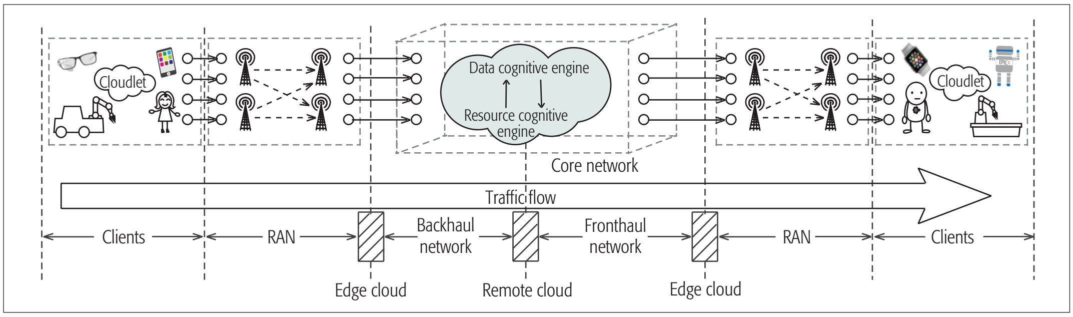

FIGURE 3. Resource cognitive engine and data cognitive engine.

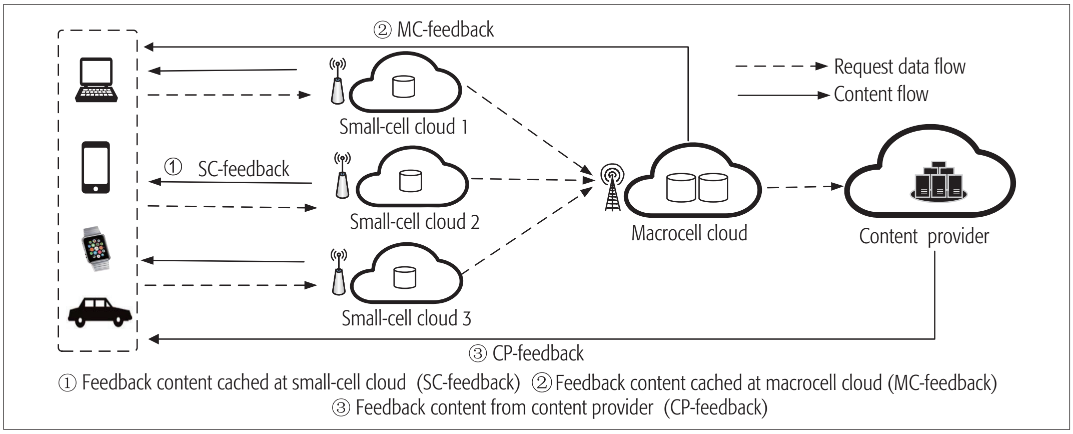
② MC-feedback
Request data flow
Content flow
Small-cell cloud 1
① SC-feedback
Small-cell cloud 2
Macrocell cloud
Content provider
Small-cell cloud 3
③ CP-feedback
① Feedback content cached at small-cell cloud (SC-feedback) ② Feedback content cached at macrocell cloud (MC-feedback)
③ Feedback content from content provider (CP-feedback)

FIGURE 4. Illustration of the delay of the caching design.

The data cognitive engine can achieve cognition of big data, and then realize intelligence of the user requirement (i.e., caching problem and computing problem) .Through resource cognitive engine and data cognitive engine, the network resource can be flexibly and efficiently divided, thus achieved ultra-low latency.

macrocell cloud. Meanwhile, if the request can be fulfilled by the macrocell cloud at the moment, the macrocell cloud is able to transmit that content to that user, that is, feedback content cached at the macrocell cloud (MC-feedback). We consider $T_{MC\text{-}SC}$ as the latency when the macrocell cloud transmits the content to the small-cell cloud, and the latency when the macrocell cloud transmits the content to that user is equal to $T = T_{MC\text{-}SC} + T_{SC\text{-}U}$. However, if the content required by that user is not cached on the macrocell cloud, the request should be further sent to the content provider. The content provider will transmit that content to the user all the way going through edge clouds, that is, feedback content from the content provider (CP-feedback). The latency from the content provider to the macrocell cloud is denoted by $T_{CP\text{-}MC}$, where the latency of the content provider to the users can be denoted by $T = T_{CP\text{-}MC} + T_{MC\text{-}SC} + T_{SC\text{-}U}$.

Table 1 lists the latency, reliability, mobility, and SDN compatibility for various applications. Therefore, in the case of the new type of network architecture, reducing the latency results in the minimum value of T. The next section provides the analysis of the caching and latency.

Latency Analysis of the Co-Located Small-Cell Clouds and Macrocell Cloud in 5G Networks

Optimization Problem

This section investigates the provisioning of caching on the small-cell cloud and macrocell cloud, with the purpose of minimizing the latency at which the user receives content. Given the scenario shown in Fig. 4, n small-cell clouds are connected to the macrocell cloud. We assume a limited caching capacity on the small-cell cloud and the macrocell cloud, where the caching capacity of n small-cell clouds is denoted by C_i, for $i = 1, \cdots, n$, with C_0 being the caching capacity of the macrocell cloud. We assume that m files in the folder are requested, and then define the set of these m files as $F = \{F_1, F_2, \ldots, F_m\}$. Therein, files are sequenced by popularity, that is, F_1 representing the most popular content and F_m being the least popular content. The popularity of each document can be obtained through the analysis of data by the small-cell cloud, and therefore we can assume that the popularity of the file is knowable.

As aforementioned, when a user requests a file in a cell, the corresponding content cached on the small-cell cloud connected to the user is first checked by the system. If the content is cached on the small-cell cloud connected to that user, the content can be directly sent to the user who requests the content. If the content is not cached on the small-cell cloud, the system will check whether the content is cached on the macrocell cloud. If the content is cached on the macrocell cloud, the content can be transmitted to the user requesting the content. If the content is not cached on the small-cell cloud or the macrocell cloud, the content will be retrieved from the content provider through the return link.

In the proposed scheme, the file popularity is different in distinct areas, and in order to establish a model for the caching strategy, we define content placement $\mathbf{x} = (x_{i,f} \in \{0,1\}, i = 1, 2, \cdots, n, f = 1, 2, \ldots, m)$. $x_{i,f} = 1$ shows that file F_f is cached on the small-cell cloud, and $x_{i,f} = 0$ exhibits that file F_f is not cached on the small-cell cloud. Moreover, $x_{0,f} = 1$ represents that the document is cached on the macrocell cloud, and $x_{0,f} = 0$ reveals that file F_f is not cached on the macrocell cloud.

In order to make the minimal average end-to-

Application category	Main delay category	Reliability	Mobility	SDN
Tactile Internet	$T_{SC\text{-}U}$, $T_{MC\text{-}SC}$, T_{CPMC}	High	Low	Medium
Remote surgery	$T_{SC\text{-}U}$, $T_{MC\text{-}SC}$, T_{CPMC}	Ultra high	Low	High
Emotion-aware computing	$T_{SC\text{-}U}$ or $T_{SC\text{-}U}$, $T_{MC\text{-}SC}$	High	Medium	High
Cognitive mobile gaming	$T_{SC\text{-}U}$ or $T_{SC\text{-}U}$, $T_{MC\text{-}SC}$	High	Medium	High
Mobile health	$T_{SC\text{-}U}$, $T_{MC\text{-}SC}$, T_{CPMC}	Ultra high	High	High
Assisted driving and transport services	$T_{SC\text{-}U}$, $T_{MC\text{-}SC}$, T_{CPMC}	Ultra high	High	High
Content delivery and gaming	$T_{SC\text{-}U}$ or $T_{SC\text{-}U}$, $T_{MC\text{-}SC}$	High	Medium	Medium
Industry automation	$T_{SC\text{-}U}$, $T_{MC\text{-}SC}$, T_{CPMC}	Ultra high	High	High

TABLE 1. Different delay for applications.

end latency, we consider the latency of different paths, that is, the latency from the small cell to the user, and also the extra fronthaul transmission latency caused if content F_f is not cached on the small-cell cloud or the macrocell cloud. Moreover, we define $T_{MC\text{-}SC}$ as the latency from the macrocell cloud to the *i*th small-cell cloud, where $T_{SC_i\text{-}U}$ indicates the latency when the *i*th small-cell cloud transmits the content to a user, and $T_{CP\text{-}MC}$ exhibits the latency from a content provider to the macrocell cloud. We then define $\lambda_{i,f}$ as the request probability of a document in these *n* cells, with *i* = 1, 2, ..., *n* and *f* = 1, 2, ..., *m*. Then we achieve the delay of the *i*th small-cell cloud as follows:

$$T_i = \sum_{f \in \mathcal{F}} \lambda_{i,f}\Big[(1-x_{o,f})(1-x_{i,f})T_{CP-MC} \\ +(1-x_{i,f})(T_{MC-SC_i} + T_{SC_i-U} + x_{i,f}T_{SC_i-U}\Big] \quad (1)$$

Thus, the optimization caching problem could be obtained as

$$\min_{\mathbf{x}} \max_{\mathbf{i} \in \{\mathbf{1},\mathbf{2},\cdots,\mathbf{n}\}} T_i \\ \text{subject to: } \sum_{f \in \mathcal{F}} x_{i,f} \le C_i \quad (2)$$

In general, the caching problem of the content is an NP-hard problem. Here, we transform the above optimization problem into a sub-modular optimization problem with a monotonic objective function [14], and the constraint condition is a matroid limitation. An approximate optimal solution can be achieved using a greedy algorithm.

Performance Evaluation

Next, we investigate the performance of the caching strategy by comparing the results provided by the proposed optimal caching strategy, the popular caching strategy, and the greedy caching strategy. In the popular caching strategy, the most popular contents are cached on the small-cell cloud and the macrocell cloud. However, in the greedy caching strategy, a minimal overall latency can be achieved as follows. Both the small-cell cloud and the macrocell cloud are initially empty, and the files are added one by one until reaching the caching capacity of the small-cell cloud and the macrocell cloud.

In the experimental setup, we use a network topology that consists of a macrocell cloud and nine small-cell clouds. For the delay analysis, we assume identical access latency for the cells, and consider $T_{MC\text{-}U}$ = 30 ms. Furthermore, we set the latency from the small-cell cloud to the macrocell cloud to $T_{MC\text{-}SC}$ = 20 ms and the latency from the content provider to the macrocell cloud to $T_{CP\text{-}MC}$ = 50 ms. For the file library, we assume that the file popularity follows the Zipf distribution [15], and the small-cell clouds acquire different content popularity. The file library contains 500 files; the macrocell cloud can store up to 20 percent of the file library, and the small-cell cloud can store up to 10 percent of the file library.

Figures 5a and 5b plot the experimental results for the performance evaluation of the caching in 5G. When the small-cell cloud and the macrocell cloud have large caching capacity, we achieve low latency when the file is requested. This is due to the fact that for a large caching capacity, several files can be cached, and thus, it is easy for a user to obtain a required content. Thus, it results in reducing the latency when the content is requested. From Figs. 5a and 5b, we can also see that the macrocell cloud significantly affects the latency. This is because the macrocell cloud can affect the popularity of whole files in the small-cell clouds. Figure 5c shows the impact of the Zipf parameter on the average delay. We observe that increasing the popularity yields decreased delay.

Then we analyze the relationship between the proposed optimal caching strategy and the proportion of the popular file, which describes specific files cached in the small-cell cloud and the macrocell cloud. It can be seen from Fig. 5d that not all the small-cell clouds cache the popular contents. For example, 60 percent of the cached popular contents in the first small-cell cloud are stored, and 70 percent of the cached popular contents in the fourth small-cell cloud are ready for reuse. However, 90 percent of the cached popular contents are stored in the seventh small-cell cloud.

Conclusion

We propose an effective approach to collect global information about available resources through an SDN-based mobile network architecture. We design an optimal caching strategy consisting of a small-cell cloud and a macrocell cloud to minimize network latency. Experimental results show that the proposed scheme exhibits significantly lower latency than the conventional caching strategies, such as the popular caching strategy and the greedy caching strategy. Our future work will mainly focus on the data analysis and services reliability of the proposed network architecture.

Acknowledgments

This work is supported by the Applied Basic Research Program funded by the Wuhan Science and Technology Bureau (Grant No. 2017010201010118), the Hubei Provincial Key Project under grant 2017CFA061, and the National Natural Science Foundation of China (Grant No. 61572220). Yongfeng Qian is the corresponding author.

References

[1] J. G. Andrews *et al.*, "What Will 5G Be?" *IEEE JSAC*, vol. 32, no. 6, 2014, pp. 1065–82.
[2] D. Liu *et al.*, "Caching at the Wireless Edge: Design Aspects, Challenges, and Future Directions," *IEEE Commun. Mag.*, vol. 54, no. 9, Sept. 2016, pp. 22–28.

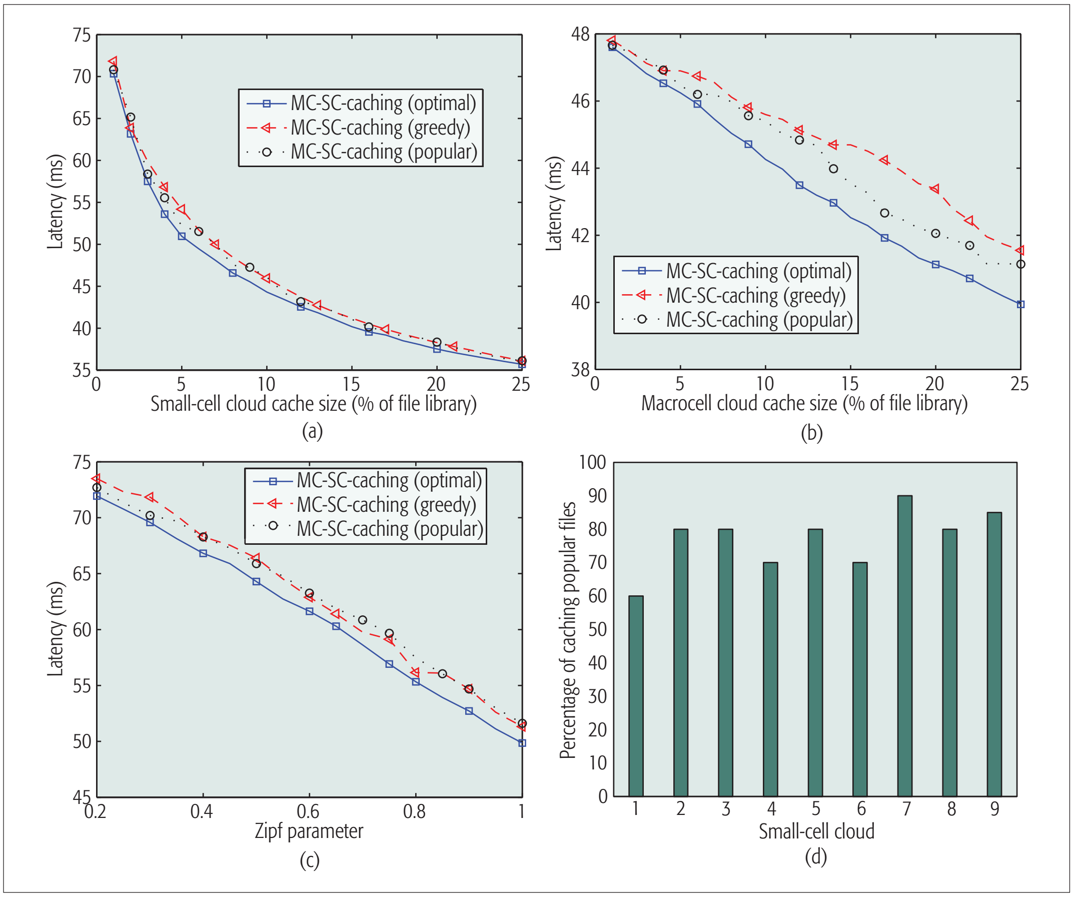

FIGURE 5. Experimental results for the performance evaluation of the aching in 5G: a) effect of the small-cell cloud cache size; b) effect of the macrocell cloud cache size; c) effect of the Zipf parameter; d) percentage of the popular content cached in different small-cell clouds.

Experimental results show that the proposed scheme exhibits a significantly lower latency than the conventional caching strategies, for example, the popular caching strategy and the greedy caching strategy. Our future work will mainly focus on the data analysis and services reliability of the proposed network architecture.

[3] S. Wang *et al.*, "A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications," *IEEE Access*, vol. 5, 2017, pp. 6757–79.

[4] X. Chen *et al.*, "Efficient Multi-User Computation Offloading for Mobile-edge Cloud Computing," *IEEE/ACM Trans. Net.*, vol. 24, no. 5, 2016, pp. 2795–2808.

[5] K. Zhang *et al.*, "Mobile Edge Computing for Vehicular Networks: A Promising Network Paradigm with Predictive Offloading", *IEEE Vehic. Tech. Mag.*, vol. 12, no. 2, June 2017, pp. 36–44.

[6] M. Chen *et al.*, "Mobility-Aware Caching and Computation Offloading in 5G Ultradense Cellular Networks," *Sensors*, vol. 16, no. 7, 2016, pp. 974–87.

[7] K. Zhang *et al.*, "Energy-Efficient Offloading for Mobile Edge Computing in 5G Heterogeneous Networks," *IEEE Access*, vol.4, 2016, pp. 5896–5907.

[8] M. Chen et al., "Green and Mobility-aware Caching in 5G Networks," *IEEE Trans. Wireless Commun.*, vol.16, no. 12, 2017, pp. 8347–61.

[9] M. Chen *et al.*, "Disease Prediction by Machine Learning over Big Healthcare Data," *IEEE Access*, vol. 5, June 2017, pp. 8869–79.

[10] M. Simsek *et al.*, "5G-Enabled Tactile Internet," *IEEE JSAC*, vol. 34, no. 3, 2016, pp. 460–73.

[11] L. Zhou, "Mobile Device-to-Device Video Distribution: Theory and Application," *ACM Trans. Multimedia Comp. Commun. Appl.*, vol. 12, no. 3, 2015, pp. 1253–71.

[12] S. Maharjan, Y. Zhang, and S. Gjessing, "Optimal Incentive Design for Cloud-Enabled Multimedia Crowdsourcing," *IEEE Trans. Multimedia*, vol. 18, no. 12, 2016, pp. 2470–81.

[13] L. Zhou, "QoE-Driven Delay Announcement for Cloud Mobile Media," *IEEE Trans. Circuits Sys. Video Tech.*, vol. 27, no.1, 2017, pp. 84–94.

[14] K. Poularakis and L. Tassiulas, "On the Complexity of Optimal Content Placement in Hierarchical Caching Networks" *IEEE Trans. Commun.*, vol. 64, no. 5, 2016, pp. 2092–2103.

[15] D. Liu and C. Yang, "Energy Efficiency of Downlink Networks with Caching at Base Stations," *IEEE JSAC*, vol. 34, no. 4, 2016, pp. 907–22.

References

Min Chen [SM'09] (minchen2012@hust.edu.cn) has been a full professor in the School of Computer Science and Technology at Huazhong University of Science and Technology (HUST) since February 2012. He is Chair of the IEEE Computer Society STC on big data. His Google Scholars Citations reached 11,000+ with an h-index of 53. He received the IEEE Communications Society Fred W. Ellersick Prize in 2017. His research focuses on cyber physical systems, IoT sensing, 5G networks, SDN, healthcare big data, and so on.

Yongfeng Qian (yongfeng@hust.edu.cn) received her M.S. degree from HUST in 2015. She is currently a Ph.D. candidate in the Embedded and Pervasive Computing Lab led by Prof. Min Chen in the School of Computer Science and Technology at HUST. Her research includes network security, data privacy, the Internet of Things, big data analytics, deep learning, mobile cloud computing, and healthcare.

Yixue Hao (yixuehao@hust.edu.cn) received his B.E. degree from Henan University, China, and his Ph.D degree in computer science from HUST in 2017. He is currently working as a postdoctoral scholar in the School of Computer Science and Technology at HUST. His research includes the 5G network, Internet of Things, and mobile cloud computing.

Yong Li [M'09, SM'16] (liyong07@tsinghua.edu.cn) received his B.S. degree in electronics and information engineering from HUST in 2007 and his Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China, in 2012. During July 2012 to August 2013, he was a visiting research associate with Telekom Innovation Laboratories and the Hong Kong University of Science and Technology, respectively. During December 2013 to March 2014, he was a visiting scientist at the University of Miami. He is currently a faculty member in the Department of Electronic Engineering, Tsinghua University. His research interests are in the areas of networking and communications.

Jeungeun Song (jsong@hust.edu.cn) is a Ph.D candidate in the School of Computer Science and Technology, HUST. Her research focuses on the Internet of Things, mobile cloud, edge computing, emotion-aware computing, healthcare big data, cyber physical systems, robotics, and so on.