ECCN: Orchestration of Edge-Centric Computing and Content-Centric Networking in the 5G Radio Access Network

メタデータ	言語: English 出版者: IEEE 公開日: 2019-06-26 キーワード (Ja): キーワード (En): Protocols, Servers, Streaming media, 5G mobile communication, Mobile handsets, Base stations, Cloud computing 作成者: 李, 鶴, 太田, 香, 董, 冕雄 メールアドレス: 所属:
URL	http://hdl.handle.net/10258/00009915

ECCN: Orchestration of Edge Centric Computing and Content Centric Networking in 5G Radio Access Network

He Li, *Member, IEEE,* Kaoru Ota, *Member, IEEE,* Mianxiong Dong, *Member, IEEE*

***Abstract*—Edge centric computing (ECC) and content centric networking (CCN) will be the most important technologies in future 5G networks. However, due to different architectures and protocols, it is still a challenge to fuse ECC and CCN together and provide manageable and flexible services. In this article, we present ECCN, an orchestrating scheme that integrates ECC and CCN into a hierarchical structure with software defined networking (SDN). We introduce the SDN technology into the hierarchical structure to decouple data and control planes of ECC and CCN and then design an SDN protocol to control the data forwarding. We also implement two demonstration applications in our testbed to evaluate ECCN scheme. The experimental results from the testbed applications and extensive simulations show the integration outperforms original structures.**

***Index Terms*—Content Centric Networking (CCN), Edge Computing, Software Defined Networking (SDN)**

I. INTRODUCTION

Content centric networking (CCN) is an emerging technology for distributing contents to users, which makes content directly addressable and routable in networks [1]. In 5G networks, the content distribution will be a very important issue and bring challenges to the current network infrastructure. A CCN-based radio access network (RAN) will improve the efficiency of content distribution in 5G networks [2]. Edge centric computing (ECC) is another novel paradigm that moves the locus of control of cloud services to the network edge devices [3]. ECC allows users to control their information themselves, which leverages resources and reduces the response time of online services.

The communication between cloud servers and end users is still a challenge in mobile computing due to the limited performance of wireless networks and RANs [4]. Both ECC and CCN can improve the communication performance by reducing the distance between users and services. Thus, the fusion of ECC and CCN will be a great opportunity for 5G network. With the fusion of ECC and CCN, service providers can keep content and computing resources much closer to the end users, which reduces the communication overhead between end users and cloud servers.

However, it is difficult to integrate ECC and CCN together because of two different protocols and architectures. Most ECC systems are developed in IPv4 network environments while CCN needs specific protocols in the network layer [5]. ECC focuses on the edge of networks, which needs more computing resources in the edge devices. CCN is a new routing paradigm, in which the routing devices have more capacity for caching contents. Therefore, it is a big challenge to organically combine these two heterogeneous technologies together.

For integrating ECC and CCN, software defined networking (SDN) provides an opportunity because of the programmable and flexible architecture [6]. Most SDN protocols, such as OpenFlow, are compatible with different networks and devices. From the literature review on CCN, it is possible to build an SDN-based CCN routing scheme. Moreover, SDN is also a promising technology for network forwarding in ECC environments.

Another opportunity is the possible connection between ECC and CCN in mobile computing. In mobile ECC environments, most edge servers are deployed near to the base station, while in CCN, the content cache services are deployed in routing devices from the core network to the base station. Thus, it is possible to deploy independent edge servers between the core network and base stations for the integration of ECC and CCN.

Therefore, in this paper, we first introduce the integration of ECC and CCN and then proposed a heterogeneous RAN structure to support ECC and CCN. We decouple the data forwarding and network control of ECC and CCN, and design a specific SDN protocol to control the data forwarding. We implement two demonstration applications in the orchestrating testbed for performance evaluation. We compare the performance of our scheme and the original network in both demonstration applications and extensive simulations.

The main contributions of this paper are summarized as follows.

- We first propose an orchestrating scheme that integrates ECC and CCN together in a 5G mobile environment. To the best of our knowledge, this is the first work to fuse ECC and CCN in the mobile environment.
- We then design a new SDN protocol for data forwarding in ECCN to support the management of both ECC and CCN. The network control with the proposed SDN protocol is also proposed in this article.
- We implement the demonstration of the integrating network with two different network application. We also take extensive simulations to evaluate the performance of ECCN in the large-scale network environment.

The remainder of this paper can be outlined as follows. Section II introduces ECC and CCN in mobile computing.

He Li, Kaoru Ota and Mianxiong Dong are with Muroran Insitute of Technology, Japan.

Section III introduces the scenario of the integration of ECC and CCN. Section IV describes the framework structure and SDN protocol. Section V presents the demonstration implementation and performance evaluation, followed by the conclusions drawn in Section VI.

II. RELATED WORK

In this section, we introduce some important related works for building the integrating ECCN scheme.

Edge computing is usually adopted for offloading overload from the cloud server to edge devices. Since edge devices have enough performance for executing complex computational tasks, there have been some works focused on computation offloading for edge computing [7]. From the experimental results of these computation offloading methods, edge computing has good efficiency for providing cloud services.

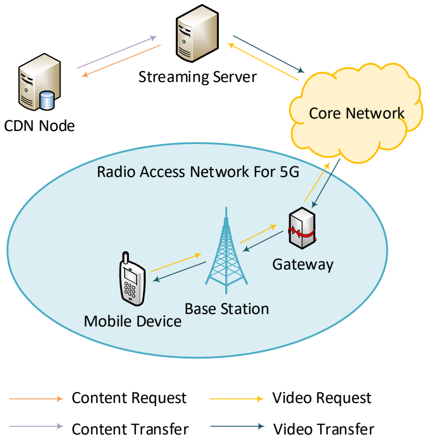
ECC is a new paradigm in which edge devices provide services directly while cloud computing plays an assistant role in the service provision. ECC is first proposed by Predro et al. [3], which encompasses several important elements including proximity, intelligence, trust, control, and human-centric design in the edge. Unlike traditional edge computing focusing on offloading tasks from cloud servers, ECC services are mainly provided in edge servers while the cloud server plays a secondary role.

CCN is another important technology for content delivery services. Contents in CCN are treated as both the contents themselves and unique names. In most CCN protocols, network forwarding is driven by names. When an end user needs a content from the network, an interest containing the name of the required content is sent to CCN nodes. If a CCN node has stored the required content after retrieving the name in its cache, the content is sent to the end user. All CCN nodes in the forwarding path will cache the content for possible requests in the feature. CCN brings advantages on availability, security, and location-dependence, which can fulfill the requirement of delivering contents in small-scale networks.

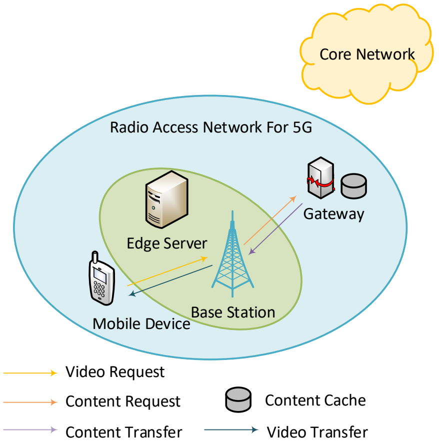
Jacobson et al. [8] in PARC thoroughly elaborated the basic properties of the CCN architecture. From model definition and routing rules, real-time streaming applications inspired by Voice over IP (VoIP), CCN security to authentication and Custodian-Based information sharing system design, providing rich enough technical details follow-up studies [9].

Different from what usually used in traditional host-centric structures, researchers paid more attention to CCN implementations. D'Ambrosio et al. [5] proposed a new hierarchical global name resolution service (NRS) to support CCN protocols in practical environments. Meanwhile, CCN also brings new opportunity for security and privacy which are considered a perennial issue in communication networks [10].

As a new network framework, many researchers try to transfer the traditional problems and applications into the CCN field, such as audio & video streaming [11], flow traffic control [12], and caching management [13]. Internet of Things (IoT) also shares many fitting points with CCN, and some works analyzed the possibility of fulfilling energy efficient communications [14]. CCN is also adopted in 5G RANs for improving the end-to-end network performance. For example, AMVS-NDN is a CCN framework for implementing the CCN protocol in commercialized WiMAX RANs [2].



(a) Convertional RAN



(b) Integrating ECC and CCN

Fig. 1. Network flows in conventional RAN and the integrating scheme of ECC and CCN

Integrating CCN into ECC is a fascinating innovation since CCN is efficient in the edge of networks. Eleonora et al. [15] presented a framework for building ECC by local nodes for IoT services. In the framework, local nodes manage data locally, and provide services to each other with the managed data. Meanwhile, functions in local nodes are integrated with cloud services. The framework also integrated with ICN platforms. However, this framework only focuses on mobile nodes which have limited computing capability.

III. ORCHESTRATING SCHEME OF ECC AND CCN FOR 5G

In this section, we first discuss the scenario of ECC and CCN in mobile computing, then introduce ECCN in the 5G radio access network.

As shown in Fig. 1, we use an example to show how ECC and CCN work together in a mobile environment. We choose a video streaming service for mobile users. Usually, mobile devices need a standard way, such as HTTP live

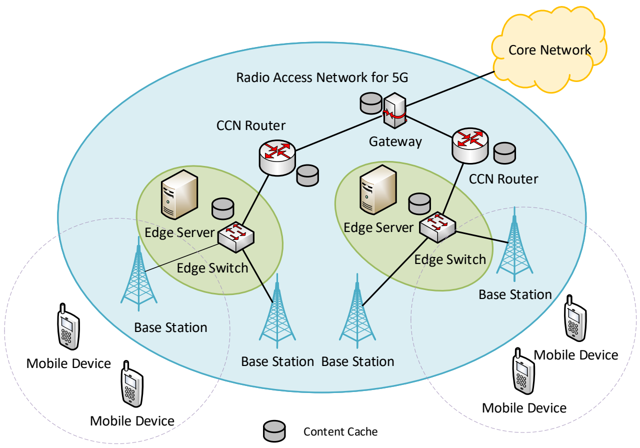


Fig. 2. Orchestrating scheme of edge centric computing and content centric networking for 5G raido access network

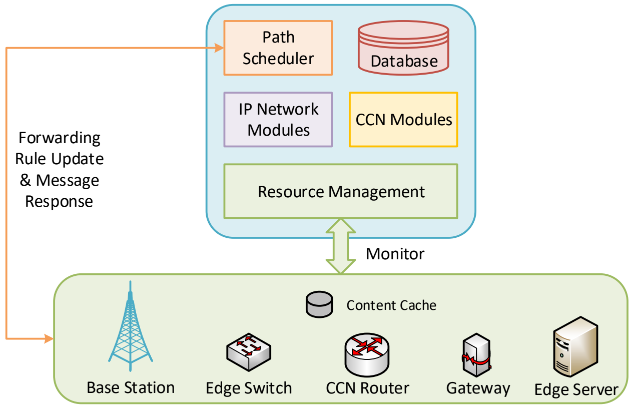


Fig. 3. Framework in the SDN controller for ECCN

streaming (HLS) or real-time streaming protocol (RTSP), for video streaming. Since video contents have different coding and encapsulation containers, a streaming server is needed for providing the video streaming service for mobile devices.

In the conventional RAN shown in Fig. 1(a), a mobile device first sends the video requests to the streaming server, then the streaming server requests related video content from a CDN server. After decoding and coding the video content, the streaming server sends the streaming video data to the mobile device. In the conventional RAN, all network flows for the video streaming should be forwarded across the core network, which brings high latency to the service and extra overhead to the core network.

Therefore, we present an example shown in Fig. 1(b) in which the RAN integrates ECC and CCN together for the video streaming service. The function of the streaming server is moved to the edge server and all contents are cached in the CCN-enabled RAN. Thus, after receiving the video request from the mobile device, the edge server sends related video content request with the CCN protocol. In the example, we assume a device in the forwarding route cached the required content and the device sends back the content to the edge server. Thus, there is no need to access the core network if a cache hit occurs in the RAN, and the latency and overhead are minimized with the integrating of ECC and CCN. If the required content is not cached in the CCN nodes in practice, the edge server still needs to download the content from the core network. Since the cache size in CCN nodes is limited to store all required contents, some CCN protocols use different cache strategies to improve the cache hit ratio. However, complex cache strategies will increase the forwarding overhead in CCN nodes. Thus, most CCN protocols choose first-in-first-out (FIFO) or least recently used (LRU) as the basic cache strategies.

We propose ECCN for the 5G RAN shown in Fig. 2 for supporting the discussed scenario. Edge servers are deployed in the devices near to the base stations, and all forwarding devices support the CCN protocol. The RAN is compatible with traditional network protocols, and all wireless communications follow the conventional way. CCN is limited between edge servers and the gateway for the compatibility with conventional network architecture.

In the proposed scheme, we focus on two important issues. The first issue is the forwarding protocol for two types of network flows. Since the proposed scheme supports both conventional and CCN network flows, a compatible forwarding protocol is needed to distinguish network flows for different forwarding strategies. For example, some conventional network flows should be forwarded with IP protocol, while CCN network flows are forwarded with content names.

The second issue is the network control for conventional and CCN network flows. Conventional and CCN network flows share same links in the RAN while conventional network protocols are ineffective in controlling CCN flows, and vice versa. Thus, we need to design a management method to control both conventional and CCN network flows in the RAN.

As a result, in ECCN for 5G RAN, we first decouple the control plane and data plane of both conventional and CCN networks, then design an SDN-based protocol for the forwarding strategies. We also design a management framework in the SDN control for control both conventional and CCN network flows.

IV. FRAMEWORK STRUCTURE AND SDN PROTOCOL

In this section, we fist introduce the framework structure of ECCN, then present the design of SDN protocol.

As shown in Fig. 3, we design a management framework in the SDN controller. Since all devices in the RAN are connected to the controller, the framework monitors the network resources in all devices and links. We also design additional fields in the SDN protocol to support the management of content cache. In the management framework, we also add two forwarding modules to support different forwarding strategies of IP and CCN protocols. The patch scheduler controls all forwarding rules in devices. The scheduler also calculates the forwarding path of each network flow. For the management of a large-scale RAN, it needs many pieces of information such as the network topology, link capacities, and the content cache to generate correct forwarding rules. Thus, we add a database for information preservation and retrieval of the RAN. The path scheduler will access the database to retrieve necessary information before generating new forwarding rules.

When a device receives a packet from a new network flow, the device sends a message with the new packet to the path scheduler. The path scheduler read the link information from the database then calculate the forwarding link for the new network flow. Since the forwarding rules of IP packet and CCN packet are different, the path scheduler accesses the corresponding modules for the forwarding strategies. For example, if the device received a new IP packet, the scheduler will generate forwarding rules with fields of the destination and source IP addresses. If the device received a new CCN packet, the scheduler will add the generate forwarding rules with fields of the content name and packet type. The framework also controls the forwarding path from base stations to edge servers. Only the network flows that needs edge services are forwarded to the edge servers and other network flows are directly forwarded to the RAN.

Therefore, it needs three types of forwarding rules in the SDN protocols. In most CCN implementations, CCN flows are supported by general IP networks. In those works, a CCN packet will be encapsulated into a UDP packet for further forwarding in the IP network. Thus, the network hardware can forward CCN flows as general IP flows. However, although some CCN works provide programmable interfaces for management, it is hard to manage those UDP packets without a centralized controller. In our work, since we added an identification protocol field in all CCN packet headers, the SDN controllers can manage CCN flows with specific forwarding strategies.

Since SDN protocols such as OpenFlow use different fields for packet identification, we add a protocol field in CCN packets and design a forwarding protocol.

In CCN forwarding, we use a rule to identify the interest packet. If the packet is an interest packet, the device will check whether the interested content is cached. If the interested content is cached, the device forwards the content back to the source port of the packet. Otherwise, the interest and the source port number are saved. If a content packet is received, the device forwards the packet to the port with the interest. After forwarding, the device caches the content if a cache miss happened. The overhead in the forwarding procedure is reasonable since most SDN devices have a hardware content-addressable memory for efficient prefix matching.

V. PERFORMANCE EVALUATION

In this section, we first introduce the testbed and the applications of the demonstration. We also discuss the experimental results with the demonstration. Then, we describe the settings of the extensive simulations. At last, we discuss the experimental results of the performance evaluation.

A. Demonstration with Network Applications

We use a small testbed for the implementation of a demonstration. The testbed consists an edge server, a gateway server, and a mobile device. As shown in Fig.4, we use KVM to provide two virtual machines connected by OpenvSwtich as the gateway and edge servers. The edge server has a virtual wireless network interface card for providing wireless access, and a virtual Ethernet network interface for connecting gateway server. The gateway server has two virtual Ethernet network interfaces, one is connected to the Internet and the other is connected to the edge server. We also modify the OpenvSwitch ovsbr1, ovsbr2 and ovsbr3 to support forwarding CCN packets.

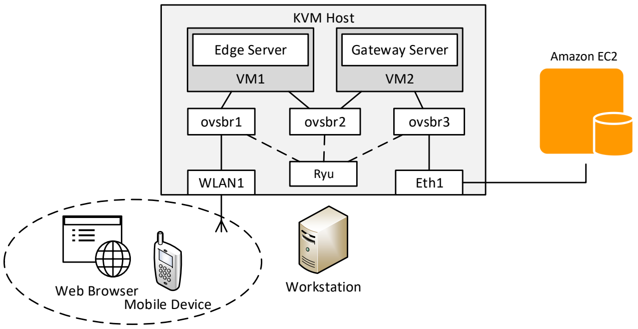
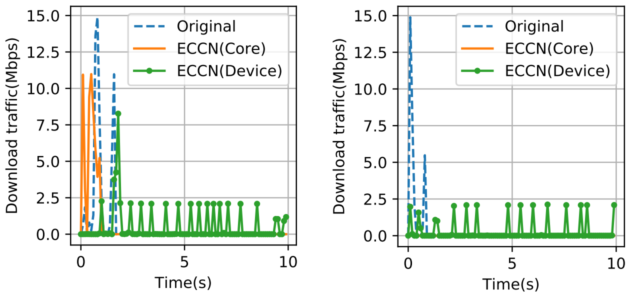
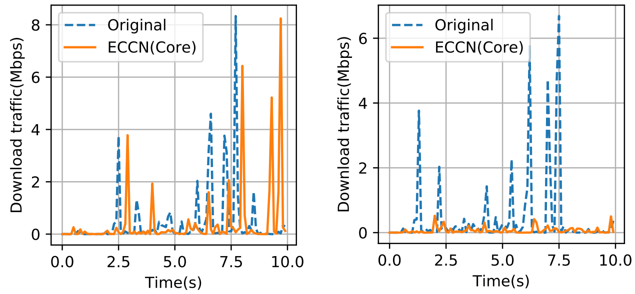


Fig. 4. Testbed framework for the demonstration applications



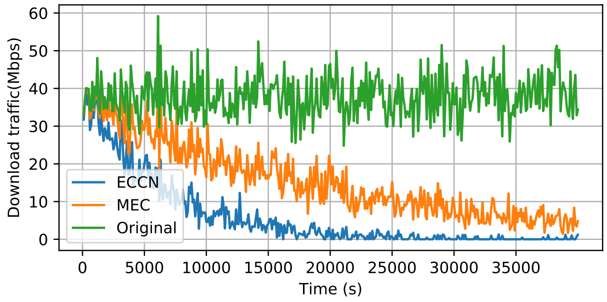
(a) First step of video streaming (b) Second step of video streaming



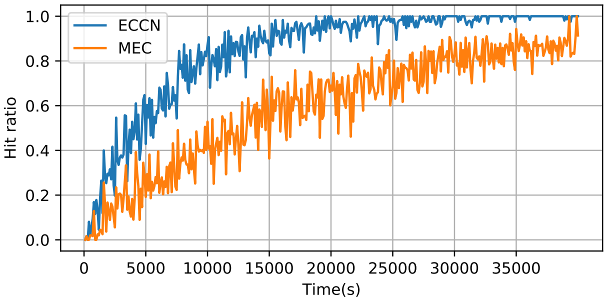
(c) First step of image identification (d) Second step of image identification

Fig. 5. Network traffic of two demonstration applications with original network and ECCN.

We implement two ECC-based applications. The first application is the video streaming service mentioned in Section III. The edge server decodes the video data and provides streaming video to the mobile device. We install and setup Nginx in the edge server for streaming video to the browser in the mobile device. Initially, all videos are stored in an Amazon EC2 server. The edge server sends a content interest packet to the gateway server for requiring video data. If the video data



(a) Download traffic from the cloud server



(b) Hit ratio in the CCN

Fig. 6. Download traffic and hit ratio in the CCN with different methods

have been cached, the gateway server sends the data back with CCN content packets. Otherwise, the gateway server asks the cloud server with conventional IP protocols.

The second application is an image identification service. Users upload an image to the server for identifying the content of the image. The server also sends back some reference images for the uploaded image. In our testbed, we install a flower image identification service in the edge server while reference flower images are stored in the cloud server. When the mobile device uploads a flower image to the edge server, the edge server will identify the species of the uploaded flower image first, and then select 10 flower images in the same species for reference. For requiring images, the edge server sends content interest packets to the gateway server. As well as the video streaming service, the gateway server sends back the cached data and asks the cloud server for cache missed data.

We test the network traffic of all links between the mobile device and the edge server, the edge and the gateway server, and the gateway and the cloud server, respectively. In the test of the video streaming service, we use two steps for evaluating the CCN protocols. In the first step, the mobile device requires a video from the edge server and receive the first 10 seconds video. In the second step, the mobile device requires the video again and receives the last 10 seconds video. We also use two steps in the image identification service. The mobile device sends the same image to the edge server in each step.

As shown in Fig. 5, we record the download traffic of the gateway server and the device denoted by “ECCN(Core)” and “ECCN(Device)”, respectively. We also record the device download traffic with original network. From Fig. 5(a), in the first 10 seconds of video streaming, both local browser and the edge server buffers video data from the remote cloud server in the first 2 seconds. Because of the low latency and reliable connection between the device and edge server, the buffered data size is set to minimum. In the second step, there is no downloading traffic between the gateway server and the cloud server because all video data are buffered in the edge server. Since the device browser cleans all buffered data before the second step, the device needs download the rest of video data from the cloud server with the original network.

In the Fig. 5(c) and 5(d), the image identification application shows similar results with the video streaming service. In the first step, both ECCN and original network needs the device to download image data from the cloud server. In the second step, as the CCN has buffered the same image data, there is little traffic except for some necessary interactions. The image identification application with the original network still needs to download the image data from the cloud server. As a result, from the download traffic results of two demonstration applications, the ECCN scheme can reduce the traffic between RAN and the cloud server.

B. Numerical simulation

We also evaluate the performance ECCN scheme in numerical simulations with a large-scale network. We use Python 2.7.13 and NetworkX 2.0 to build the simulation application. We use a tree topology RAN for connecting devices and the core network. We use a 500m x 500m area with 30 mobile devices moving in the Random WayPoint model. We place 8 base stations in the area and use seven switches to connect all base stations. In each base station, an edge server is deployed for edge services. We use a set of 15 services and a device in a given interval select a service from the edge server. For each service, edge servers need to download related contents from the RAN. The total cellular network capacity is set to 300Mbps for download. We assume the capacity of each CCN node is big enough for buffering all data transferred in each simulation. The size of required data is uniformly distributed in the range from 40MB to 160MB. We use 3000 files and the requirement probability of each file is uniformly distributed. The interval between two service requests is uniformly distributed in the range from 20s to 160s. We also test the performance of the original model and a mobile edge computing method denoted by ”MEC” for comparison. In the mobile edge method, devices provide edge services and CCN protocol is built between devices [15]. We set the simulation time to 40000s per each execution. Simulations are executed 10 times and the average results are recorded.

We test the download traffic and hit ratio in the CCN with three different methods. As shown in Fig. 6(a), two CCN-based methods decrease the download traffic from the cloud server. With the original mode, the entire 5G network needs to download around 50Mb data from the cloud server per each second. Since required data is buffered in each node, transferring repeating data is reduced between CCNs and the cloud server. The download traffic with ECCN is reduced near to 0 after 20000s while the download traffic with the mobile edge computing method is more than 3Mbps after 35000s.

Our method performs better than the mobile edge computing method since the hit ratio of ECCN is higher. From the results shown in Fig. 6(b), the hit ratio with ECCN is near to 100% after 20000s while the mobile edge computing method only has 60 % hit ratio at the same time. Because of the mobility of devices, the nearby nodes buffered required data will move out of the communication range then the device has to require the same data from the cloud server. In ECCN, since all data are buffered in the RAN, the hit ratio will be more stable. Moreover, the capability of each mobile device is not enough for buffering all required data. Thus, ECCN is a more appropriate method for providing contents to mobile devices in 5G networks.

VI. CONCLUSIONS

We proposed a new scheme named ECCN to integrate ECC and CCN in a 5G RAN. We also introduce SDN to manage two different network technologies in the hierarchical structure. Two demonstration applications are implemented with ECCN and the download traffic between RAN and the cloud server are reduced with buffers in CCN devices. We also test our method with numerical simulations, focusing on the network traffic and hit ratio in service provision. From the experimental results, ECCN outperforms the mobile edge computing method because of stable CCN devices in 5G RAN. As a result, our solution is an appropriate way to improve the service capability of 5G mobile networks.

ACKNOWLEDGMENTS

This work is supported by JSPS KAKENHI Grant Number JP16K00117, JP15K15976, JP17K12669, KDDI Foundation, and Research Fund for Postdoctoral Program of Muroran Institute of Technology. Mianxiong Dong is the corresponding author.

REFERENCES

[1] D. Perino and M. Varvello, "A reality check for content centric networking," in *Proceedings of the ACM SIGCOMM Workshop on Information-centric Networking*, ser. ICN '11. New York, NY, USA: ACM, 2011, pp. 44–49.

[2] B. Han, X. Wang, N. Choi, T. Kwon, and Y. Choi, "AMVS-NDN: Adaptive mobile video streaming and sharing in wireless named data networking," in *2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2013, pp. 375–380.

[3] P. Garcia Lopez, A. Montresor, D. Epema, A. Datta, T. Higashino, A. Iamnitchi, M. Barcellos, P. Felber, and E. Riviere, "Edge-centric computing: Vision and challenges," *SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 5, pp. 37–42, Sep. 2015.

[4] X. Tao, K. Ota, M. Dong, H. Qi, and K. Li, "Performance guaranteed computation offloading for mobile-edge cloud computing," *IEEE Wireless Communications Letters*, vol. PP, no. 99, pp. 1–1, 2017.

[5] M. D'Ambrosio, C. Dannewitz, H. Karl, and V. Vercellone, "Mdht: A hierarchical name resolution service for information-centric networks," in *Proceedings of the ACM SIGCOMM Workshop on Information-centric Networking*, ser. ICN '11. New York, NY, USA: ACM, 2011, pp. 7–12. [Online]. Available: http://doi.acm.org/10.1145/2018584.2018587

[6] A. Gudipati, D. Perry, L. E. Li, and S. Katti, "Softran: Software defined radio access network," in *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '13. New York, NY, USA: ACM, 2013, pp. 25–30.

[7] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, October 2016.

[8] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '09. New York, NY, USA: ACM, 2009, pp. 1–12. [Online]. Available: http://doi.acm.org/10.1145/1658939.1658941

[9] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26–36, July 2012.

[10] A. Chaabane, E. De Cristofaro, M. A. Kaafar, and E. Uzun, "Privacy in content-oriented networking: Threats and countermeasures," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 3, pp. 25–33, Jul. 2013. [Online]. Available: http://doi.acm.org/10.1145/2500098.2500102

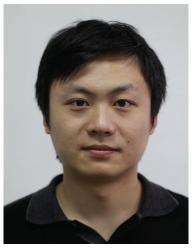
[11] P. Gusev and J. Burke, "Ndn-rtc: Real-time videoconferencing over named data networking," in *Proceedings of the 2Nd ACM Conference on Information-Centric Networking*, ser. ACM-ICN '15. New York, NY, USA: ACM, 2015, pp. 117–126.

[12] S. Oueslati, J. Roberts, and N. Sbihi, "Flow-aware traffic control for a content-centric network," in *Proceedings of 2012 IEEE INFOCOM*, March 2012, pp. 2417–2425.

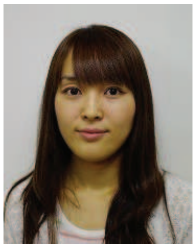
[13] Z. Liu, M. Dong, B. Gu, C. Zhang, Y. Ji, and Y. Tanaka, "Impact of item popularity and chunk popularity in ccn caching management," in *Proceedings of 2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, Oct 2016, pp. 1–6.

[14] E. Baccelli, C. Mehlis, O. Hahm, T. C. Schmidt, and M. Wählisch, "Information centric networking in the iot: Experiments with ndn in the wild," in *Proceedings of the 1st ACM Conference on Information-Centric Networking*, ser. ACM-ICN '14. New York, NY, USA: ACM, 2014, pp. 77–86.

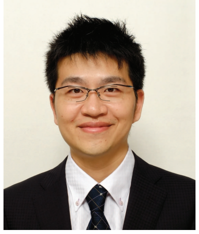
[15] E. Borgia, R. Bruno, M. Conti, D. Mascitti, and A. Passarella, "Mobile edge clouds for information-centric iot services," in *2016 IEEE Symposium on Computers and Communication (ISCC)*, June 2016, pp. 422–428.



He Li received the B.S., M.S. degrees in Computer Science and Engineering from Huazhong University of Science and Technology in 2007 and 2009, respectively, and Ph.D. degree in Computer Science and Engineering from The University of Aizu in 2015. He is currently a Postdoctoral Fellow with Department of Information and Electronic Engineering, Muroran Institute of Technology, Japan. His research interests include cloud computing and software defined networking.



Kaoru Ota was born in Aizu Wakamatsu, Japan. She received M.S. degree in Computer Science from Oklahoma State University, USA in 2008, B.S. and Ph.D. degrees in Computer Science and Engineering from The University of Aizu, Japan in 2006, 2012, respectively. She is currently an Assistant Professor with Department of Information and Electronic Engineering, Muroran Institute of Technology, Japan. She serves as an editor for IEEE Communications Letter.



Mianxiong Dong received B.S., M.S. and Ph.D. in Computer Science and Engineering from The University of Aizu, Japan. He is currently an Associate Professor in the Department of Information and Electronic Engineering at Muroran Institute of Technology, Japan. Dr. Dong serves as an Editor for IEEE Communications Surveys and Tutorials, IEEE Network, IEEE Wireless Communications Letters, IEEE Cloud Computing, and IEEE Access.