NeiMEC：Automatically Building Neighbor Relationship between Mobile Edge Platforms in Multi-access Edge Computing Environment

Yen-Chiu Chen, Chun-Chieh Wang and Jian-Cheng Chen
Information and Communications Research Laboratories (ICL)
Industrial Technology Research Institute (ITRI)
Hsinchu, Taiwan, R. O. C.
{amanda.chen, jjwang, JianCheng}@itri.org.tw

***Abstract*—Multi-access Edge Computing (MEC) is a key technology towards achieving a 5G system. Via MEC technology, both the service latency and cost of the backhaul network can be decreased in 5G environments; however, the MEC servers work independently, without considering the status of other MEC servers, which may lead to some difficulties in meeting the 5G requirements defined by ITU-R. This paper proposes a novel architecture for multiple MEC servers, named NeiMEC, to define the neighbor relationship of multiple MEC servers. Via NeiMEC, the MEC servers can refer to the statistical information of neighbor MEC servers to apply an advanced content cache replacement policy when its content cache is full. The NeiMEC-based content cache replacement policy will improve the hit ratio of the contents (by up to 42%) above the hit ratio of systems that don't consider neighbor servers. Consequently, the latency of accessing service is improved.**

Keywords—Multi-access Edge Computing (MEC); 5G system; neighbor MEC; Content cache replacement policy

I. INTRODUCTION

With the popularity of mobile devices and Internet services continuously, not only the throughput of mobile network keeps increasing, but also the latency for user experience is expected more and more strictly. For future 5G network, the properties of 5G performance requirements defined by International Telecommunication Union-Radio communication (ITU-R)[1] are low latency, high capacity, and more consistent experience for 5G service. For instance, the over-the-air latency is 1 ms, and area traffic capacity is 10 Tbps/km^2. Moreover, the range of the end-to-end latency is 5-10 ms proposed by next generation mobile networks (NGMN)[2] and 5GPPP[3]. However, in current 4G network architecture, all packets should go through Evolved Packet Core network (EPC), it is obviously difficulty to meet the latency requirement for 5G services. Furthermore, the cost to establish high capacity for current Backhaul network is too expensive and inefficiency. Thus, Multi-access Edge Computing (MEC) technology is considered as a cutting-edge technology towards 5G which is currently standardized in European Telecommunications Standards Institute (ETSI) Industry Specification Group (ISG).

According to MEC market forecasts from 2016 to 2021 year[4] by Mind Commerce, the MEC market will be $82.2 billion by 2021, where the MEC market represents a combination of MEC Cloud Server(7%), MEC Equipment(19%), MEC Platform System(14%), MEC Software & API(56%), and MEC Services(3%). ETSI MEC ISG propose five MEC use cases to show the advantage of MEC technology as follows: distributed data caching, dynamic content optimization, intelligent video analytics, location service, Augmented Reality content delivery. To take distributed data caching as an example, 20% improvement for loading a Web page and Backhaul traffic savings up to 35% when popular content/data are stored at a MEC server which is near/at a Base Station[5]. However, until now, most MEC application scenarios are deployed under MEC servers work independently without considering the status of other MEC servers. Hence, the latency for MEC services is not significantly decreased.

This paper proposes a novel architecture for multiple MEC servers, named as NeiMEC, to define the neighbor relationship between multiple MEC servers. According to the MEC neighbor relationship, some statistical information, such as a referenced count of a content cache or CPU loading information of a MEC server, can be shared between neighbor MEC servers. By the shared information based on NeiMEC, the performance of a MEC system can be improved.

In this paper, one MEC application scenario is set as content cache replacement when the content cache is full at a MEC server. With the popularity of mobile devices, it is more convenient for mobile users to watch movies or obtain digital online content. Mobile users may watch a movie from one service range of a MEC server to another in a time period. However, each MEC server has different space limitation. How to select and drop contents when the content cache is full at a MEC server is an important problem. The fundamental content cache replacement algorithms, such as Least Frequently Used Algorithm (LFU) and Least Recently Used Algorithm (LRU) applied to a MEC server are only based on the statistical information at the same MEC server. The replacement policy without considering the information of the neighbor MEC servers may lead to mistakenly drop popular but unused contents in a short time period. And then the content should be reloaded at a remote content server in cloud environments. Consequently, the time to reload contents will be much than the

978-1-5386-5896-3/18/$31.00 ©2018 IEEE

time to access the contents at a MEC server which is located at logical edge of the mobile network.

The main contributions proposed in this paper as follows: (1) to propose a mechanism to automatically build neighbor relationships between multiple MEC servers; (2) to improve the hit ratio of the content cache replacement by the information of neighbor MECs. The remainder of the paper is structured as follows. An overview of related work is presented in Section 2. In Section 3, a system architecture for defining multiple MEC servers, named as NeiMEC, and an advanced content cache replacement policy are detailed. A content cache replacement policy and the advanced content cache replacement policy with considering the information of other MEC servers are compared in Section 4. Section 5 concludes this paper.

II. Related Works

A. Neighbor Relationships of Edge-based Servers

MEC is one of the computation technologies originated from the concept of Cloud data-centers. The end users or mobile devices are usually far from a Cloud datacenters which are geographically centralized. Hence, the concept of setting light-weight servers closer to the end user overcomes the shortage such as large round-trip delay or service quality degradation in Cloud computing. Moreover, Mobile Cloud Computing (MCC) and Fog computing are also the extensions of Cloud and Edge computing, where MCC can be presented as cloudlet. Regardless of MEC, cloudlet or Fog computing, light-weight servers are located closer to the end user to provide low-latency services. Hence, this kind of light-weight is called as *edge-based servers* in this paper.

Currently, the concept of neighbor relationships for edge-based servers is used in some work. Dimas et al.[6] propose a recovery mechanism for overloaded mobile edge computing by clustering a set of the MEC servers according to the geographic locations of the MEC servers, where the set of MEC servers neighbor with each other. J Michel et al.[7] geographically define the neighbor relationship of cloudlets to propose a cloudlet-based proximal discovery service for machine-to-machine applications. For a set of data centers located on a logical ring at the edge of the network, C Fricker et al.[8] propose an offloading scheme for data centers in fog computing by forwarding some traffic to a neighboring data center when a data center is overloaded.

However, there is no precise definition or mechanism to automatically build neighbor relationships for multiple Edge-based servers. For operators or management information system engineers, it wastes time and cost to configure all equipment one by one, especially for edge-based servers in small cell network environments. Therefore, how to automatically build the neighbor relationship of edge-based servers in mobile network environments is important.

To the best of our knowledge, the interface between two MEC servers is being standardized by ETSI MEC ISG[9], and currently there is no study results about the mechanism to build neighbor relationships of multiple MEC servers. Here, one mechanism to automatically build neighbor relationships for multiple MEC servers is proposed in this paper.

B. Content Cache Strategies

Essentially, MEC technology is expected to reduce the traffic and provide low latency of accessing services. If the content needed by mobile users can be get from the content cache of a MEC server, the mobile users do not waste too much time to get the contents stored in cloud environment. Hence, how to effectively make use of the content cache at a MEC server is an interesting issue.

Most of the studies about caching sharing for base stations (BSs) by the MEC technology are the location or the size of the content cache and the link flow. J. Elias et al. proposed an optimal geographic caching of content in cellular networks by combining linear content coding techniques and cellular network coverage models from stochastic geometry [10]. Based on the economics framework of contract theory, an approach for providing incentives for caching in small cell networks (SCNs) is proposed by K. Hamidouch et al.[11].

With respect to the content cache replacement policy, a MEC server can depend on conventional caching strategies, including Least Recently Used (LRU) and Least Frequently Used (LFU), to replace the contents of its content cache when the content cache is full. LRU discards the least recently used content and LFU discards the least frequently used content. Ge Ma et al.[12] investigate conventional LRU and LFU caching strategies for edge network mobile video delivery and propose a caching strategy based on some factors that affect content caching performance.

III. System Architecture

In this section, a system architecture, named as NeiMEC, is defined to describe the neighbor relationship for multiple MEC servers. Moreover, one MEC application scenario is set as content cache replacement that considers the information from neighbor MEC servers to show the impact of the proposed NeiMEC system.

A. Automatically Building MEC Neighbor Relationship

NeiMEC system is designed based on the Multi-access Edge Computing Framework and Reference Architecture proposed by ETSI MEC ISG[9], which is shown as Figure 1. In the MEC Architecture, a Mobile edge host includes a Mobile edge platform (MEP) and a virtualization infrastructure. The MEP hosts ME services composed by one or multiple ME apps, where a ME app is considered as a virtualized network function running on a virtual machine. The virtualization infrastructure provides compute, storage and network resource to ME services. After receiving the requirements from Mobile users by UE app or Customer facing service (CFS) portal, Mobile edge orchestrator and Operation Support System can manage Mobile edge hosts via mobile edge platform manager and Virtualization infrastructure manager. For consistency, a MEC server represent a Mobile edge host of the ETSI MEC Architecture in this paper.

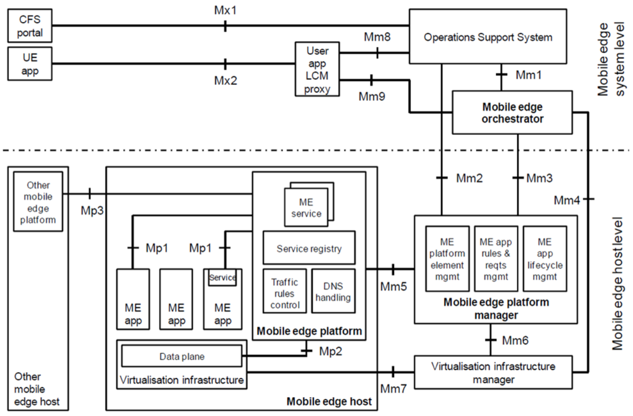


Figure 1. ETSI MEC Framework and Reference Architecture.

With different deployment demand, MEC servers can be deployed [13] at the base station (BS) site, at the 3G Radio Network Controller (RNC) site, at a multi-Radio Access Technology (RAT) cell aggregation site, and at an aggregation point which may also be at the edge of the mobile network. Furthermore, one MEC server can cooperate with one base station or a small cell, or one MEC server works toward multiple base stations or small cells depending on the capacity of the equipment.

In proposed NeiMEC system, one MEC server cooperating with a base station is considered, and MEC, BS represent the MEC server and the base station such eNodeB or small cell in Figure 2, respectively. Each BS is managed by a NMS (network management system) or an EMS (element management system, EMS), where the NMS or EMS is virtualized as a ME app.

The NMS or EMS cooperates with a MEP by ETSI MEC reference point Mp1. Multiple MEPs can communicate with each other by reference point Mp3[9]. Additionally, a neighboring manager, denoted as NeiMgr, is responsible for determining and establishing the neighbor relationship between multiple MEPs by collecting the identifications (ID) of the MEPs and the IDs of corresponding BSs of each MEP. The trigger time to establish the MEP neighbor relationship is when one of the MEPs attempts to refer some statistical information of its neighbor MEPs.

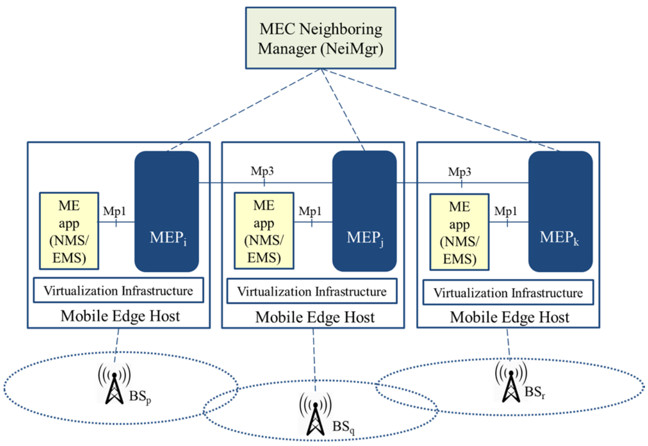


Figure 2. NeiMEC System Architecture.

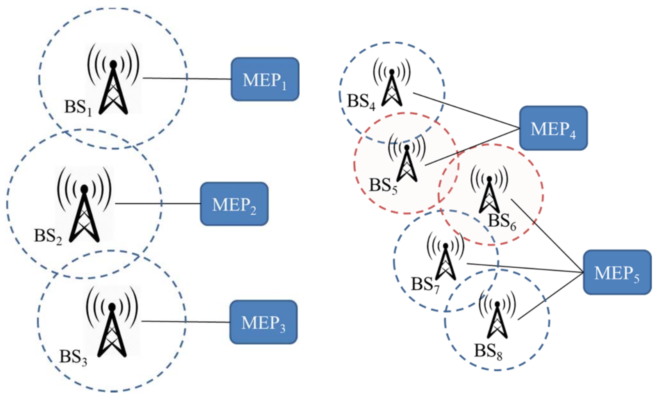
According to the Automatic Neighbor Relation (ANR) management defined by 3rd Generation Partnership Project (3GPP)[14], each BS can use the ANR function to define the neighbor relationship of other BSs. Certainly, the ANR function resides in the BS. When BS_p is defined as a neighbor of BS_q, it means some part of the communication range of BS_p overlaps some part of the communication range of BS_q. That is, the communication range of BS_p and BS_q is somewhere overlapped.

In this paper, the neighbor relationship between two MEPs is defined below.

Definition 1. *Neighboring MEP:* MEP_i is defined as a neighbor of MEP_j when BS_p is a neighbor of BS_q, where MEP_i and MEP_j cooperate with BS_p and BS_q, respectively.

In Figure 3(a), BS_1 is a neighbor of BS_2 by ANR communication protocol, thus, MEP_1 is a neighbor of MEP_2, where MEP_1 and MEP_2 cooperate with BS_1 and BS_2, respectively. Moreover, In Figure 3(b), if BS_5 is a neighbor of BS_6, MEP_4 is a neighbor of MEP_5, where MEP_4 and MEP_5 cooperate with BS_5 and BS_6, respectively.

Note that, the neighbor relationship of multiple MEPs is not fixed at all the time, especially when the corresponding BSs are small cells. The neighbor relationship of multiple MEPs will be changed if the communication ranges of BS_p and BS_q is not overlapped by some conditions, such as the communication signal of BS_q is weakened or BS_p does not cooperate with MEP_i. Different from traditionally static configuration, NeiMEC system provides dynamic configure to describe the neighbor relationship between multiple MEPs according to **Definition 1**. For example, MEP_4 is not a neighbor of MEP_5 in Figure3 (b) when BS_5 does not cooperate with MEP_4, or the communication range of BS_5 is shrink and not overlapped to that of BS_6.



(a) (b)

Figure 3. (a) One MEP cooperates with one site (b) One MEP works toward multiple sites.

The schematic diagram of automatically building the neighbor relationship between MEPs is shown in Figure 4. A NMS or EMS manages a set of BSs and sends the information of BSs to a MEP in steps 1-3, where the BSs are neighbors defined by ANR communication protocol. Each MEP send its ID and corresponding BS ID to NeiMgr. When one of the MEPs, such as MEP_2, attempts to refer some statistical information of its neighbor MEPs, MEP_2 sends a query message to NeiMgr in steps 5. NeiMgr checks the neighbor information collected in priori step 1-3, and replies MEP_2 about the information of MEP_1, e.g. MEP_1 ID and IP address. And then, MEP_1 and MEP_2 can build a connection and communicate with each other in step 8.

According to the **Definition 1**, the neighbor relationship between MEP_1 and MEP_2 can be automatically built when the communication ranges of BS_1 and BS_2 is not overlapped. Moreover, this automatically building process can decrease the time and cost of the manually setting operations when the corresponding BSs are a large set of small cells.

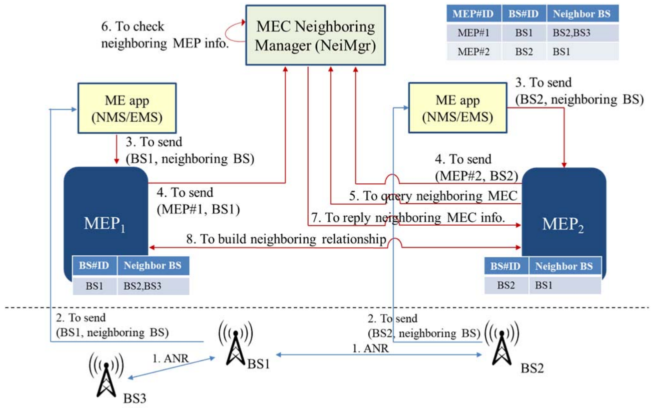


Figure 4. Automatically building process in NeiMEC.

B. *Content Cache Replacement Considering Neighbor MEC Statistical Information*

In order to show the advantage of the proposed NeiMEC system, one MEC application scenario is set as content cache replacement that considers the information from neighbor MEC servers.

The size of the storage space at a MEC server is finite. When the content cache is full at a MEC server, some content will be selected to replace by the fundamental content cache replacement algorithms, such as Least Frequently Used Algorithm (LFU) and Least Recently Used Algorithm (LRU). However, some contents which are popular but unused in a short time period maybe mistakenly dropped by the replacement policy without considering the information of the neighbor MEC servers.

Let the LFU Algorithm without considering the statistical information of the neighbor MEPs be presented as $F(MEP_i, content_{ID})$, a scenario is shown in Figure 5. There are a lot of users watching movies by mobile devices, and the most popular contents are stored at the MEC servers rather than the servers in cloud environments. When the cache of MEP_2 is full, Video#01will be deleted by the LFU Algorithm.

However, some mobile users who are watching Video#01 are moving from the communication range of BS_1 to BS_2 in a short period time, but they cannot immediately access Video#01 at MEP_2 since Video#01 was deleted from MEP_2 by the LFU Algorithm. In other words, they have to access the Video#01 content from the video server in cloud environments. Obviously, if the content is already at the MEC server, the latency time to access contents at the MEC servers is shorter than that at the servers in cloud environments. Furthermore, by improving the hit ratio of accessing contents at the MEC servers, the latency time will be decreased.

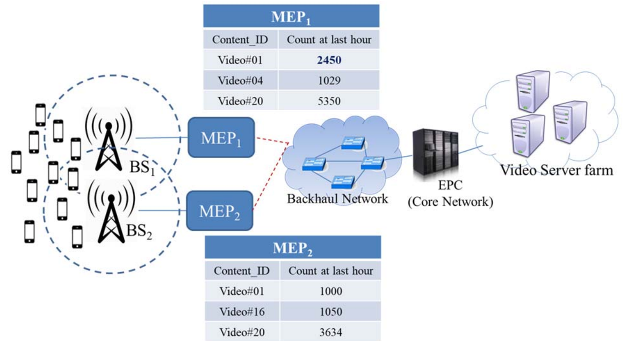


Figure 5. A content cache scenario.

In NeiMEC system, the statistical information of a MEP can be shared between its neighbor MEPs. Hence, a NeiMEC-based LFU Algorithm can be modified as the following **Definition 2**.

Definition 2. *NeiMEC-based LFU Algorithm*

$$F'^{(MEP_i, content_{ID})} = F(MEP_i, content_{ID}) + \sum_{j \in Neighbor(i)} \alpha_j \times F(MEP_j, content_{ID})$$

where MEP_j and α_j present the neighbor of MEP_i and the weight to refer the statistical information of MEP_j, respectively.

As the above scenario, since MEP_2 can get the statistical information of its neighbor, MEP_1, Video#16 will be selected by the NeiMEC-based LFU Algorithm. The mobile users can immediately access Video#01 which is still stored at MEP_2. Consequently, a NeiMEC-based LFU Algorithm can be applied to improve the hit ratio of accessing contents at the MEC servers.

Extended to NeiMEC system, a NeiMEC-based content cache replacement system architecture is shown as Figure 6. One ME service at each MEP periodically collects the information of its content cache, such as the number of requests for accessing a content in a time period. Additionally, a Centralized Replacement Management (CRM) manages the information of content cache at each MEP and provides a replacement suggestion based on NeiMEC-LFU algorithm to a MEP.

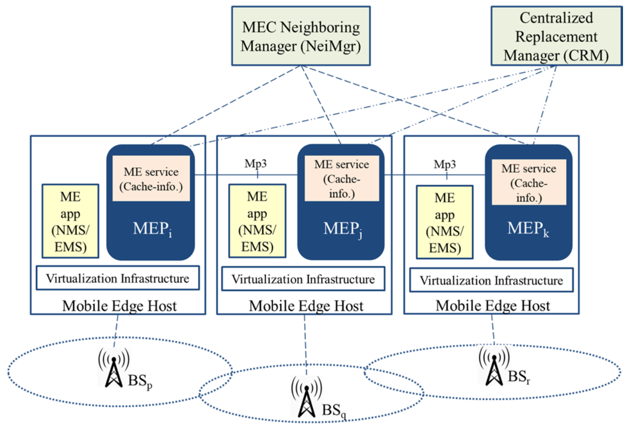


Figure 6. NeiMEC-based content cache replacement system.

Figure 7 demonstrates a process of the recommended replacement by CRM in the NeiMEC-based content cache replacement system. MEP_2 sends a request to ask CRM what content should be dropped when the space of content cache at MEP_2 is full in step 1. CRM consults with NeiMgr and gets the information about the neighbors of MEP_2 in steps 2-3. In this case, CRM gets the information of MEP_1 and MEP_3 that are the neighbors of MEP_2. Then, CRM queries the number of requests for accessing contents per hour at MEP_1 and MEP_3 in steps 4-5. Based on the information from each cache of neighbor MEPs, CRM uses the result of NeiMEC-LFU algorithm to notify MEP_2 about the recommended replacement in steps 6-7.

Specially, CRM can set the weights α according to the current events at that time. Take an example, a local concert is holding closely to MEP_i, CRM can set α_j be the average weights of all neighbor MEP_j defined in **Definition 2**. On the other way, CRM may set α_j be the average weights of all MEP_j managed by CRM defined in **Definition 3** when a current event is followed by global mobile users.

Definition 3. *Global-based LFU Algorithm*

$$F'(MEP_i, content\_ID) = \sum_{j=1}^{m} \alpha_j F(MEP_j, content\_ID),$$

where *m* presents the number of all MEPs managed by CRM and α_i denotes as the weights to refer the statistical information of each MEP_j.

However, how to determine that a current event is locally or globally impact, called local event or global event, is not discussed in this paper.

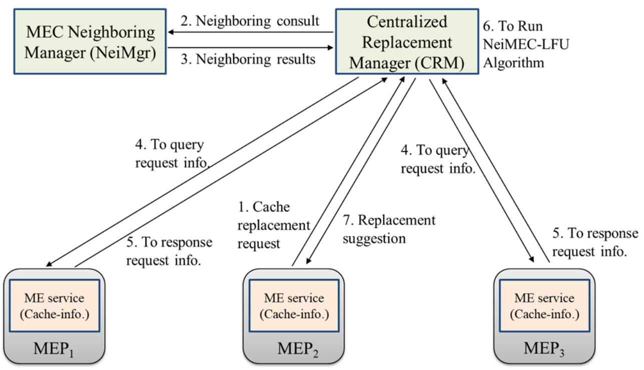


Figure 7. A process of the recommended replacement by CRM in the NeiMEC-based content cache replacement system.

IV. Experimental Results

NeiMEC-based content cache replacement is set to show the advantage of the proposed NeiMEC system. By the NeiMEC-based LFU Algorithm, a MEC server considers the information from neighbor MEC servers when the content cache is full.

As shown as Figure 8, this paper randomly deployed 16 BSs and 1000 UEs between $100km^2$. For the simulation of NeiMEC-based content cache replacement, each UE follows a Zipf distribution to access a set of 1000 video files from BS with skew parameter γ. In order to present the variety of the popularity close to the real world, let the skew parameter γ be from 0.4 to 1.4. Table I gives the parameters used in the simulation.

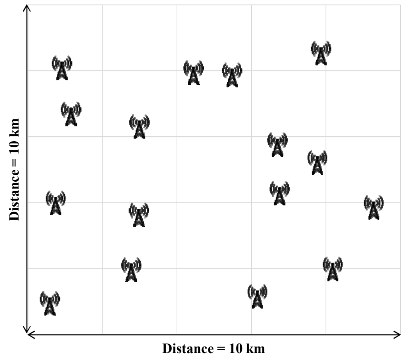


Figure 8. 4x4 BSs distribution.

TABLE I. Simulation Environment

Parameter	**Value**
Distribution Scale	10,000m×10,000m = 100 km^2
Number of UEs	1000
Number of Videos	1000 (1GB/per)
The popularity character of Videos	To follow a Zipf distribution with skew parameter γ
The skew parameter γ	0.4 ~ 1.4
Number of BSs	16
Cache size	5~15GB[15]

The latency time to get a video content from the video server in cloud environments is much more than the latency time to get the video content in a MEC server of the edge network. In other words, the better hit ratio of accessing contents at the MEC servers, the shorter latency time to access needed content. Therefore, the following performance comparison is indexed by hit ratio.

Referring **Definition 2**, the frequency of content in NeiMEC-based LFU Algorithm is referred the statistical information of neighbor MEP_i. For simplicity, let notation α represent as the average weight of the referring neighbor MEP. Note that the result of NeiMEC-based LFU Algorithm can be seen as the result of LFU Algorithm when α is zero.

Two types of scenarios were considered in the experiments: 1) global event and 2) local event. Figure 9 and 10 show the effect of the popularity profile on the performance with global events and local event, respectively. The hit ratio comparison of referring neighbor MEP weight α when the cache size is 15GB. With the respect to different Zipf distribution with skew parameters γ, the hit ratio is still greater if the referring neighbor MEP weight α is greater for any skew γ. Note that, it can be obviously that the difference between the hit rates increases almost linearly when cache size is increased. However, as shown in Figure 9, it can be clearly seen that global-based algorithm holds all information from among BSs that can significantly improve hit ratio.

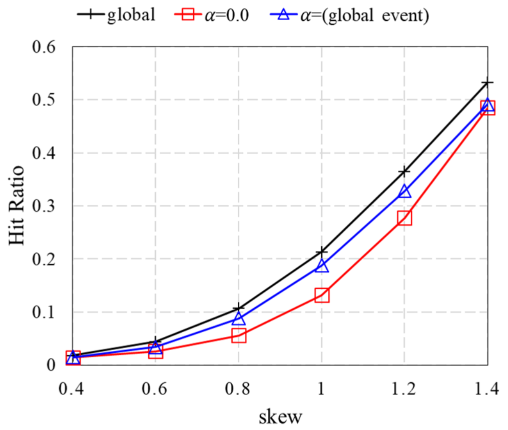


Figure 9. Performance comparison of the proposed algorithms for different Zipf distribution parameters γ under global events.

In the other hand, Figure 10 shows the hit ratio performance for local events. The prefer query of user's will form a localization region while skew γ increasing. The global-based algorithms will be interfered with other information that decreases the hit ratio. Therefore, the proposed NeiMEC-based LFU Algorithm can present the parameter α to improve the hit ratio for MEP.

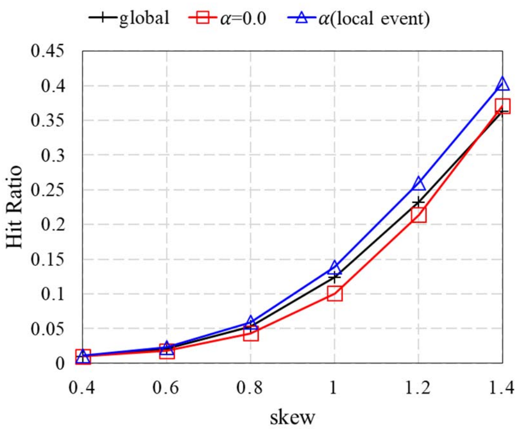


Figure 10. Performance comparison of the proposed algorithms for different Zipf distribution parameters γ under local events.

Let's consider the Zipf distribution with skew parameters γ is 1.0 in Figure 10. When the cache size is 15GB and the referring neighbor MEC weight α are 0 and 0.7, the hit ratio are 0.133 and 0.189, respectively. That is, the hit ratio is improved approximately 42% when a MEC server refers the information from its neighbor MEC.

V. CONCLUSIONS

The MEC market is full of business opportunities and optimism. In addition, there are interesting research challenges, such as the deployment for MEC system, cache-enabled MEC, mobility management for MEC, green MEC, privacy-aware MEC and security issues [16]. In this paper, a novel mechanism to build neighbor relationships between multiple ME platforms (abbr. as NeiMEC) is proposed. Moreover, a NeiMEC-based LFU Algorithm can be used to improve the hit ratio of accessing needed contents on the cache when the content cache of a MEC server is full. By improving the hit ratio on the cache of a MEC server, mobile users can get contents on a MEC server rather than on a content server in cloud environments. That is, the latency time to accessing needed contents can be significantly decreased.

REFERENCES

[1] ITR-U M.2083-0, "IMT-Vision-Framework and Overall Objectives of the Future Development of IMT for 2020 and Beyond," September 2015.

[2] NGMN Alliance, "NGMN_5G_White_Paper," 2015. [Online]. Available:https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf

[3] 5G Infrastructure Association, "5G Vision," February, 2015. [Online]. Available:https://5g-ppp.eu/wp-content/uploads/2015/02/5G-Vision-Brochure-v1.pdf

[4] Mobile Edge Computing (MEC): Market Assessment and Forecasts 2016 – 2021, October 2016.

[5] Rolf Schuster, "Boosting User Experience by Innovating at the Mobile Network Edge." [Online]. Available: http://www.etsi.org/images/files/technologies/MEC_Introduction_slides__SDN_World_Congress_15-10-14.pdf

[6] Dimas Satria, Daihee Park and Minho Jo, "Recovery for Overloaded Mobile Edge Computing", *Future Generation Computer Systems*, vol. 70, issue C, May 2017, pp. 138－147.

[7] Jonas Michel and Christine Julien, "A Cloudlet-Based Proximal Discovery Service for Machine-to-Machine Applications," *The 5th International Conference on Mobile Computing, Applications, and Services*(MobiCASE 2013) , pp. 215－232, 2013.

[8] Christine Fricker, Fabrice Guillemin, Philippe Robert, and Guilherme Thompson, "Analysis of an Offloading Scheme for Data Centers in the Framework of Fog Computing," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems* (TOMPECS), vol. 1, no. 4, Sep. 2016.

[9] ETSI GS MEC 003, "Mobile Edge Computing (MEC); Framework and Reference Architecture, V1.1.1" 2016. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/01.01.01_60/gs_MEC003v010101p.pdf

[10] Jocelyne Elias, Bartlomiej Blaszczyszyn, "Optimal Geographic Caching in Cellular Networks with Linear Content Coding," *International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks* (WiOpt 2017), Paris, France, May 2017.

[11] Kenza Hamidouche, Walid Saad, Mérouane Debbah, "Breaking the Economic Barrier of Caching in Cellular Networks: Incentives and Contracts," *IEEE Global Communications Conference* (GLOBECOM 2016), Washington, USA, Dec. 2016.

[12] Ge Ma, Zhi Wang, Miao Zhang, Jiahui Ye, Minghua Chen, and Wenwu Zhu, "Understanding Performance of Edge Content Caching for Mobile Video Streaming," *IEEE Journal on Selected Areas in Communications* (J-SAC), vol. 35, no. 5, May 2017.

[13] ETSI GS MEC 002, "Mobile Edge Computing (MEC) Technical Requirements V1.1.1," 2016. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/MEC/001_099/002/01.01.01_60/gs_MEC002v010101p.pdf

[14] 3GPP TS 32.511, "Telecommunication management; Automatic Neighbour Relation (ANR) management; Concepts and requirements, (Rel. 14)", April 2017.

[15] Pol Blasco, Deniz Gunduz, "Learning-Based Optimization of Cache Content in a Small Cell Base Station," *IEEE International Conference on Communications* (ICC 2014) , Sydney, Australia, June 2014.

[16] Yuyi Mao, Changsheng You, Jun Zhang, Kaibin Huang, and Khaled B. Letaief, "Mobile Edge Computing: Survey and Research Outlook," *Computing Research Repository* (CoRR) abs/1701.01090 (2017).