GoEdge: A Scalable and Stateless Local Breakout Method

Kazuya Okada
ATR
Kyoto, Japan
loppy@atr.jp

Shigeru Kashihara
ATR
Kyoto, Japan
shigeru@atr.jp

Nao Kawanishi
ATR
Kyoto, Japan
river24@atr.jp

Nobuo Suzuki
ATR
Kyoto, Japan
nu-suzuki@atr.jp

Keizo Sugiyama
ATR
Kyoto, Japan
ke-sugiyama@atr.jp

Youki Kadobayashi
NAIST
Nara, Japan
youki-k@is.naist.jp

ABSTRACT

5G mobile communication networks (5G) are expected to provide a wide range of new emerging services through flexible communication with high data rates and low network latencies. In 5G, Multi-access Edge Computing (MEC) is one of the key technologies contributing to performance improvement for various use cases. MEC servers are deployed at edges of mobile networks, e.g., on a base station, and enable provision of new services to users. Unlike the conventional end-to-end communication, MEC needs to allow flexible communication between a client and a server. In this paper, to feasible the flexible communication, we propose an implementation design of a scalable and stateless local breakout method for MEC. It presents a function of scalable traffic steering between a client and a server. Also, a preliminary evaluation shows the feasibility of our proposed mechanism.

CCS CONCEPTS

• **Networks → Network design principles**; **Mobile networks**;

KEYWORDS

Multi-Access Edge Computing (MEC), Local Breakout, Routing, IP Anycast, DNS

ACM Reference Format:
Kazuya Okada, Shigeru Kashihara, Nao Kawanishi, Nobuo Suzuki, Keizo Sugiyama, and Youki Kadobayashi. 2018. GoEdge: A Scalable and Stateless Local Breakout Method. In *TOPIC '18: Theory and Practice for Integrated Cloud, Fog and Edge Computing Paradigms 2018 Workshop, July 27, 2018, Egham, United Kingdom.* ACM, New York, NY, USA, Article 4, 6 pages. https://doi.org/10.1145/3229774.3229776

1 INTRODUCTION

Toward the goal of 5G network service, research and development has accelerated worldwide. Table 1 shows projects and organizations related to 5G. They are actively delivering white papers, recommendations, specifications, and so on. In [10], ITU-R asserts that 5G should support emerging new use cases, including applications requiring very high data rate communications, a large number of connected devices, and ultra-low latency and high-reliability applications. Also, to make the use cases feasible, 5G has to overcome technical issues including high data rates, reduced latency, low energy, high scalability, high connectivity, and high security [2]. In particular, reduced latency, i.e., ultra-low latency, is the most crucial factor in 5G, to satisfy the requirement of around 1 ms, for emerging applications such as autonomous driving, augmented reality, virtual reality, and tactile internet.

Table 1: 5G Projects and Organizations

Country	Projects and organizations
International	ETSI, ITU-R/ITU-T, 3GPP, NGMN, 5GAA
Europe	METIS, 5GNow, MiWaveS, iJOIN, 5GPPP
US	5G Americas
China	IMT-2020 (5G) Promotion Group, National Science and Technology Major Project
Korea	5G Forum
Japan	The Fifth Generation Mobile Communications Promotion Forum (5GMF)

Edge computing technology has attracted a great deal of attention as one of the approaches to achieving ultra-low latency. Deploying processing nodes at the edge of networks, edge computing is expected to accomplish various functions like high QoS, distributed processing, bandwidth saving, and battery saving, in addition to ultra-low latency [11]. However, it has not yet been standardized. In existing proposals, Cloudlet [18], Fog Computing [3], and Multi-access Edge Computing (MEC) [7] have been proposed so far as the core technology. While Cloudlet and Fog Computing target the application to general networks, MEC is aimed at enabling services within the radio access network (RAN) as one of the technologies related to 5G. This paper focuses on MEC.

MEC is expected to make versatile attractive services and applications possible. Researchers have been proposing diverse methods for each topic and proving prospective effectiveness via mathematical and simulation approaches so far. The next phase requires proposals and evaluations based on prototype systems in a real environment. However, for the present, since MEC has a new feature to flexibly handle connections, not conventional end-to-end connection, there are few reference models for implementing MEC. In particular, to provide the above diverse services and applications, the implementation design of MEC should be simple and scalable.

In this paper, to feasible the flexible communication for 5G, we propose an implementation design of a scalable and stateless local breakout method. In this approach, applying mature DNS and IP routing manners, MEC possesses scalable local breakout by stateless control. The contribution of the paper is as follows. The paper provides a simple, implementable way to make scalable and stateless local breakout for MEC. It then contributes that not only carriers but also researchers can also have the experimental environment to deploy their proposals as the next step of mathematical and simulation models. That it is, the prototype system illustrates the implementation as a reference model. We also show preliminary evaluations for the feasibility of the proposed method.

2 RELATED WORK

The Industry Specification Group for Multi-access Edge Computing (ISG-MEC) of the European Telecommunications Standard Institute (ETSI) has been conducting the standardization of MEC. At first, the standardization of MEC had been furthered as "Mobile" Edge Computing. However, to expand the application to various networks like Wi-Fi and wired networks, MEC has been used as the abbreviation for "Multi-access" Edge Computing since September 2016 [15]. Also, although the specification of ETSI, i.e., ETSI GS MEC-IEG 005 [5], states the importance of the MEC proof of concept (PoC) framework, no specifications have described concrete information for MEC implementation and experiments as of this writing.

Many researchers have been studying MEC technologies, and various survey papers that focus on architecture, security, and so on have been published to provide the research trends surrounding MEC [1, 13, 14, 17, 19, 21]. While a variety of technologies and functions are required to achieve MEC, there are few implementation models.

We here introduce three related works as implementation models related to traffic steering of MEC. Reference [16] designed, implemented and evaluated MEC with a cooperative caching system to reduce the network latency in Content Delivery Networks (CDN). The article applies Apache ZooKeeper Service[1] which is a centralized service for enabling highly reliable distributed coordination to provide efficient content caching through cooperation caching among MECs. As a result, it contributes to reduction of network latency and traffic overload.

Also, to offload mobile traffic to the edge network, reference [12] proposed a virtual architecture of a MEC server and functional blocks of a Virtual Machine (VM) for a local breakout. In their testbed environment, the local breakout VM in the MEC server manages traffic control based on the information of IP address and port number as Software Defined Networking (SDN).

[1]http://zookeeper.apache.org

Reference [9] designed, implemented, and evaluated the MEC platform utilizing SDN for fulfilling low latency. It constructed a MEC environment that targeted video-streaming application, and OpenFlow switch achieved traffic steering to the data center and the MEC server. It contributes to providing an implementation model based on SDN. In the method, an OpenFlow switch located in a base station properly divides user traffic between the Internet and MEC based on packet header information, i.e., src/dst IP addresses, src/dst port number, protocol type, and so on. However, a service generally has multiple IP addresses, and a single provider also supplies multiple different types of services. For instance, a provider supplies various services such as a search engine, webmail, file sharing, online storage, and so on. In this case, as OpenFlow identifies traffic flow based on the above information, it needs to make different rules for each service. As a result, the rule swells, and it then consumes many resources of the OpenFlow table on the switches. Also, when for certain reasons, a service provider may change the assignment of IP addresses without notice, a network operator has to maintain the OpenFlow rules. Furthermore, a massive number of OpenFlow switches are required to be deployed on the edge of a network such as mobile base stations. To alleviate the above problems, the next section proposes a local breakout method based on IP anycast manner.

3 GoEdge

This section proposes a scalable and stateless local breakout mechanism for MEC by using DNS and IP anycast called GoEdge. Sections 3.1 defines requirements of local breakout for MEC. Section 3.2 explains design and implementation of GoEdge with commodity protocol and software. Also, Section 3.3 illustrates scale-out and scale-in design of GoEdge.

3.1 Requirements for Local Breakout

The paper focuses on a local breakout method based on IP anycast manner. Figure 1 shows the diagram of reference MEC architecture, which is standardized in ETSI MEC ISG [6]. In the diagram, our target functions, i.e., *Traffic rule control, DNS handling, Data plane and ME app*, are surrounded by the red line.

This section discusses requirements for achieving a local breakout. The following are three key requirements that a local breakout should satisfy.

Generality: As MEC is expected to adapt to diverse use cases such as content delivery, Augmented Reality (AR) and Virtual Reality (VR), the local breakout for MEC should not depend on any specific use case. That is, independence from any use case is required.

Scalability: Mobile carrier networks comprise much complicated network equipment. At present, the number of base stations, i.e., eNodeB, in a single carrier has reached a few hundred thousand for covering a country or region [4]. In 5G, as a new radio access network (RAN) plans to employ extremely high frequency (ETH) bands for providing more high-speed wireless access to users, a single base station will cover only dozens of meters. As a result, mobile carriers will need to deploy more base stations compared with existing networks. Accordingly, MEC local breakout has to

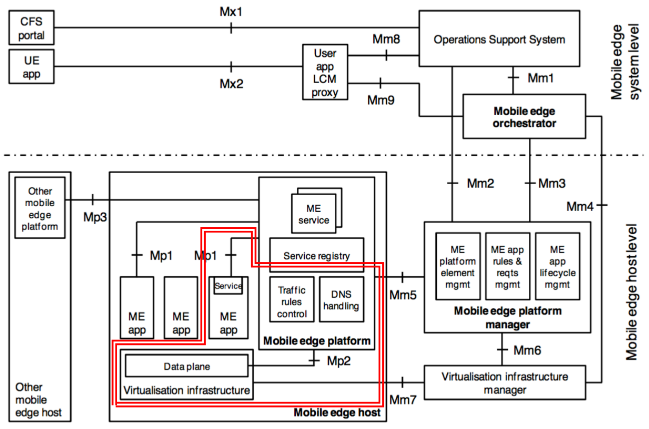

Figure 1: MEC reference architecture (standardized in ETSI MEC ISG)

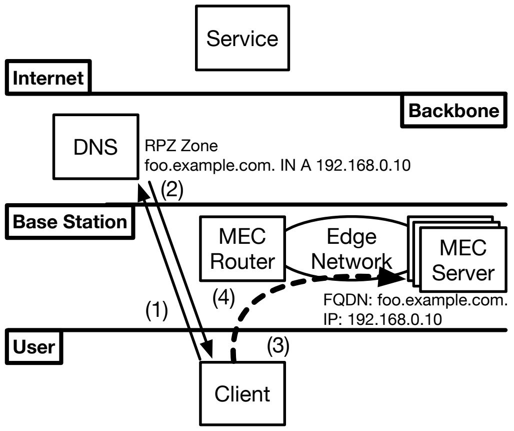

Figure 2: Structure and Communication Procedure of GoEdge

```
foo.example.com. IN A 192.168.0.10
bar.example.com. IN A 192.168.0.11
foo.example.net. IN A 192.168.0.12
```

Figure 3: An example of an RPZ zone

be flexible in dynamic base station deployment on mobile carrier networks.

No modification for client: In ETSI ISG-MEC, besides 5G, MEC assumes to be deployed to wired and Wi-Fi networks. Thus, a local breakout has to properly work on general ISP networks too. Also, if a specific protocol or application is employed for utilizing a local breakout, as a client needs them to take advantage of MEC-capable networks, the spread of 5G service may take much time. To smoothly deploy MEC, it is required for a local breakout to have no modification for protocol and application to a client.

3.2 Design and Implementation

The main idea of GoEdge satisfying the above requirements is to utilize the combination of DNS and IP anycast for a local break. This section explains the detailed design and implementation of GoEdge.

In our proposal, we utilize IP anycast manner. IP anycast is one of the communication types on the Internet. In unicast communication, a packet is forwarded to a specific destination host based on the destination IP address in the packet header. On the other hand, in IP anycast, as multiple hosts can share the same IP address, a packet is routed to the closest host to the user. Also, IP anycast has the characteristic of easily making the distribution of hosts. That is, IP anycast can provide the proximity for the communication, and it then contributes to reducing the latency. Also, since IP anycast is applied to Root DNS system operations, the root DNS instances are widely distributed and achieve load balancing all over the world by using the IP anycast. By this advantage, even if one site of the IP anycast members becomes unavailable due to a problem, the traffic is automatically rerouted to the secondary closest site. Thus, IP anycast is utilized for global load balancing on the IP layer without any modification to clients and servers[8].

The main idea of GoEdge is to control traffic on a MEC router located within a base station. GoEdge comprises an edge network and a DNS resolver as illustrated in Figure 2. The edge network also consists of a MEC router and MEC servers, and then MEC applications are hosted on MEC servers. A MEC router holds general routing information for the backbone network and the Internet. Also, a route of a link that connects to a MEC router typically has highest priority on the routing table. Therefore, GoEdge utilizes the mechanism for localizing the traffic on an edge network. GoEdge first assigns dedicated IP prefixes for an edge network, but the MEC router does not distribute the routes to the backbone network. That is, the IP prefixes are shared among MEC routers that are unreachable from other edge networks. If a client sends traffic to the IP prefixes on an edge network, the traffic is forwarded to each edge network and does not leak to the backbone network. Therefore, GoEdge can localize traffic by classifying it based on routing information that a MEC router holds.

To take advantage of traffic localization, it is necessary to notify IP addresses of MEC applications on a MEC server to clients. For this, GoEdge employs DNS resolver with Response Policy Zone (RPZ) [20]. The RPZ replies an arbitrary IP address for an Fully Qualified Domain Name (FQDN) in a DNS query without recursive name resolution. Although it was originally designed for blocking malicious domains, i.e., a blacklist, on DNS resolvers by controlling DNS replies, GoEdge applies the mechanism for notifying IP addresses of MEC applications to clients. For instance, if an FQDN, e.g., foo.example.com., means a MEC application, as the DNS server holds the mapping of FQDN and IP address on the RPZ zone file, it can reply the IP address of the MEC application to the client. Figure 3 illustrates an example of an RPZ zone. Each entry shows an FQDN of a MEC application and an IP address that is assigned to the MEC application. Also, as for other traffic that is not routed to a MEC application, a DNS server replies an IP address on the Internet as a general resolver.

Figure 2 also depicts the communication procedure of GoEdge. First, in (1), a client sends a DNS query to the DNS resolver for

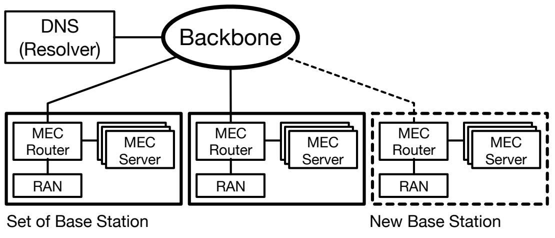

Figure 4: Scale-out and Scale-in Design of GoEdge

resolving the destination name. Then, if the FQDN is included on the RPZ file, the DNS server replies the IP address for it to the client in (2). In (3), after receiving the reply, the client starts the communication to the MEC application. In (4), the MEC router forwards the packets to the MEC application, and it does not forward them to the Internet. If the destination means an IP address on the Internet, the traffic is regularly forwarded based on their routing table. Through the above procedure, GoEdge can control user traffic and properly lead them to MEC applications.

GoEdge shows a new use case of a combination of IP anycast and DNS in local networks such as mobile networks. GoEdge leverages successful deployment of IP anycast-based CDNs as mentioned in [8]. As we described above, we apply the simple and scalable traffic steering method to achieve local breakout for MEC. Typically, traffic control including traffic steering in carrier networks needs manual configurations for routers and maintains the routing configurations. In GoEdge, the traffic steering rules are pre-configured at the routers as static routes. Accordingly, the network operators do not need additional configurations and management for routers. GoEdge effectively reduces the control and management loads in the networks by using a combination of IP anycast and DNS.

3.3 Scale-out and Scale-in Design of GoEdge

Mobile and wired network operators will deploy new MEC on their facility such as a base station, and they may also occasionally remove the facility. Accordingly, scale-out and scale-in design for flexible operation of MEC is required. For this, GoEdge provides the following way as depicted in Figure 4. If a new base station with MEC is deployed, a mobile carrier can deploy the new base station in the following manner. The new base station utilizes the same IP prefixes for a MEC network that have already been used in other base stations. Additionally, the base station connects to the backbone network of the operator. After this deployment, MEC traffic coming from clients under the base station is forwarded to the new MEC facility. During the deployment, the operator does not need any changes on a DNS server because there are no specific configurations and communications between MEC routers and the DNS server.

4 EVALUATION

This section investigates the feasibility and performance of GoEdge. Section 4.1 first explains the experimental environment for the evaluation. Section 4.2 then evaluates the feasibility of GoEdge in a content cache scenario. Sections 4.3 and 4.4 present results for memory consumed and lookup latency, respectively.

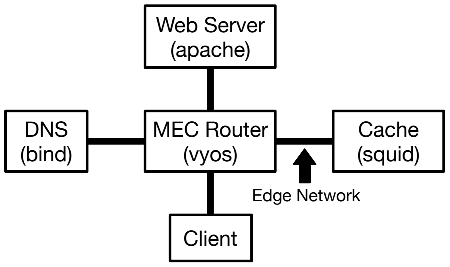

Figure 5: Experimental Topology

4.1 Experimental Setup

We here investigate the feasibility and performance of GoEdge in the experimental environment consisting of conventional DNS software as a DNS resolver (*bind9.10.3-P4*), an open source software router as a MEC router (*VyOS1.1.8*), and a container platform (*Docker 18.03.0~ce-0*) for MEC application host. Figure 5 depicts the edge network topology employed in our evaluation.

The environment is deployed in virtual machines on top of KVM hypervisors. The KVM hypervisors run on two servers with Intel®Xeon®CPU E3-1230 v5 @ 3.40GHz (6 cores) and 8GB physical memory.

4.2 Feasibility Test

This section investigates the behavior of GoEdge in a content cache scenario in order to prove feasibility. In this scenario (see Figure 5), the web server hosts a content file of 100 MB that was generated by *dd* command. The MEC server works as a content cache server, and it is deployed by *Squid 3.3.8*[2], which is an open source proxy software as a Docker instance. Therefore, in the scenario, we measure the effectiveness by the installation of a MEC server. To evaluate the fundamental performance, the delay from the MEC router to the Web server is changed between 0 and 100 msec by *tc* command. In the experiment, the client downloads the content file from the Web server by *wget* command 100 times per delay.

Figure 6 presents a fundamental result for the downloading time from the Web server and the MEC server. In the graph, the x-axis denotes the delay from the MEC router to the Web server, while the y-axis indicates downloading time for a 100MB piece of content. Note that "cache" on the y-axis means the result for a MEC server. As shown in the result of the Web server, the downloading time is proportionally increasing with the delay. On the other hand, when GoEdge is activated, the client can download the content file for the minimum time. The result proves that GoEdge can properly identify user traffic and lead them to the MEC server.

4.3 Memory Consumed

This section evaluates how the size of the RPZ zone impacts memory usage of a DNS server. With the increase of MEC applications, the number of DNS records for the instances will also be increasing in the RPZ zone file on the DNS server. Accordingly, the DNS server may be expected to require large memory.

[2]Squid: http://www.squid-cache.org/

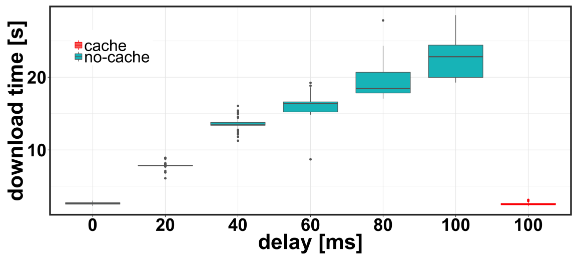

Figure 6: Downloading time for Web server and MEC server

Table 2: Memory Consumed on DNS server with Different Zone Size

#RPZ entries	Memory [KB]
1,000	24,892
10,000	28,028
100,000	63,072
1,000,000	238,520

Table 2 shows the measurement results of the consumed memory size by DNS software for the size of the RPZ zone file. The measurement employs DNS zone lists that are generated from the Alexa top 1 million website domains[3]. Form the result, we can see that the size of the memory consumed for the DNS process is only about 240 MB even if 1 million domains listed in the RPZ zone file are employed. Therefore, GoEdge can handle the services without high-end servers for the increase in MEC applications.

4.4 Lookup Performance

This section measures lookup latency from a client for different sizes of RPZ zone. The measurement employs four record sizes: 100, 1,000, 10,000, and 100,000. In the RPZ file, FQDNs are randomly chosen from the Alexa top 1 million sites. In the experiment, a client makes lookup 1,000 times by the *dig* command in each record size.

Figure 7 shows the changes of the latency for the different size of RPZ record. The x-axis denotes the record size while the y-axis indicates the latency for lookup in log-scale. In the case where the record size is 1,000, the average for lookup latency is around 100 ms. The latency is quite long for name resolution. Therefore, improvement of the latency is required to provide more than 100 services on MEC. Even if the DNS resolver cannot handle DNS queries due to increasing clients on the network, the servers easily scale-out by adding other DNS resolvers as part of usual operations.

5 DISCUSSION

This section discusses adaptability, scalability, and limitations of GoEdge. We also discuss deployment issues of GoEdge for real networks.

5.1 Scalability

In 5G, as operators will continuously deploy new base stations for improving communication quality and their coverage to provide

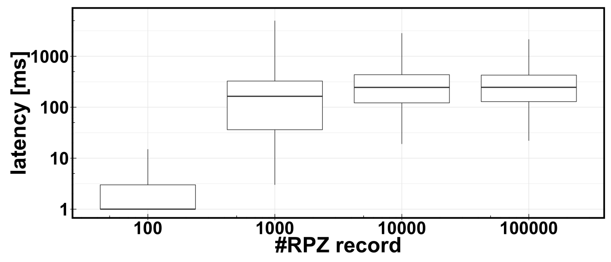

Figure 7: Lookup Latency for Different Sizes of RPZ Zone

various mobile services, scalability of MEC is a significant issue. As explained in Section 3.3, GoEdge can scale-out and scale-in service components relatively easily compared with the existing method. Since the process of installing a new MEC instance is that GoEdge adds an RPZ record in the DNS RPZ zone file, any change to routers and clients on the network edge is not required except that. Accordingly, GoEdge flexibly provides scale-out and scale-in network size.

5.2 IP address assigned in edge networks

This section discusses the dedicated IP prefix of the edge network located within a base station. The experiment in Section 4 assigned a private IPv4 prefix (10.0.0.0/24) to the edge network. However, such private IPv4 addresses sometimes may conflict with the internal address range used on client devices. For instance, it is commonly employed for enterprise VPN networks. When a client device connects to an internal network via a VPN session and the same private IP address range with the edge network is assigned, the traffic to MEC instances forwards to the VPN network on the client. To avoid such a problem, GoEdge can use global IPv4/IPv6 addresses which are not used in the backbone network instead of private IPv4 addresses. Also, as the global IPv4 address space managed by IANA was exhausted a few years ago, the use of IPv6 addresses is a valid measure for edge networks.

Another problem is a hardcoded IP address in an application on a client. GoEdge handles traffic by the DNS mechanism, but in this case, DNS cannot resolve the hardcoded IP addresses. Therefore, GoEdge cannot lead traffic to a MEC instance.

5.3 Load Balancing

When a large number of clients concentrate on a base station, MEC may not be able to provide sufficient performance to applications. At present, GoEdge does not support a load-balancing function on an edge. Since IP address by DNS response cannot be changed depending on a load of MEC instances, it is necessary to introduce a load-balancing mechanism for MEC instances. As one solution for the problem, installing a load balancer in front of the edge instances and manipulating the MEC instances may be effective.

5.4 Mobility

As ISG-MEC in ETSI assumes MEC is also adapted to wired networks in addition to mobile networks, MEC has a potential to be deployed in various network. Since a client freely moves, it may be

[3]http://www.alexa.com/

change its location and connect to multiple networks such as 5G, Wi-Fi, and wired networks at the same time. If the connected MEC instance is changed by the movement, GoEdge needs to forward the user traffic from the previous instance to the new one. Therefore, it is required to migrate the connections and synchronize the process state among the MEC instances.

5.5 Comparison with previous work

The GoEdge has the following advantages over the previous work.

Low Cost: Although other approaches including SDN-based local breakout [9] need dedicated hardware and controllers, GoEdge can employ entry model L3 switches or routers and open-source DNS software. Thus, operators and engineers can reuse their matured IP networking knowledge for GoEdge operations. Therefore, GoEdge contributes to reducing OPEX of the MEC in mobile operators efficiently. Also, in laboratories, researchers can conduct original experiments of his/her idea on a simple network using GoEdge.

Stateless and Scalable: GoEdge has stateless control and management as other advantages. As GoEdge entities are entirely stateless for individual user communication, the DNS server and router are independent in GoEdge. Also, packets forwarding on routers do not keep any session information of user communication. Thus, this stateless architecture contributes to the scalability of MEC in mobile networks, as explained in Section 3.3.

6 CONCLUSION

This paper presented the design and implementation of the scalable and stateless local breakout for MEC, GoEdge. GoEdge employs matured DNS and IP anycast manners for a local breakout of MEC. It contributes simple and scalable network management compared with existing approaches because it does not need to manage the state of individual flows and sessions at the network side. Also, the experimental result showed that as GoEdge can adequately control traffic, it reduces the latency.

ACKNOWLEDGMENTS

This research was conducted under an R&D contract for radio resource enhancement, organized by the Ministry of Internal Affairs and Communications, Japan.

REFERENCES

[1] N Abbas, Y Zhang, A Taherkordi, and T Skeie. 2018. Mobile Edge Computing: A Survey. *IEEE Internet of Things Journal* 5, 1 (2018), 450–465.

[2] Ian F. Akyildiz, Shuai Nie, Shih-Chun Lin, and Manoj Chandrasekaran. 2016. 5G roadmap: 10 key enabling technologies. *Computer Networks* 106 (sep 2016), 17–48. https://www.sciencedirect.com/science/article/pii/S1389128616301918

[3] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. 2012. Fog Computing and Its Role in the Internet of Things. *Proceedings of the first edition of the MCC workshop on Mobile cloud computing* (2012), 13–16. arXiv:arXiv:1502.01815v3

[4] NTT DoCoMo. 2018. NTT DOCOMO FACTBOOK. (January 2018). https://www.nttdocomo.co.jp/english/info/media_center/pdf/factbook.pdf

[5] ETSI. 2015. Mobile-Edge Computing (MEC); Proof of Concept Framework. (2015). http://www.etsi.org/deliver/etsi

[6] ETSI. 2016. Mobile Edge Computing (MEC); Framework and Reference Architecture. (2016). http://www.etsi.org/deliver/etsi

[7] ETSI. 2016. Mobile Edge Computing (MEC); Technical Requirements. (2016). http://www.etsi.org/deliver/etsi

[8] Ashley Flavel, Pradeepkumar Mani, David A Maltz, Nick Holt, Jie Liu, Yingying Chen, and Oleg Surmachev. 2015. FastRoute: A Scalable Load-aware Anycast Routing Architecture for Modern CDNs. In *Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation (NSDI'15)*. USENIX Association, Berkeley, CA, USA, 381–394. http://dl.acm.org/citation.cfm?id=2789770.2789797

[9] Anta Huang and Navid Nikaein. 2017. Demo: LL-MEC A SDN-based MEC Platform. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking (MobiCom '17)*. ACM, New York, NY, USA, 483–485.

[10] ITU-R. 2015. *IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond.* Technical Report.

[11] G Klas. 2017. Edge Computing and the Role of Cellular Networks. *Computer* 50, 10 (2017), 40–49.

[12] S Q Lee and J u. Kim. 2016. Local breakout of mobile access network traffic by mobile edge computing. In *2016 International Conference on Information and Communication Technology Convergence (ICTC)*. 741–743.

[13] P Mach and Z Becvar. 2017. Mobile Edge Computing: A Survey on Architecture and Computation Offloading. *IEEE Communications Surveys & Tutorials* 19, 3 (2017), 1628–1656.

[14] Y Mao, C You, J Zhang, K Huang, and K B Letaief. 2017. A Survey on Mobile Edge Computing: The Communication Perspective. *IEEE Communications Surveys & Tutorials* 19, 4 (2017), 2322–2358.

[15] Iain Morris. 2016. ETSI Drops 'Mobile' From MEC. (2016). http://www.lightreading.com/mobile/mec-(mobile-edge-computing)/etsi-drops-mobile-from-mec/d/d-id/726273

[16] Morteza Neishaboori. 2015. *Implementation and Evaluation of Mobile-Edge Computing Cooperative Caching.* G2 Pro gradu, diplomityö. Aalto University. http://urn.fi/URN:NBN:fi:aalto-201509184436

[17] Rodrigo Roman, Javier Lopez, and Masahiro Mambo. 2018. Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges. *Future Generation Computer Systems* 78 (jan 2018), 680–698. https://www.sciencedirect.com/science/article/pii/S0167739X16305635

[18] M Satyanarayanan, P Bahl, R Caceres, and N Davies. 2009. The Case for VM-Based Cloudlets in Mobile Computing. *IEEE Pervasive Computing* 8, 4 (2009), 14–23.

[19] T Taleb, K Samdanis, B Mada, H Flinck, S Dutta, and D Sabella. 2017. On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration. *IEEE Communications Surveys & Tutorials* 19, 3 (2017), 1657–1681.

[20] Paul Vixie and Vernon Schryver. 2016. *Response Policy Zones.* Internet-Draft draft-vixie-dns-rpz-00. IETF Secretariat. http://www.ietf.org/internet-drafts/draft-vixie-dns-rpz-00.txt http://www.ietf.org/internet-drafts/draft-vixie-dns-rpz-00.txt.

[21] Shuo Wang, Xing Zhang, Yan Zhang, Lin Wang, Juwo Yang, and Wenbo Wang. 2017. A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications. *IEEE Access* 5 (2017), 6757–6779. arXiv:1703.10750