Master's Thesis

Title

Experimental Evaluation of Accommodation Methods of M2M/IoT Terminals in Mobile Core Networks

Supervisor

Professor Morito Matsuoka

Author

Masaki Ueno

February 6th, 2019

Graduate School of Information Science and Technology

Osaka University

Master's Thesis

Experimental Evaluation of Accommodation Methods
of M2M/IoT Terminals in Mobile Core Networks

Masaki Ueno

Abstract

Due to recent and rapid increase in cellular network traffic, as well as the rapid growth of M2M/IoT services, handling congestion in Long Term Evolution and 5th generation cellular networks has become a critical issue. Since some M2M/IoT terminals have different communication characteristics from traditional rich user terminals, control plane congestion becomes serious when accommodating massive number of those terminals into cellular networks. Standardization organizations, including 3rd Generation Partnership Project, publicizes several standards for accommodating M2M/IoT communication. Moreover, various existing studies have argued that introducing virtualization technologies such as software defined network and network function virtualization into mobile core networks is a possible solution to deal with congestion issues. However, most of performance evaluations shown in those studies are mainly based on mathematical analysis and simulation experiments. Experimental evaluations with implementation on real network environment are necessary in order to examine the actual performance of those methods.

In this thesis, we show the experimental evaluation results of the performance of a mobile core network to assess the impact of massive accesses from M2M/IoT terminals. First, we constructed the experimental environment based on open-source implementation of mobile core networks and emulated user terminals/base stations. We then conducted experiments on simultaneous attach requests from multiple user terminals and measured the processing delay at each mobile core node. The results of our evaluations clarified that the Mobility Management Entity (MME) becomes a bottleneck of attach procedure as shown in various existing works. When an MME node is operated on a virtual machine with 1 GHz / 1 core CPU, simultaneous attach requests from 128 user terminals increases the bearer establishment time by up to around 450%. We also provide possible solutions for avoiding the impact of massive accesses from M2M/IoT terminals: resource enhancement on an MME and distributing attach request from multiple user terminals on purpose.

Keywords

Mobile Core Network

M2M/IoT Communications

Long Term Evolution (LTE)

5G Cellular Network

virtualized Evolved Packet Core (vEPC)

Contents

List of Figures

List of Tables

1 Introduction

Handling congestion in cellular networks including Long Term Evolution (LTE) [1] and 5th generation mobile communication system [2] has become a critical issue due to recent and rapid increase in users of mobile terminals and their functional enhancement. In addition, the concept of attaching M2M/IoT terminals to cellular networks has attracted a lot of attention. Because of this trend, a more urgent problem is the increasing traffic load on mobile core networks, especially on the control plane.

There are some types of M2M/IoT terminals which have different communication characteristics from traditional rich user terminals: an enormous number of terminals perform periodical communication with a small amount of data packets. Low Power, Wide Area (LPWA) [3] networks have been proposed for accommodation of such terminals and realize low power consumption with wide coverage. Long Range WAN (LoRaWAN™) [4] proposed by the LoRa Alliance™, SIGFOX [5] proposed by SIGFOX S.A., and Wi-Fi HaLow™ [6] proposed by the Wi-Fi Alliance®, are major standardized LPWA communication protocols.

One of shortcomings of LPWA is that a new network infrastructure must be constructed for deployment of services. Conversely, an obvious advantage exists in using cellular radio networks is their effective utilization of existing infrastructure. Moreover, it has been noted that it is difficult for non-cellular LPWA, including LoRaWAN to provide proper Quality of Service (QoS) for each M2M/IoT connection, and the data transfer speeds and latency are inferior to cellular radio [7]. Therefore, using the cellular radio network in some cases depending on M2M/IoT applications would be preferable especially for rapidly deploying services with wide coverage.

On the other hand, there are concerns about congestion due to M2M/IoT communication characteristics when accommodating them into cellular networks. For this reason, standardization organizations, including the 3rd Generation Partnership Project (3GPP) publicizes several cellular-based access technologies of accommodating M2M/IoT communication such as Narrow Band IoT (NB-IoT) [8] and enhanced Machine Type Communication (eMTC) [9]. Moreover, various existing works [10–13] have argued that virtualization technologies such as Software Defined Network (SDN) and Network Function Virtualization (NFV) are possible solutions for improving network capacity of mobile core networks.

Our research group focused on mobile core network architecture for accommodating M2M/IoT

terminals [14, 15]. In [15], we conducted mathematical evaluations of models of mobile core network architecture considering the bursty access from massive M2M/IoT terminals, and we clarified the effect of server virtualization, optimal resource allocation, and C/U plane separation. In the evaluation, the processing loads of signaling messages were determined by a simple queueing model and the number of statements obtained by OpenAirInterface [16], an implementation of LTE and an Evolved Packet Core (EPC) network written in C language. However, the actual signaling processing load does not always correlate with the number of statements of implementation codes, because the actual signaling processing is performed by execution codes generated after compiling implementation codes, and counting the number of statements does not take the behavior of conditional branches into account. Therefore, observing the signaling processing load on the real system is required, in order to examine the C/U plane separation and resource allocation of mobile core network nodes.

In this thesis, we show the experimental evaluation results of the performance of a mobile core network to assess the impact of massive accesses from M2M/IoT terminals. First, we construct the network system for experimental evaluations, based on open-source implementation of mobile core networks and emulated user terminals and radio base stations. We also build up massive number of emulated user terminals and radio base stations on the public cloud computing platform. Then, we conduct experiments of simultaneous access from at most 128 user terminals. We evaluate the processing delay of the attach procedure at each mobile core node, and discuss how the attach requests from multiple user terminals affects the performance of the mobile core network nodes.

The remainder of this thesis is organized as follows. Section 2 shows existing works related on this thesis. Section 3 explains the mobile core network and the signaling flow for attach procedure that are utilized in our experimental evaluations. Section 4 describes details of our experiment, including network configurations, node placement, parameters, experiment procedure and measurement method. Section 5 shows the results of the experiments and discussions. Section 6 concludes this thesis with a summary and statements of future work.

2 Related Work

Various methods have been proposed to date for the improvement of capacity of M2M/IoT communications in cellular networks.

In [17], the authors presented two design of LTE Evolved Packet Core (EPC) architectures, one of which is based on Software Defined Network (SDN), and the other is based on Network Function Virtualization (NFV). They also provided the performance comparison of two EPC implementation on their LTE testbed. The results showed that the SDN approach is preferable for handling large amount of user data traffic because SDN switches are optimized for data forwarding. On the other hand, the SDN approach have its bottleneck on the communication between SDN switches and SDN controllers, thus NFV approach is adequate for handling extreme signaling processing load. However, the implementation of their EPC is not fully compliant to standards such as the utilized transport protocol for communications between radio access network and Mobility Management Entity (MME). Also, the fairness of comparison between SDN and NFV approach is not ensured because of the difference of implementation.

Study [18] proposed an analytical model based on open queueing network to model Virtualized Network Functions (VNFs) with several components, and chains of VNFs. The authors of [19] introduced an adaptive mechanism for the user plane virtualization of the LTE Packet Data Network Gateway (P-GW). The authors applied SDN and NFV concepts to their proposed mechanism so that it can be adaptive to workload changes. From the evaluation results, the number of P-GW instance has changed in response to the number of user equipments, and the load balancing has correctly taken effect by their proposed mechanism. However, the performance of proposed architecture in these studies are only evaluated by numerical evaluation and network simulator. Moreover, the number of signaling packets used in [18] is too small compared with the standards of LTE architecture.

In study [20], the authors compared between the performance of EPC running on KVM hypervisor and Docker [21]. The evaluation results showed that Docker is feasible because of its low virtualization overhead. However, as well as [19], the authors only focused on the data plane of LTE network, that is, how the proposed architecture handle control plane signaling, and the effects of those control plane signaling on the performance is not provided.

Study [22] explored 2 approaches of reducing the latency in LTE networks using SDN, NFV

and fog computing. A fog gateway and a General Packet Radio Service Tunneling Protocol (GTP) gateway determines and forwards user data to fog services or external services. The authors characterized both approaches by the placement of the function of GTP encapsulation and decapsulation. The results of their experimental evaluations on the LTE testbed confirmed that both approaches could reduce the latency of LTE accesses. However, there is a assumption that bearers, dedicated data path for user terminals, are established in advance. Since load on control plane generated by signaling messages for establishing bearers is not ignorable in terms of accommodating M2M/IoT terminals, the effect of control plane signaling must be evaluated. Additionally, resources given to each EPC node is not provided.

On the other hand, performance evaluation we conducted is based on LTE/EPC standards, and the amount of resource given to each EPC node is accurately provided.

3 Mobile Core Network

3.1 Network Model

Figure 1 depicts the model of a mobile core network that includes EPC nodes, interfaces between nodes, and bearers established when UE starts data transmission. The nodes have the following functions.

- **User Equipment (UE):** User terminals, including smartphones, tablets, and M2M/IoT terminals.
- **evolved Node Base (eNodeB):** Radio base stations that exchange control messages and data packets with UEs through radio channels. eNodeBs also exchange data packets with the S-GW and control messages with the MME.
- **Mobility Management Entity (MME):** The node that performs the core of the signaling processing, such as authentication of UEs, handling UEs' handover in wireless networks, and bearer setting for data-plane packet transmission between UEs and external IP networks.
- **Home Subscriber Server (HSS):** The database node that manages user specific information, such as the contract information of each user, data for authentication, and the location data of each UE.
- **Serving Gateway (S-GW):** The node that relays IP packets between UEs and the P-GW according to control from MME. It also performs as an anchor point when UEs move between eNodeBs.
- **Packet Data Network Gateway (P-GW):** The node that exchanges IP packets with external IP networks.

Each node is connected by the following logical interfaces built on an IP network.

- **S1-C (S1-MME):** The control plane interface that connects the eNodeB and the MME to exchange control messages between UEs and the MME through the eNodeB.
- **S1-U:** The data plane interface that connects the eNodeB and the S-GW to exchange IP packets between UEs and the S-GW through the eNodeB.

- **S6-a:** The control plane interface that connects the MME and the HSS to exchange control messages such as authentication data and location data.
- **S11:** The control plane interface that connects the MME and the S-GW to exchange control messages including bearer information of each UE.
- **S5/S8:** The data plane interface that connects the S-GW and the P-GW to exchange user data.
- **SGi:** The data plane interface that connects the P-GW and the external IP network to exchange IP packets between UEs and the external IP network.

3.2 Signaling Flow for Attach Procedure

When a UE connects to the mobile network, three data-plane bearers are established before starting data transmission: a radio bearer between the UE and the eNodeB, an S1 bearer between the eNodeB and the S-GW, and an S5/S8 bearer between the S-GW and the P-GW. Figure 2 shows the signaling flow to establish the bearers when a UE attaches to the mobile core network before data transmission. Several abbreviations are used in the figure; req., res. ans. and cmp. mean request, response, answer and complete, respectively. Ctxt stands for Context and Msg. stands for Message. Additionally, UE and eNodeB are depicted as a single node (UE+eNodeB), and we omitted signaling messages between UE and eNodeB since we did not evaluate them for this paper.

As shown in Figure 2, many control plane messages are exchanged between mobile core nodes before starting data transmission. Consequently, the load on control plane nodes becomes large when considering that massive M2M/IoT terminals are accommodated into the cellular network and their data transmissions are synchronized due to the application characteristics, even when transmitting small amounts of data per UE. This is because the signaling flow in Figure 2 is required in attach procedure regardless of the data size to be transmitted. Consequently, assessing the performance of the mobile core network against the access concentration is important.

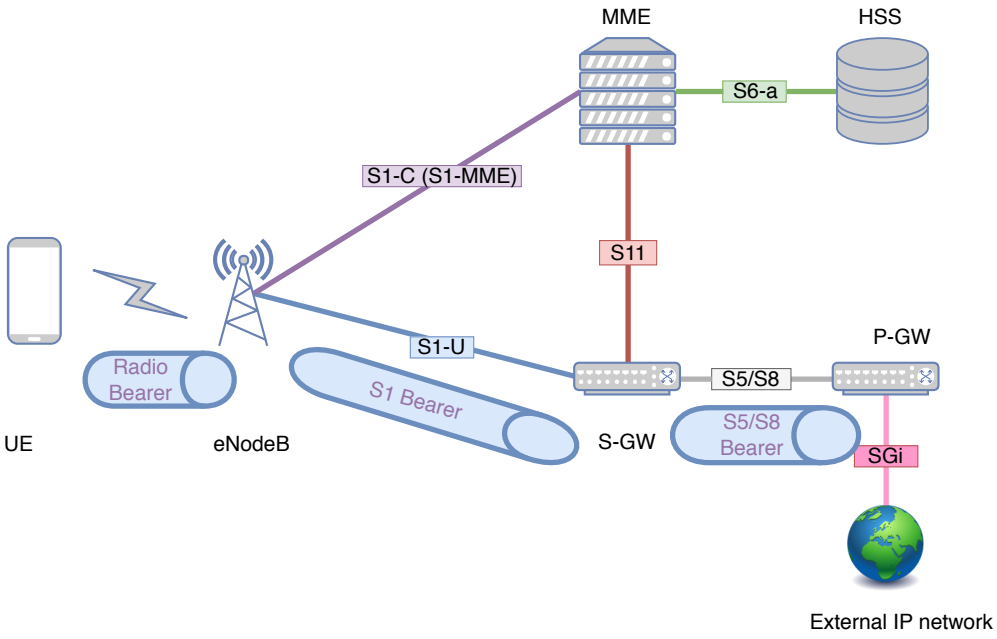


Figure 1: Mobile core network model

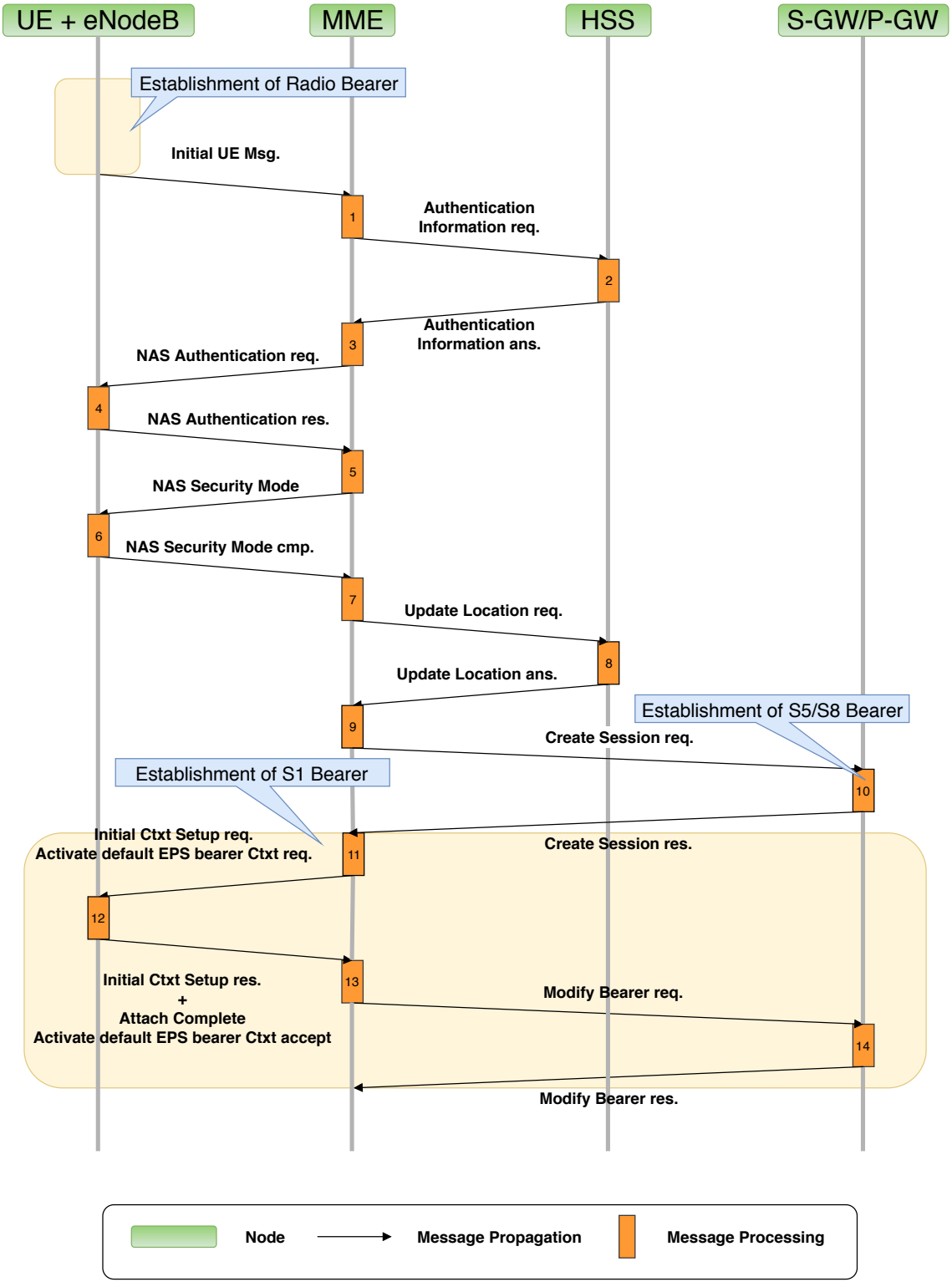


Figure 2: Signaling flow in attach procedure

4 Setup of Experiment

4.1 OpenAirInterface (OAI)

We exploited OAI, an open-source implementation of LTE/EPC networks, to construct an experimental environment including a mobile core network, UEs and eNodeBs. OAI includes components to operate UEs and eNodeBs on either actual equipment or in a simulator called OAISIM, and components to operate EPC nodes on servers. In OAI, an S-GW and a P-GW are implemented as a single node and the S5/S8 interface between the S-GW and the P-GW is realized by the interprocess communications. Therefore, S-GW and P-GW are referred to as "SP-GW" in what follows. Note that an eNodeB exists for each UE because of the current limitations of the OAISIM.

4.2 Network Configuration

In our experiment, OAISIM and EPC nodes are deployed on independent networks. We describe their detailed settings in the following subsections.

4.2.1 EPC Network

Figure 3 depicts the configuration of EPC nodes in the experimental environment. The MME, the HSS, and the SP-GW were installed on separate virtual machines on a single physical host running VMWare ESXi 6.0 update 2. Table 1 shows the specifications of each node and virtualization environment

All LTE/EPC logical interfaces except for SGi belong to independent network segments from the Local Area Network (LAN) of our laboratory in order to avoid impacting LAN traffic on the experimental network. In Figure 3, 192.168.3.0/22 represents the LAN of our laboratory, where the SGi interface was placed. 172.1.0.0/16 and 192.168.4.0/22 are independent network segments for the S1-C/S1-U and S6-a/S11 interfaces, respectively. As we describe in section 4.2.2, since OAISIM are executed on a public cloud computing platform, we have created globally accessible gateway to the EPC network. The gateway is enabled to handle Stream Control Transmission Protocol (SCTP) packets, which is utilized for S1 Application Protocol (S1AP) communications. The gateway forwards SCTP packets from the Internet to EPC nodes, and vice versa.

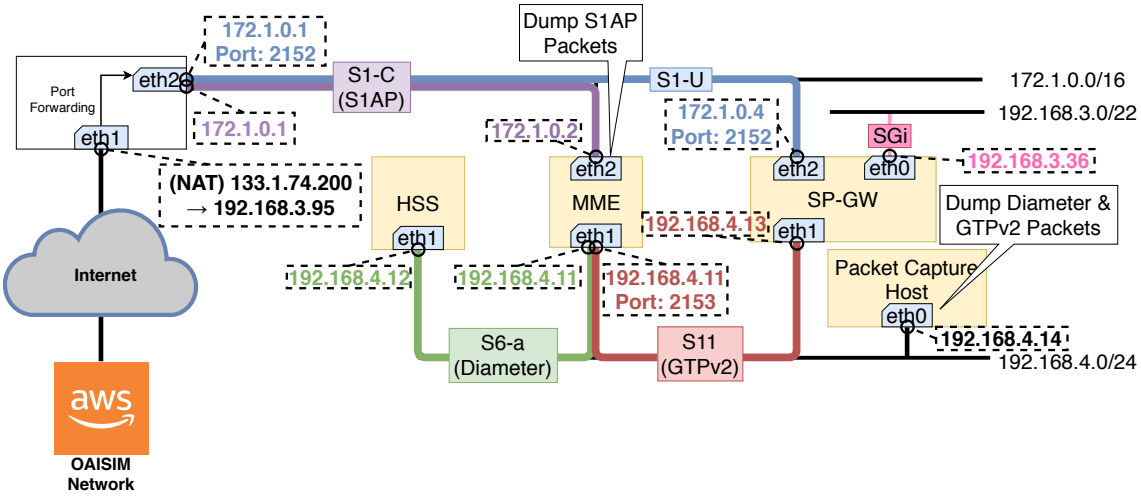


Figure 3: Configuration of EPC network

Table 1: Specifications of EPC nodes and server virtualization environment

Node Name	Operating System	Kernel Version	CPU Clock [GHz]	CPU Core	Memory Size [GB]
MME	Ubuntu 14.04 LTS	3.13.0-24-generic	1	1	1
HSS	Ubuntu 14.04 LTS	3.13.0-24-generic	1	1	4
SP-GW	Ubuntu 14.04 LTS	4.7.5	1	1	4
VM Host	VMWare ESXi 6.0u2	build-3620759	2.40	12	16

4.2.2 OAISIM Network

We used Amazon Web Service (AWS) [23] Elastic Computing Cloud (EC2) as a public cloud computing platform for executing OAISIM. Figure 4 depicts the detailed configuration of OAISIM network. To reduce the propagation delay between OAISIM and our EPC nodes, we created our virtual private cloud (VPC) in Asia Pacific (Tokyo) region (identified as `ap-northeast-1`). We created three subnets in our VPC. One is a public subnet, which is able to allocate both global and private IP addresses to instances. The others are private subnets, which is only able to allocate private IP addresses to instances. Two private subnets are created on a separated availability zones (`apne1-az2` and `apne1-az4`) to avoid lack of resource on each availability zone. L3 connectivity among three subnets is ensured by the router in the VPC.

During experiments, a NAT instance is created on the public subnet during experiments and OAISIM instances are created on the private subnets. Packets sent from OAISIM instances to the Internet are routed as follows.

(1) Packets sent from OAISIM instances are relayed to the NAT instance by the router.

(2) The NAT instance applies IP masquerade to packets sent from private subnets, then send them to the router.

(3) The router forwards packets sent from the NAT instance to the Internet gateway.

We used `t2.micro` instance for the NAT instance and `m5.large` instance for OAISIM instances. The specifications of those instances are described in Table 2. Moreover, we measured Round Trip Time (RTT) from each private subnet to our LAN by `ping` command. Table 3 shows the average RTT of 100 ICMP packets sent from instances created on each private subnet to our globally accessible gateway. From these results, we can estimate that the one-way delay between OAISIM instances and our EPC network is roughly 5.95 [ms], because the propagation delay among the globally accessible gateway and EPC network is sufficiently small.

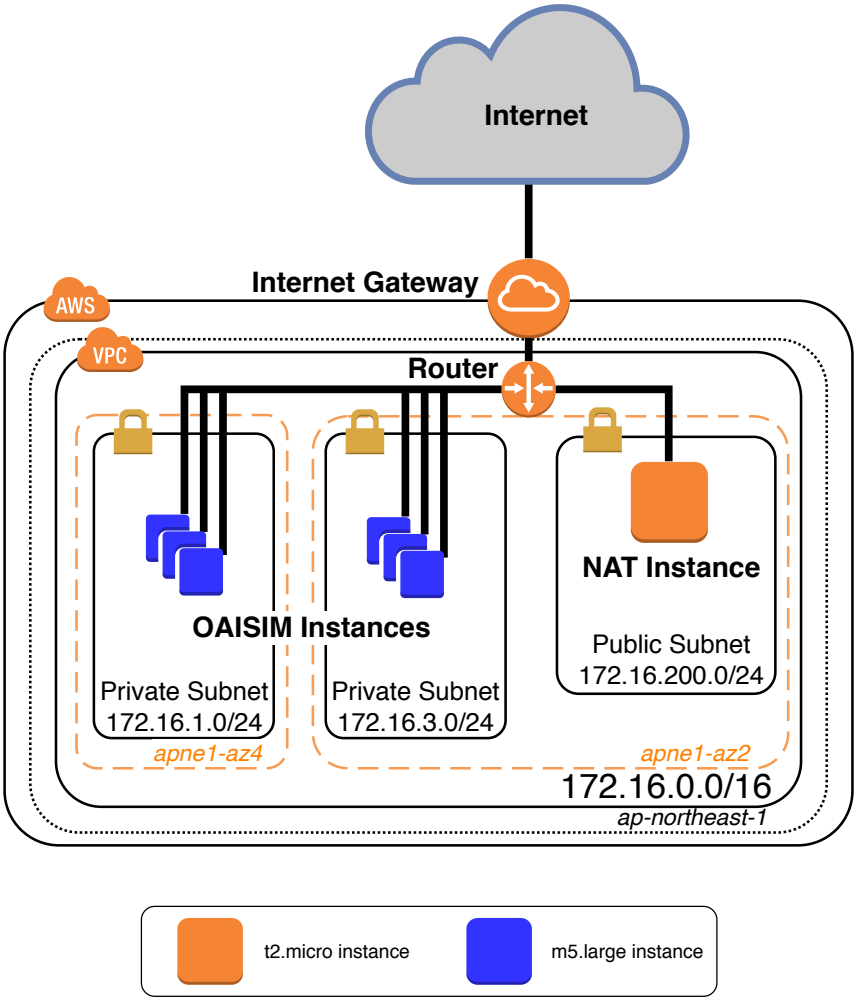


Figure 4: Configuration of OAISIM network

4.3 Measurement Method

The assessment of experimental results was conducted based on packet capture data obtained by `tcpdump`. The packet capture was operated on the following two network interfaces, as depicted in Figure 3:

- **eth2 on the MME:** Monitors S1AP signaling packets passing through S1-C and S1-U interfaces.
- **eth0 on the Packet Capture Host:** Monitors Diameter signaling packets passing through an S6-a interface and GPRS Tunneling Protocol version 2 (GTPv2) signaling packets passing through an S11 interface.

The packet capture is activated just before an experiment begins and terminated just after the experiment finishes. Consequently, all signaling packets passing through S1-C, S1-U, S6-a, and S11 interfaces during the experiment are recorded in the packet capture data. Since each signaling packet contains identifiers of UEs, we can evaluate a detailed bearer establishment procedure of each UE.

To prevent time differences between the two capture points, the MME and the Packet Capture Host synchronize their clocks by Network Time Protocol (NTP). The MME runs as an NTP server, and an NTP client on the Packet Capture Host refers to the NTP server on the MME. NTP clients on the HSS and the SP-GW also refer to the NTP server on the MME to synchronize clocks of other nodes. NTP clients on OAISIM instances refer to Amazon Time Sync Service, which is accessible from AWS EC2 instances.

4.4 Experiment Procedure

In our experiments, multiple UEs began the attach procedure over a short time duration, to assess the impact of massive accesses from UEs on the performance of the EPC and the bearer establishment procedure. For that purpose, the following steps were utilized for the experiment:

(1) Activate a NAT instance on the public subnet. Then, configure the router to forward packets from private subnets to the NAT instance.

(2) Activate required number of OAISIM instances on the private subnets. Each private subnet contains half of required number OAISIM instances.

(3) Notify the OAISIM instances of the time which is 120 [sec] after as synchronization time point. Additionally, a value of T_{expect} [sec] is given to each instance.

(4) Send a signal to each OAISIM instance to execute an activation command of the eNodeB and UE.

(5) The eNodeB that is operated on each instance adjusts the timing when sending Initial UE Msg. by the following procedure.

   (a) t_{adjust} is set to a random value between (0, T_{expect}) [sec].

   (b) The message transmission time point is calculated as the time point which is t_{adjust} [sec] after the notified synchronization time point.

   (c) Sent Initial UE Msg. to the MME at the message transmission time point.

The above steps made it possible to send the concentrated attach request messages (Initial UE Msg. in Figure 2) from the UEs in the OAISIM instances to the MME. In addition, the concentration level of attach request messages could be configured by T_{expect}.

4.5 Evaluation Metrics

The bearer establishment time for a certain UE is defined as the time difference between the time points when Initial UE Msg. that include an identifier of the UE arrives at the MME and the time point when Modify Bearer res. that include an identifier of the same UE arrive at the MME. The bearer establishment time includes the message propagation delays among the OAISIM instance and EPC nodes and the processing delay of messages at the MME, the HSS, the SP-GW, and the eNodeB and UE in the OAISIM instance. In addition, delays in processing each message (represented by orange square boxes in Figure 2) were evaluated. In detail, for each message processed, the processing delay was defined as a time difference between when the corresponding signaling message arrived at the node and when the corresponding message left the node after processing. Moreover, memory usage on the virtual machine for the MME is also evaluated.

4.6 Synchronization Accuracy

We conducted two types of experiments. First, we evaluated the impact of the number of UEs connected simultaneously on the mobile network. In this experiment, we utilized 1, 2, 4, 8, 16, 32, 64 and 128 OAISIM instances and T_{expect} was set to 0 [sec]. Next, we evaluated how the concentration level affects the EPC performance. In this experiment, we utilized 128 OAISIM instances and T_{expect} was set to 0.1, 0.2, 0.4, 0.8, 1.6, 3.2, 6.4 [sec]. We carried out ten times of experiment for each number of instances and the values of T_{expect}. Table 4 shows the relationship between the number of OAISIM instances and the values of T_{expect} and actual synchronization accuracy observed from experiments. Here, the actual synchronization accuracy is defined as the time difference between the time point when the first Initial UE Msg. arrives at the MME and the time point when the last Initial UE Msg. arrives at the MME. We present the minimum and maximum values of the actual synchronization accuracy among ten experiments. From these results, we obtained the actual synchronization accuracy close to T_{expect} regardless of the number of OAISIM instances and setting of T_{expect}. Note that when we set the number of instances to 1, it is obvious that the actual synchronization accuracy becomes 0 because only one Initial UE Msg. is sent through during experiments. When we set the number of instances to 128 and T_{expect} to 6.4 [sec], we obtained an abnormal maximum value of the actual synchronization accuracy. This is because a transmission of Initial UE Msg. of one UE was extremely delayed in one experiment. We can obtain 6.4×10^3 [sec] as the maximum value of the actual synchronization accuracy when we eliminate the abnormal result. Those results mean that the number of OAISIM instances and settings of T_{expect} accurately had an effect in controlling the concentration level of attach requests from UEs.

Table 2: Specifications of Instances

Instance type	vCPU	Memory [GiB]
`t2.micro`	1	1
`m5.large`	2	8

Table 3: Results of `ping` from AWS network to our laboratory LAN

CIDR of Subnet	Availability zone	Avg. RTT [ms]
172.16.1.0/24	`apne1-az4`	11.3
172.16.1.0/24	`apne1-az2`	12.5

Table 4: Actual synchronization accuracy

Number of Instances	T_{expect} [s]	Actual Synchronization Accuracy [ms]
1	0	$0 - 0$
2		$3.9 \times 10^{-2} - 6.2 \times 10^{0}$
4		$1.4 \times 10^{-1} - 1.1 \times 10^{1}$
8		$3.5 \times 10^{0} - 1.0 \times 10^{1}$
16		$6.2 \times 10^{0} - 9.8 \times 10^{0}$
32		$7.7 \times 10^{0} - 1.1 \times 10^{1}$
64		$6.8 \times 10^{0} - 1.4 \times 10^{1}$
128		$8.9 \times 10^{0} - 1.5 \times 10^{1}$
128	0.1	$9.2 \times 10^{1} - 1.1 \times 10^{2}$
	0.2	$1.9 \times 10^{2} - 2.0 \times 10^{2}$
	0.4	$3.8 \times 10^{2} - 4.0 \times 10^{2}$
	0.8	$7.7 \times 10^{2} - 8.0 \times 10^{2}$
	1.6	$1.6 \times 10^{3} - 1.6 \times 10^{3}$
	3.2	$3.1 \times 10^{3} - 3.2 \times 10^{3}$
	6.4	$6.2 \times 10^{3} - 1.2 \times 10^{4}$

5 Evaluation Results

5.1 Bearer Establishment Time

Figure 5 shows the relationship between the number of OAISIM instances in logarithmic scale and the average bearer establishment time. Error bars laid on y axis explain the minimum and maximum bearer establishment time of ten experiments, where we set $T_{expect} = 0$ [sec] for all OAISIM instances. From these results, we can observe that the bearer establishment time slightly increased when the number of OAISIM instances increased from 1 to 64. However, when we set the number of OAISIM instances to 128, the bearer establishment time sharply increased; it increased by around 450% compared with the case when the number of OAISIM instances was 1. Figure 6 depicts the average message processing time at each EPC node as a function of the number of OAISIM instances. This graph presents the breakdown of bearer establishment time except for the message processing time at eNodeBs and UEs and one way delay between OAISIM network and the globally accessible gateway. Since eNodeBs and UEs were operated on OAISIM simulator, we omitted the processing time of eNodeBs and UEs in the latter evaluations as well. As shown in Figure 6, it is obvious that the increase in the bearer establishment time was mainly caused by the increase in the processing time at the MME. Consequently, we can conclude that the increase in the number of UEs significantly affected on the performance of the MME.

Figure 7 shows the relationship between T_{expect} given to OAISIM instances represented by logarithmic scale and the average bearer establishment time, where 128 OAISIM instances are activated. Error bars laid on y axis explain the minimum and maximum bearer establishment time of each ten experiments. Note that the minimum bearer establishment time in case of $T_{expect} = 0.2$ [sec] is abnormally smaller than other results. This is because one of ten experiments with $T_{expect} = 0.2$ [sec] was conducted by only 99 OAISIM instances. We can obtain 2.8 [sec] as the minimum bearer establishment time when we eliminate the abnormal result. As shown in this figure, larger T_{expect} reduced bearer establishment time even though the number of OAISIM instances is unchanged. When we set T_{expect} to 6.4 [sec], the bearer establishment time was reduced by about 79 % compared with the case of $T_{expect} = 0$ [sec]. Figure 8 represents the relationship between given T_{expect} and the message processing time on EPC nodes. In the graph we plot the average additional delay, caused by setting T_{expect}, on the top of each result. In this context, average additional delay represents the delay before each UE starts its data transmission

regardless of the actual load on the MME. It is calculated as a half of T_{expect}, because each eNodeB executed on OAISIM instances select t_{adjust} between (0, T_{expect}) randomly. As shown in Figure 8, the message processing time at the MME significantly mitigated by increasing T_{expect}. However, large T_{expect} causes substantial additional delay to each UE. We will discuss on this issue in detail in Section 5.4.

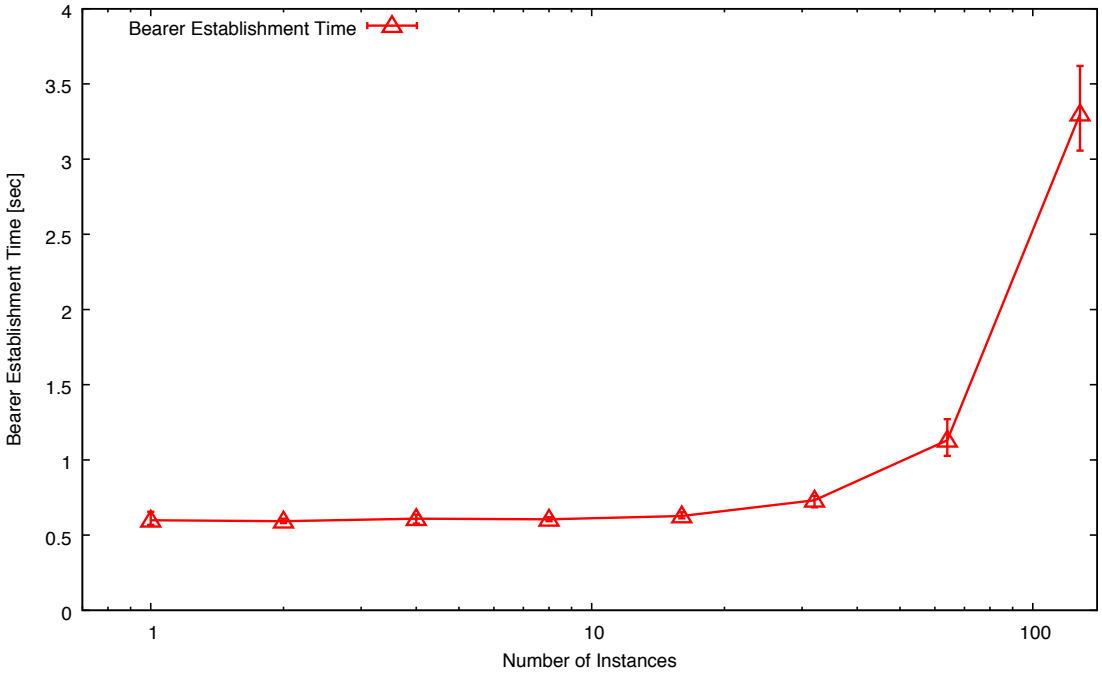


Figure 5: Relationship between the number of OAISIM instances and bearer establishment time

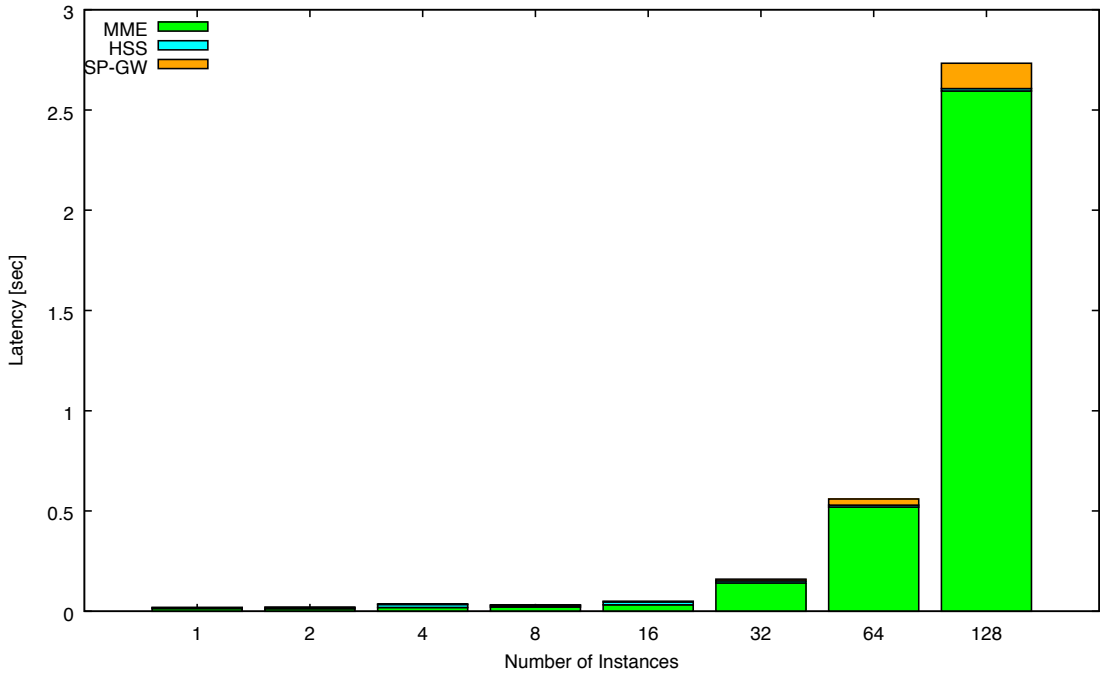


Figure 6: Relationship between the number of OAISIM instances and signaling processing time on each EPC node

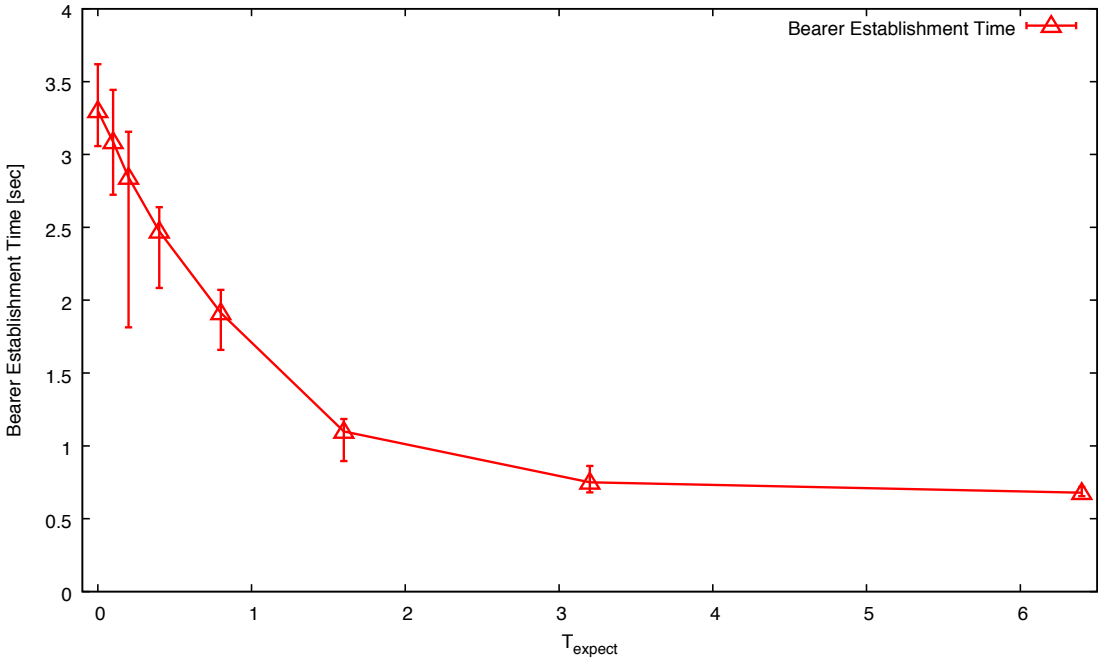


Figure 7: Relationship between T_{expect} and bearer establishment time

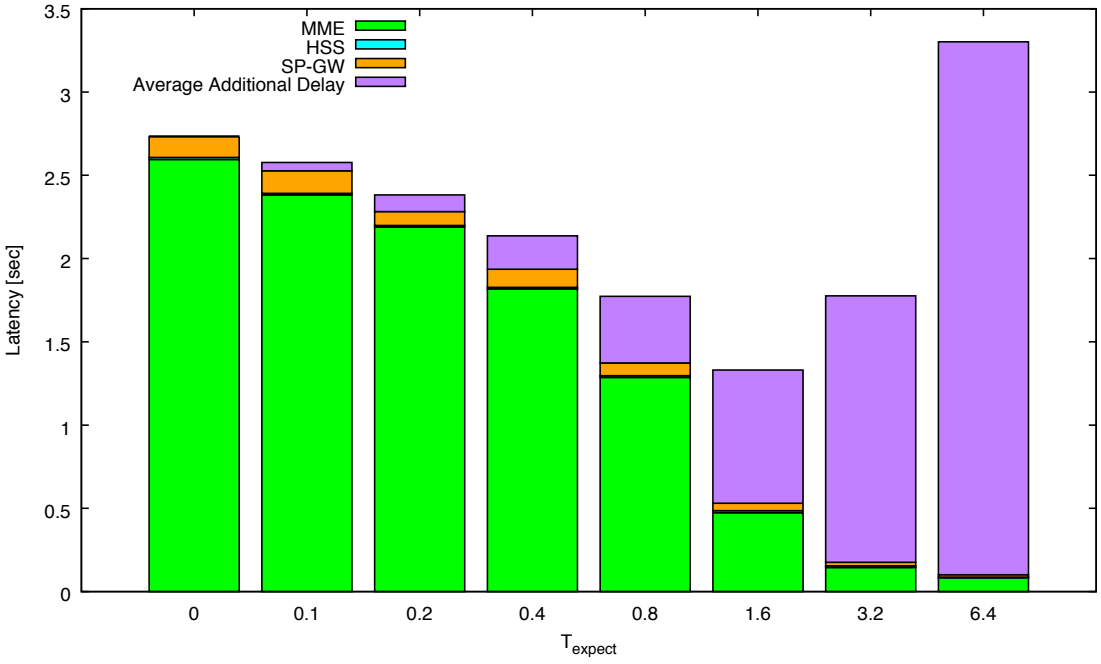


Figure 8: Relationship between T_{expect} and signaling processing time on each EPC node

5.2 Queue Length at the MME

Figure 9 presents the temporal changes in the queue length at the MME, calculated as follows:

- When a signaling packet arrives at the MME at a certain time point, the queue length at that time point is incremented by one.
- When a signaling packet is sent from the MME at a certain time point, the queue length at that time point is decremented by one.

Note that the figures plot the results of all ten experiments.

As shown in the figures, the configuration of T_{expect} apparently affected the queue length at the MME. The queue length kept large during experiments when T_{expect} was set to 0, 0.1, 0.2, 0.4, 0.8 [sec]. This indicates that the packet processing speed at the MME is roughly the same as the packet arrival speed at the MME. Since the concentration level of attach request decreased when T_{expect} became large, the queue length became less obviously.

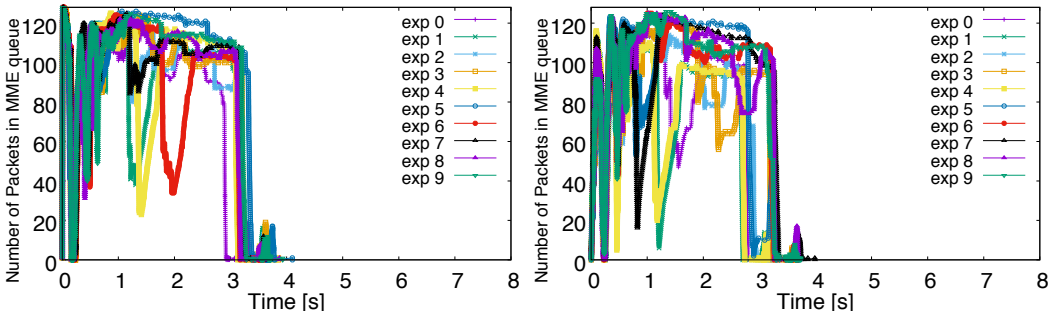
5.3 Memory Usage on the MME

We also evaluated the relationship between the number of OAISIM instances and the memory usage on the virtual machine for the MME. The memory usage is obtained by `vmstat` command, which shows the temporal change of the memory usage at one second intervals. Define M_{free}, M_{buffer}, M_{cache} as the value of `free`, `buffer`, `cache` obtained from output of `vmstat`, respectively. At this time, we can calculate the available amount of memory M_{avail} as follows [24]:

$$M_{avail} = M_{free} - M_{buffer} + M_{cache} \tag{1}$$

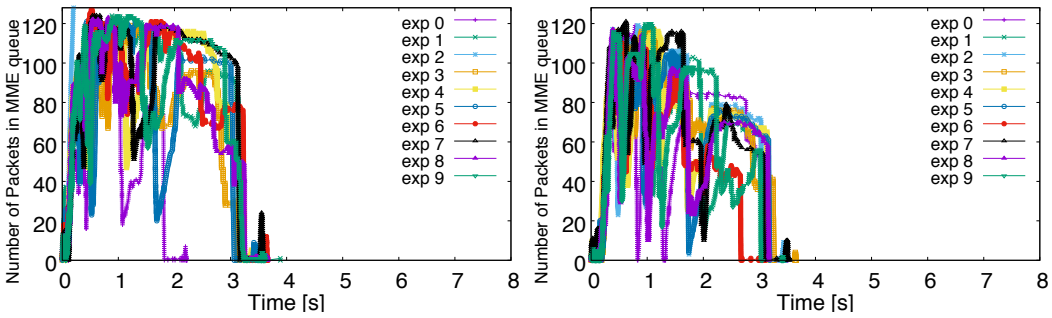
The average M_{avail} before UEs send connection requests is calculated as the average M_{avail} from 40 seconds before the synchronization time to that from 20 seconds before the one. Similarly, the average M_{avail} after processing connection requests from UEs is calculated as the average M_{avail} from 20 seconds after the synchronization time to that from 40 seconds after the one. Therefore, the consumed amount of memory is calculated as the difference between the value of M_{avail} before and after processing UEs' connection requests.

Figure 10 shows the memory usage for accommodating UEs on the MME. Note that this graph shows the relationship between memory usage and the number of connected UEs, which is defined



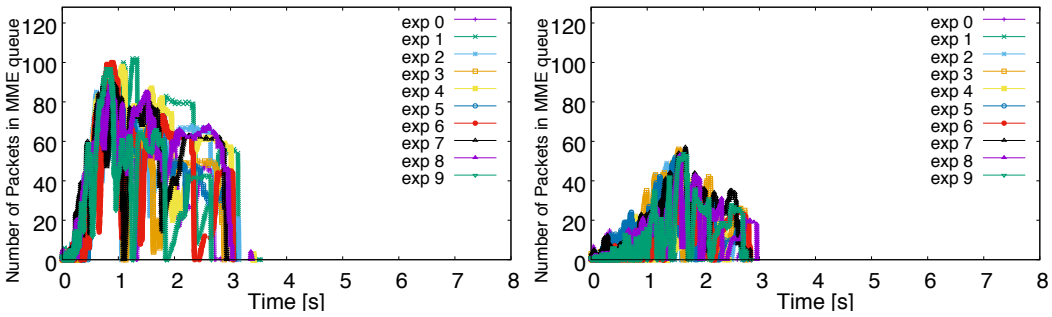
(a) $T_{expect} = 0$ [sec]

(b) $T_{expect} = 0.1$ [sec]



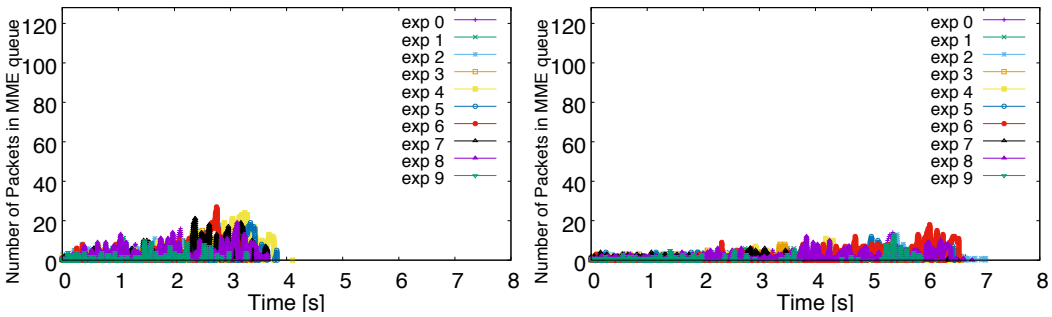
(c) $T_{expect} = 0.2$ [sec]

(d) $T_{expect} = 0.4$ [sec]



(e) $T_{expect} = 0.8$ [sec]

(f) $T_{expect} = 1.6$ [sec]



(g) $T_{expect} = 3.2$ [sec]

(h) $T_{expect} = 6.4$ [sec]

Figure 9: Queue length at the MME

as the number of Modify Bearer Response message arrived at the MME. The figure shows that the memory usage on the MME linearly correlated with the number of connected UEs. Since the virtual machine of the MME has 1 GB memory, only approximately 10 % memory was used even when the MME processed attach requests from 128 UEs. Note that we have tried to accommodate 256 and more UEs by activating more OAISIM instances, then the application of the MME has crashed when some UEs sent attach requests.

5.4 Discussion

As shown in Figure 6 and Figure 8, the main factor of the inflation of the bearer establishment time was the increased signaling message processing time at the MME. Moreover, Figure 9 illustrates how the growth of the queue length at the MME strongly affected the processing time on the MME. Since this is the result with up to 128 UEs, the increased processing delay on the MME became a more critical issue when accommodating a larger number of UEs. When a particularly large number of M2M/IoT terminals simultaneously connect to the mobile network and start data transmission, the data transmission is delayed due to the concentration of attach requests, which increased the processing delay at the MME. Since some M2M/IoT terminals only transmit a small amount of data, the inflation of bearer establishment time becomes a substantial overhead on their communication.

The simplest way to deal with the above problems is the enhancement of computing resources for an MME. However, since most M2M/IoT terminals have an extremely low Average Revenue Per Unit (ARPU) compared to traditional terminals (for example, 2.20 USD per month [25]), it is difficult to recover the cost of reinforcing computing resources. Additionally, some M2M/IoT terminals communicate periodically; that is, not all of them always utilize the network. For this reason, the enhanced resource is temporally wasted. Consequently, static enhancement of computing resources is not always a desirable method for network operators in terms of OPEX and CAPEX. Therefore, methods such as server virtualization, optimal and adaptive resource allocation, and C/U plane separation with SDN technologies are required when accommodating M2M/IoT terminals.

One possible way to deal with the above problems is temporarily distributing attach requests from UEs intentionally. As shown in the results of our experiments in Figure 8, decreasing the concentration level of attach requests from UEs by using larger value of T_{expect} significantly re-

duces the processing time on the MME. Therefore, increasing the value of T_{expect} can decrease the load on the MME when accommodating massive M2M/IoT terminals to mobile core networks. However, it is obvious that introducing T_{expect} brings additional delay in starting data transmission regardless of the actual load on the MME, as we described in Section 5.1. Therefore, we should discuss whether or not the decrease of the bearer establishment time by introducing T_{expect} can compensate the additional delay of the data transmission. From Figure 8, we observe that the processing time on the MME decreased by roughly 2.5 [sec] when T_{expect} increased from 0 [sec] to 6.4 [sec]. However, $T_{expect} = 6.4$ [sec] causes 3.2 [sec] additional delay in average, thus the total latency of message processing became larger than the one of $T_{expect} = 0$ [sec]. In this case, introducing $T_{expect} = 6.4$ [sec] has a bad effect on the data transmission performance of UEs. On the other hand, when we set T_{expect} to 1.6 [sec], the processing time on the MME is decreased by about 2.0 [sec]. The total amount of latency is reduced by 1.2 [sec] even when we considered 0.8 [sec] of average additional delay. In this case, introducing $T_{expect} = 1.6$ [sec] becomes more reasonable. We expect that the suitable value of T_{expect} changes according to various factors such as the number of UEs to be attached and the amount of computing resources for EPC nodes.

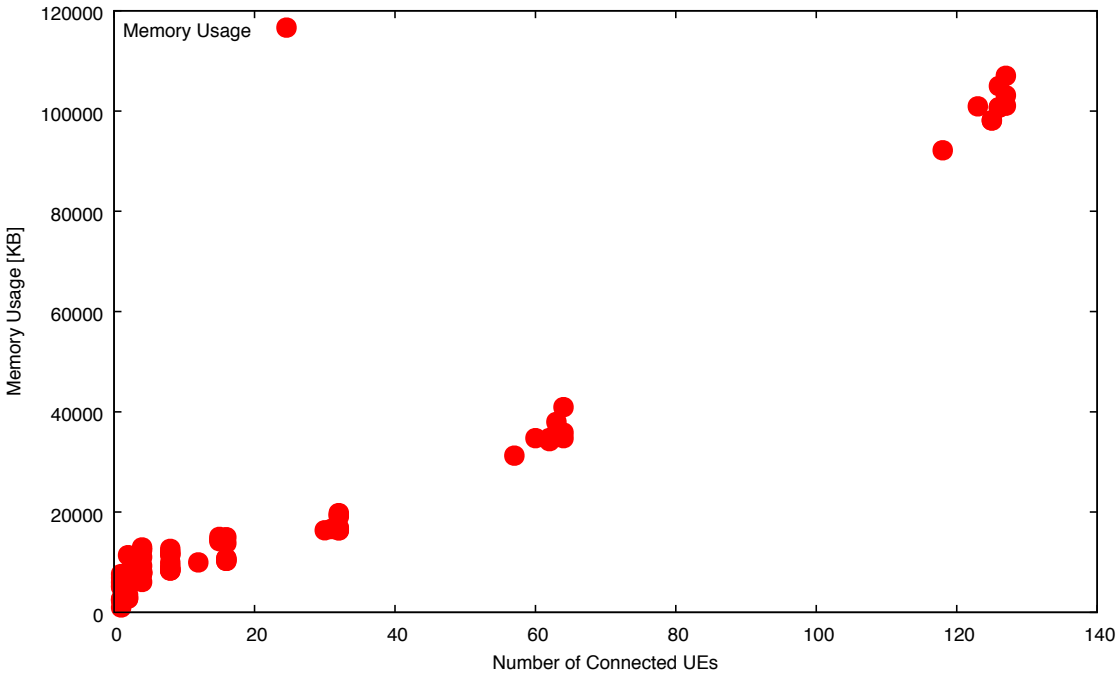


Figure 10: Relationship between the number of connected UEs and memory usage

6 Conclusion

In this thesis, we presented the experimental evaluation results of the performance of a mobile core network to assess the impact of massive accesses from M2M/IoT terminals. We established a simple experiment model based on open-source implementation of mobile core networks and emulated user terminals. Then, we evaluated the processing delay of the attach procedure at each mobile core node. We revealed that the simultaneous attach requests from 128 UEs increased the bearer establishment time on the MME by up to about 450%. We also found the relationship between the inflation of the processing time on the MME and queue length at the MME. Furthermore, we clarified that the optimal setting of T_{expect} can improve the bearer establishment time including additional delay. For employing network slicing, which is considered in future 5G networks, the experimental results in this thesis can be applied to resource provisioning of EPC nodes according to UEs' communication characteristics in each slice.

In future work, we plan to investigate the effect of server resources (such as CPU speed and memory size) on the bearer establishment time to estimate the amount of resources required to accommodate massive M2M/IoT terminals. Furthermore, it is important to assess the effects of applying server virtualization and C/U plane separation with SDN to mobile core networks.

Acknowledgment

I'd like to show my great appreciation to Professor Morito Matsuoka. His advice and insights supported my research activity in various situations. Also, I am grateful to Professor Masayuki Murata. He gave me insightful and precise comments and advice for many times. And I'd like to express my sincere thanks to Associate Professor Go Hasegawa. Without his guidance, persistent help and warm encouragement, my research would not be possible. Finally, I'd like to offer my special thanks to students of Matsuoka Laboratory for their support of my laboratory life.

References

[1] D. Astely, E. Dahlman, A. Furuskär, Y. Jading, M. Lindström, and S. Parkvall, "LTE: The Evolution of Mobile Broadband," *IEEE Communications Magazine*, vol. 47, no. 4, pp. 44–51, Apr. 2009.

[2] P. Marsch, I. D. Silva, O. Bulakci, M. Tesanovic, S. E. E. Ayoubi, T. Rosowski, A. Kaloxylos, and M. Boldi, "5G Radio Access Network Architecture: Design Guidelines and Key Considerations," *IEEE Communications Magazine*, vol. 54, no. 11, pp. 24–32, Nov. 2016.

[3] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low Power Wide Area Networks: An Overview," *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 855–873, Secondquarter 2017.

[4] LoRa Alliance, "LoRaWAN - What is it?" *A technical overview of LoRa and LoRaWAN*, 2015.

[5] "Sigfox - The Global Communications Service Provider for the Internet of Things (IoT)," available at https://www.sigfox.com.

[6] "Wi-Fi HaLow," available at https://www.wi-fi.org/discover-wi-fi/wi-fi-halow.

[7] R. S. Sinha, Y. Wei, and S. H. Hwang, "A Survey on LPWA Technology: LoRa and NB-IoT," *ICT Express*, vol. 3, no. 1, pp. 14 – 21, Mar. 2017.

[8] *Cellular System Support for Ultra-low Complexity and Low Throughput Internet of Things (CIoT)*, Third Generation Partnership Project, Nov. 2015, V13.1.0.

[9] *Study on provision of low-cost Machine-Type Communications (MTC) User Equipments (UEs) based on LTE*, Third Generation Partnership Project, Jun. 2013, V12.0.0.

[10] A. Tawbeh, H. Safa, and A. R. Dhaini, "A Hybrid SDN/NFV Architecture for Future LTE Networks," in *Proceedings of 2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[11] A. Basta, W. Kellerer, M. Hoffmann, H. J. Morper, and K. Hoffmann, "Applying NFV and SDN to LTE Mobile Core Gateways; The Functions Placement Problem," in *Proceedings of*

the 4th Workshop on All Things Cellular: Operations, Applications, & Challenges. ACM New York, NY, USA, Aug. 2014, pp. 33–38.

[12] Z. A. Qazi, V. Sekar, and S. R. Das, "A Framework to Quantify the Benefits of Network Functions Virtualization in Cellular Networks," *CoRR*, vol. abs/1406.5634, Jul. 2014.

[13] F. Z. Yousaf, J. Lessmann, P. Loureiro, and S. Schmid, "SoftEPC — Dynamic Instantiation of Mobile Core Network Entities for Efficient Resource Utilization," in *Proceedings of 2013 IEEE International Conference on Communications (ICC)*, Jun. 2013, pp. 3602–3606.

[14] G. Hasegawa and M. Murata, "Joint Bearer Aggregation and Control-Data Plane Separation in LTE EPC for Increasing M2M Communication Capacity," in *Proceedings of 2015 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2015, pp. 1–6.

[15] S. Abe, G. Hasegawa, and M. Murata, "Effects of C/U Plane Separation and Bearer Aggregation in Mobile Core Network," *IEEE Transactions on Network and Service Management*, vol. 15, no. 2, pp. 611–624, Jun. 2018.

[16] "OpenAirInterface," available at http://www.openairinterface.org.

[17] A. Jain, Sadagopan N S, S. K. Lohani, M. Vutukuru, "A Comparison of SDN and NFV for Re-designing the LTE Packet Core," in *Proceedings of 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Nov. 2016, pp. 74–80.

[18] J. Prados-Garzon, P. Ameigeiras, J. J. Ramos-Munoz, P. Andres-Maldonado, and J. M. Lopez-Soler, "Analytical Modeling for Virtualized Network Functions," in *Proceedings of 2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2017, pp. 979–985.

[19] L. J. Chaves, I. C. Garcia, and E. R. M. Madeira, "An Adaptive Mechanism for LTE P-GW Virtualization Using SDN and NFV," in *Proceedings of 2017 13th International Conference on Network and Service Management (CNSM)*, Nov. 2017, pp. 1–9.

[20] H. Chang, B. Qiu, C. Chiu, J. Chen, F. J. Lin, D. de la Bastida, and B. P. Lin, "Performance Evaluation of Open5GCore over KVM and Docker by Using Open5GMTC," in *Proceedings*

of NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium, Apr. 2018, pp. 1–6.

[21] "Docker - Build, Ship, and Run Any App, Anywhere," available at https://www.docker.com/.

[22] C. A. García-Pérez and P. Merino, "Experimental Evaluation of Fog Computing Techniques to Reduce Latency in LTE Networks," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 4, p. e3201, Jun. 2018.

[23] "Amazon Web Services (AWS) - Cloud Computing Services," available at https://aws.amazon.com/.

[24] *Manual page vmstat(8)*, Sep. 2011.

[25] S. Kechiche, "Cellular M2M Forecasts and Assumptions: 2010-2020," GSMA Intelligence, Tech. Rep., 2014.