A Local Mobility Management Entity (MME) for Stationary Devices in virtual Evolved Packet Core (vEPC)

Bo-Jun Qiu, Jia-Ru Li, Yu-Sen Hsueh, You-Min Lin, Jyh-Cheng Chen, Ping-Fan Ho
Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan

Abstract—**In the future, there will be many devices in 5G networks, and these devices will cause a great number of control messages. The core network should handle these control messages locally to maintain the quality of other services. Furthermore, the local MME should turn off the location update flows in the initial attach procedure to reduce the number of control messages. We implement an MME, which can turn off the location update flows, and then compare the performance.**

I. INTRODUCTION

In the future 5G networks, many devices will access the networks to get communications with the other machines. Internet of Things (IoT) will also provide many innovative services. The characteristics of IoT are different from human beings. The traffic in control messages is expected to be much more than that in data plane. The large amount of control messages sent from machines may congest the networks, thus deteriorate the quality of service of other applications.

To reduce the large amount of control messages sent from IoT devices, one of the potential solutions is to use the local Mobility Management Entity (MME) to handle the control messages locally instead of sending them to the remote and central MME. Therefore, the quality of other applications may not be affected. Moreover, for some stationary devices, the MME can turn off the *location update* in the *initial attach* procedure because these devices will not move. Thus, it can further reduce the number of control messages.

To localize MME and turn off location update flows, we need to virtualize and manage the functions of MME, and then decide whether to perform the location update flows or not. In this paper, we implement a MME which can turn off the location update flows in the initial attach procedure. We call it as S-MME. The implementation is done in a virtual Evolved Packet Core (vEPC) called *Reconfigurable Core (RECO)* [1], [2]. We also compare the performance between the original MME and S-MME.

II. RECO

A. RECO architecture

Fig. 1 shows the architecture of Radio Access Network (RAN) and RECO, including MME, Serving/Packet Data Network Gateway (S/P-GW), and Home Subscribe Server (HSS). It essentially is based on the traditional 4G core network. In RECO, currently, S/P-GW and HSS are the same as those in openair-cn [3] because we focus on MME which belongs to the control plane and is more complex and important than other entities.

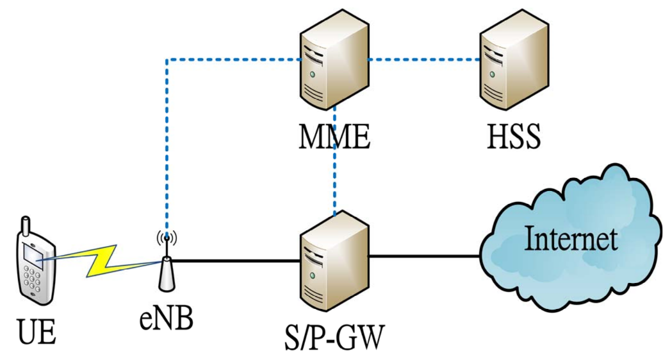

Fig. 1. Architecture of RAN and RECO. The User Equipment (UE) and Evolved Node B (eNB) belong to RAN, and the MME, S/P-GW, HSS belong to EPC.

B. RECO MME

We have made the MME in RECO reconfigurable. Fig. 2 shows the architecture of RECO MME, including an identifier and a dynamic linking framework. The identifier will identify the type of the traffic, and produce a corresponding configuration file. Please note that currently the identifier is just a pseudo one. We implemented a check-box to produce the configuration file. The dynamic linking framework then will link some customized libraries according to the configuration file, and compose a customized MME with some common libraries. The MME is customized according to the features of the type of traffic. We then can create a network slice with the other entities in the EPC to serve this type of traffic with better performances.

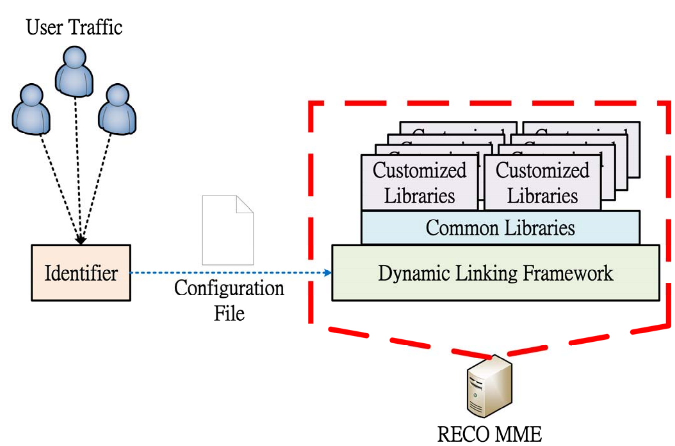

Fig. 2. Architecture of RECO MME.

C. Stationary MME (S-MME)

S-MME is a special MME which removes the location update flows from the initial attach procedure for stationary devices. We assume the location information of the stationary devices is already existed in the HSS when the user registers. We then implement the corresponding libraries, and use the dynamic linking framework to compose an S-MME in RECO.

III. EVALUATIONS

A. Initial attach time

We test the original MME and S-MME in the following two scenarios. The first one is to run MME/S-MME and HSS on the same hardware device. The other one is to run MME/S-MME and HSS on different hardware devices. We set these two scenarios because the flow we remove in S-MME is location update, which is the call flow between MME and HSS. Letting MME/S-MME and HSS run on the same or different hardware devices may affect the time to finish the location update, which then will further affect the time of a UE's initial attach.

In Fig. 3, it is obviously that the initial attach time of S-MME is shorter than that of original MME no matter in which scenario. This is because the S-MME does not have to finish the location update flows between MME and HSS. The initial attach time in the different hardware devices is longer than that in the same hardware device for MME. This is because it needs longer time to send the packets between two different hardware devices.

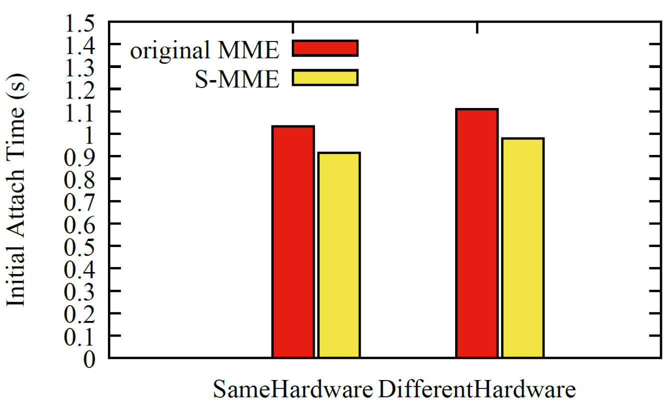

Fig. 3. Initial attach time of the original MME and S-MME under two scenarios. The 'SameHardware' means MME/S-MME and HSS run on the same hardware device. In the 'DifferentHardware' case, MME/S-MME and HHS run on the different hardware device.

In Table I, we can see that S-MME needs less than one second in average to finish the initial attach procedure. The original MME, however, needs longer than one second. Moreover, S-MME reduces 10% of initial attach time than the original MME.

B. Other performance metrics

In this section, we compare S-MME with the original MME in terms of throughput, latency, CPU utilization, and memory usage.

TABLE I
INITIAL ATTACH TIME OF ORIGINAL MME AND S-MME

Scenario	original MME	S-MME	Unit	Improvement
Same hardware	1.033	0.914	s	11.48 %
Different hardware	1.109	0.979	s	11.72 %

s = second, improvement = (MME - S-MME)/ MME.

In Table II, we can see that the performance of the original MME and S-MME are almost the same for each metric. This is because the removed location update flows are in the control plane and the flows are called when MME is running the initial attach procedure. Therefore, it will not affect the data plane or other performance metrics after the MME has finished the initial attach procedure.

TABLE II
PERFORMANCE OF ORIGINAL MME AND S-MME

	original MME	S-MME	Unit
Throughput	41.96	42.17	Mbps
Latency	66	65.3	ms
CPU	0.286	0.307	%
Memory	49.4	49.4	MB

Mbps = megabit per second, ms = millisecond, % = percentage, MB = megabyte

IV. CONCLUSIONS

In future 5G networks, there will be many devices connected to the EPC. They will cause a great number of control messages that affect the quality of human services. One of the good solutions is to handle these control messages with local MME. It can turn off the location update flows in the initial attach procedure to decrease the number of control messages. With the experiments in this paper, we show the performance can be improved by turning off the location update flows without causing other overheads.

V. ACKNOWLEDGMENTS

We would like to thank Chia-Wei Chang, Chin-Yu Lin, and Yi-Hua Wu for their contributions to this paper. This work was supported in part by the Ministry of Science and Technology of Taiwan under grant numbers MOST 106-2221-E-009-046-MY3, 106-2221-E-009-047-MY3, 106-2218-E-009-016.

VI. REFERENCES

[1] Chia-Han Wu, Wei-Jen Chen, and Jyh-Cheng Chen, "RECO: A Reconfigurable Core Network for Future 5G Communication Systems" MobiCom, Snowbird, Utah, USA, pp. 594-596, Oct. 2017.

[2] http://reconet.org.

[3] https://gitlab.eurecom.fr/oai/openair-cn.