東海大學資訊工程學系研究所

碩士論文

指導教授:呂芳懌

在 SGW 和 eNB 下運行邊緣計算之效能評估--以防火牆為例

Performance evaluation of edge computing under the SGW and eNB -- Taking firewall as an example

研究生:林勝政

中華民國 108 年 7 月 14 日

東海大學碩士學位論文考試審定書

東海大學資訊工程學系 研究所

研究生 林 勝 政 所提之論文

在SGW和eNB下運行邊緣計算之效能評估--以防火牆為例

經本委員會審查，符合碩士學位論文標準。

學位考試委員會
召 集 人 ＿＿＿＿＿＿＿＿ 簽章

委 員 ＿＿＿＿＿＿＿＿

＿＿＿＿＿＿＿＿

＿＿＿＿＿＿＿＿

指 導 教 授 ＿＿＿＿＿＿＿＿ 簽章

摘要

在 the fifth generation(5G)中，因需大量導入 IOT sensor，使得資料量呈現爆炸性的成長。在如此高流量的環境下卻要有低延遲的條件，這會讓無線傳輸的安全性的實現，變得極為困難。在導入 Software Define Network(SDN)與 Mobile Edge Computing(MEC)後，無線網路安全的問題有了新的解答。但是，在 5G 中因使用通訊範圍較小的 small cell，與 4G 相比，換手的次數較為頻繁，使得網路服務或功能的遷移，變成一個極需解決的問題。在本文中，我們提出了一個在 EPC 或 eNB 下的防火牆建立流程，並結合 EPS-AKA 程序，達到減少封包傳遞的數量。我們也探討 EPC 中的實體故障，以 load balance 的方法，平均分配故障實體的工作量。另外，針對頻繁換手問題，我們考慮了 Intra-MME, Inter-MME, Inter-EPC，三個環境下的換手流程，並結合 UE handover 的程序，精簡封包數量，降低換手的時間。最後，我們也有提到在非信任環境的換手，並提出一個可行的方法。我們也期望未來能夠此流程能夠擴大成其他服務也能使用，例如，即時翻譯。

中文關鍵詞: Fifth generation (5G)，軟體定義網路，行動邊緣計算，防火牆，容錯，換手，功能遷移，

Abstract

In the near future, a huge amount of network will flow through the fifth generation (5G) network since a tremendous number of IOT devices/sensors will soon connect to their application platforms via 5G. In such a heavy-traffic environment, low-latency requirement will seriously impact wireless transmission security. Also, 5G adopts Software Defined Network (SDN) and Mobile Edge Computing (MEC) which conduct short transmission delays and user-defined security may be a solution. Also, in 5G due to employing small cells of small communication ranges, compared with those adopted by 4G, the number of handover will be relatively frequent, that the migration of network services or functions will be another problem yet to be solved. In this paper, we propose a firewall establishment process which installs firewalls in an EPC or eNB. We also implement a fault tolerant mechanism to detect the hardware failures in EPC and then distribute the workload of the failed network entity to other entities of the same functions following the principles of load balance. To solve the problem of frequent handover, we design a handover procedure for each of the three environments, including Intra-MME, Inter-MME and Inter-EPC, which are tightly integrated with UE handover procedure, aiming to reduce the number of transmitted messages and the time consumed by handover. Finally, we also expect that this firewall migration process can be applied to other services, e.g., the migration of instant translation function, in the near future.

Keywords: Fifth generation (5G), Software Defined Network (SDN), Mobile Edge computing(MEC), firewall, fault tolerance, handover, function migration

Content

Content of Figure

Content of Table

Chapter 1 Introduction

With the fast development of wireless networks, people recently have extremely relied on handheld devices to process and handle their everyday-life activities. The situation has significantly increased in recent years [1][2]. Also, in the future, IoT traffic which carries sensed data may seriously congest network. To mitigate this problem, researchers are trying to accelerate the development of 5G network (or simply 5G). 5G devices will be soon available in the middle 2019. However, when people enjoy convenient and colorful lives through Internet and 5G systems, data security of handheld devices will be one of the key issues for secure communication [3-5].

In the past two decades, malicious programs or viruses are often spread with data to intrude network systems. Hackers have used these methods to act illegally [6], e.g., eavesdropping network packets, hacking the Internet, implanting virus, setting backdoors, etc. Although many solutions have been proposed, the security of wireless transmission when using mobile phones with limited resources still has attracted researchers' attention. Currently, the standards of 5G networks are almost completed. How to use 5G features including SDN (Software Defined Network), NFV (Network Function Virtualization), Network Slicing and MEC (Mobile Edge Computing) to enforce mobile-phone security is also one of the important research trends at current stage.

The design of SDN [7-9] decouples control plane and data plane of a communication system, allowing administrators to re-plan functions of a network with programs. It is a new method to control network traffic and provides a well-defined platform for the development of network services and applications. The data centers of Facebook and Google utilize Openflow protocol [10] as the protocol to control software

defined switches, i.e., OpenvSwitches (called OvS or User Plane Function (UPF) in 5G). Many telecom companies in the world have organized the Open Network Foundation (ONF) [11] to accelerate the development of SDN. To improve the flexibility and efficiency for entire 5G networks, OpenvSwitch will replace SGW and PGW [12][13] in 5GC (5G Core) to deliver network packets, and the operations of Openflow switch are managed by SDN controllers. MEC [14][15] as one of the prospective mechanisms in 5G supports cloud/fog computing with the help of edge computers geographically close to users in the network. This can significantly decrease the chance that users request network services directly from core network repeatedly, consequently lowering the probability of network congestion and significantly reducing their events' response time.

Today, firewalls are provided by using dedicated machines which are often deployed in the demilitary zone of a system. It is hard for network administrators to set up individual packet filter policies for a specific user. Also, firewall systems are usually installed at a fixed location. Users with mobile phones may move anywhere and anytime. How to provide these users with movable firewalls is an engineering challenge.

Therefore, in this paper, we proposed a firewall architecture for mobile users based on MEC. The firewall functions are implemented in edge computers. The purpose is to reduce the distance of packet delivery and network latency, hence lowering the probability of packets loss and eavesdropping. All edge computers are managed by an EPC or eNB. Together with SDN controller, they control packet delivery and provide firewall service. If firewalls are installed in EPC, a packet *p* that needs to be detected and filtered by firewall will be sent to SGW (outbound) or PGW (inbound) which will follow their settings to transmit *p* to the corresponding edge computers. Voice and RTP packets are sent to their destination via PGW (or SGW) directly without delivering them to firewalls because they usually do not intrude a system, e.g., issuing DOS/DDOS

attacks which are detected by IDS/IPS (Intrusion Detection System / Intrusion protection System). If firewalls are installed in eNBs, when the packet *p* that needs to be detected arrives at eNB, eNB will forward *p* to a packet management mechanism to perform firewall services. Those without the need of packet filtering will be forwarded to UE (inbound) or SGW (outbound) without sending them to this management mechanism. The management mechanism is also developed in in this study.

When UE hands over, its serving firewall has to be migrated NMAG. Our method integrates the proposed firewall infrastructure with existing handover process to reduce the amount of signaling packets. Further, when a network entity E fails, the services that E offers will be assigned to E's standby network entities, based on these standby entities' workloads, aiming to achieve load balance among them. Other reasons are shortening the interruption of network services provided to UE and preventing these entities from overloading.

The contributions of this study are as follows:

1) We have designed a firewall establishment procedure based on SGW or eNB to protect UE.

2) When an edge computer, e.g. E1, fails, the firewall services provided by E1 will migrate to other edge computers immediately to continue UE's firewall services. If SGW fails, UE's packet *p* would transfer to other SGW to keep *p* to be normally delivered.

3) Because of small cell in 5G [16], the number of base stations increases given a fixed area. The amount of handovers will be higher compared to that of a 4G environment. We propose the method of firewall migration which is individually integrated with Intra-MME, Inter-MME and Inter-EPC handover so that firewall services provided to UE will not be interrupted too long during and after handover.

The rest of this paper is organized as follows. Chapter 2 describes the related studies and background of this paper. Chapter 3 introduces the architecture of the proposed firewall architecture, Chapter 4 describes the processes of firewall migration in the events of network entity failure and UE handover. Experimental results are presented and discussed in chapter 5. Chapter 6 concludes this paper and addresses our future studies.

Chapter 2. Background and Related work

2.1 EPS-AKA

EPS-AKA (EPS Authentication and key agreement) is a security protocol developed for mutual authentication between users and 4G networks. As shown in Figure 1, when UE is switched on, it sends an Attach Request (including IMSI, UE Security Capacity, KSI, ...) -- step1, to MME via a eNB. After receiving it, MME follows the contents of this message to prepare an Authentication Data Request (including IMSI, SN ID, Network Type, ...) -- step2, and sends the message to HSS -- step3. HSS then looks for the SN ID and IMSI in its own database. If the SN ID or IMSI is invalid, the authentication is then terminated. Otherwise, according to the IMSI, some other parameters and the UE's corresponding key K will be retrieved for HSS to generate *n* authentication vectors $AV_i, 1 \leq i \leq n$. HSS further encapsulates the *n* AVs into the Authentication Data Response -- step4, and sends it to MME -- step5, MME selects one from the *n* AVs, keeps XRES and K_{ASME} in the AV, and prepares a User Anthentication Request (including RAND, $AUTN_{HSS}$, KSI,…) -- step6. After that, MME sends this message to UE -- step7. UE generates $AUTN_{UE}$, RES and K_{ASME} according to RAND, SN ID, SQN and LTE K, and verifies whether $AUTN_{HSS}$ is equal to $AUTN_{UE}$ or not -- step8. If they are equal, the authentication on UE side completes, and an User Anthentication Response (including RES) is sent to MME -- step9. On receiving the response, MME compares RES with XRES -- step10. If they are equal, the authentication on MME side finishes. MME will notify the underlying eNB to start serving this UE.

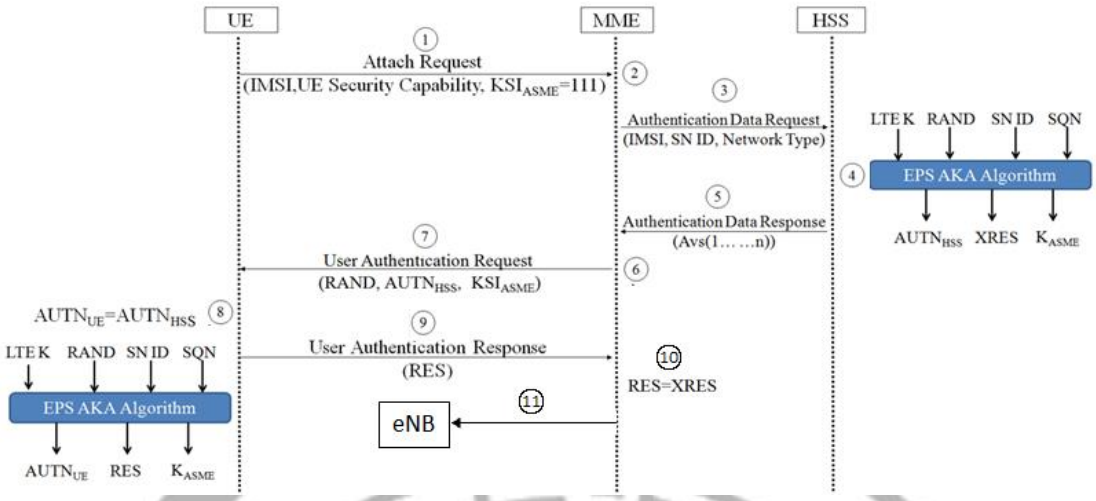

Figure 1. EPS-AKA authentication process.

2.2 Related Studies

Literature indicates that mobile security is one of the important issues in wireless networks. Arins [17] allowed users to install matching rules as user-defined security mechanism into edge computers of telecom companies. Openflow was employed to protect users from network attacks. However, the study only supported three APIs for users to develop their high-level services. Actually, more APIs are required in the near future. Also, it would be better for authors to investigate the situation in which UE hands over and network components fail. Zope *et al.* [18] utilized a SDN controller, Openflow protocol and load-balance policies to construct a virtual network. The controller empoyed was floodlight [19][20]. Authors also introduced how to help researchers to develop and test their applications, and how to gradually develop a friendly environment needed by the underlying network. However, this paper did not address UE handover and authentication. Gray *et al.* [21] set up a virtual firewall in a cluster environment, in which the Commodity off the Shelf (COTS) hardware server was connected to the element manager (EM) which was responsible for monitoring network statuses and managing system operations. However, the firewall was built in a

cluster environment. All its computers need to have the ability in dealing with network traffic.

Fichera *et al.* [22] presented a 5G scenarios which integrates SDN-based edge networks, clouds and the Internet of Things to improve the reliability and robustness of 5G environments by adding an SDN orchestrator to this scenario. The data transmission paths could adaptably go through different network domains to perform the SDN orchestration so that switches along the paths could dynamically adjust the transmission connection to the next switch, e.g., *S*, based on the load of *S*, attempting to avoid network congestion and ensure reliable network services.

Zhang *et al.*[23] placed MEC servers to the location close to eNBs to compress or calculate nearby popular video programs or multimedia contents so that the programs and contents could be fast stored in the cache of the eNB, smart vehicles or handheld devices to reduce the frequency that users extract data from the corresponding core network, aiming to achieve a better transmission efficiency. However, the experiments were conducted on a network simulator, rather than on a real system. The drawbacks are that traffic flow and the number of people are dynamically changed, meaning that the results presented were a little far from the real situation.

Bellavista *et al*.[24] predicted user mobility patterns in a hostile environment and proactively migrated virtual functions to the next MEC node in advance, thus significantly reducing the time of service disruption. Authors also investigated the migration of virtual fuctions in a reactive mode. The simulated environment utilized OpenCV and LibSNM[25] on the Elijah platform. Guoa *et al.* [26] employed the Path-set Database Generation, Flow-table Management and Routing Decision functions in a STAR architecture to help flow table for properly keeping current flow entries. The purpose is to avoid those no longer being used entries from occupying flow-table so as to reduce the chance of flow-table overflow, consequently realizing a high-quality

network transmission environment with low load, low delay and high transmission efficiency. Authors also compared the performance of three architectures including LRU+OSPF, AC+OSPF and STAR.

Liu *et al.* [27] optimized the joint placement of satellite gateways and SDN controllers in a 5G satellite network with ground gateways (like SGW and PGW) to reduce the propagation delay between ground and the satellite to a limited value and obtain maximum transmission reliability between them. Authors used the Simulated Annealing Algorithm (SAA) to deploy the satellite gateways and claimed that the optimal enumeration algorithm (OEA) could achieve an optimal solution. In fact, this system utilizes the simulated annealing and clustering hybrid algorithm (SACA) to complete the joint placement of the gateway, and authors claimed that its controller can approximate to its optimal reliability.

Chapter 3. System architecture

This chapter describes the architectures of our firewall systems when they are installed in SGW or eNB, presents how these systems work when one of their network entities fails and depicts the corresponding procedures when UE hands over.

3.1 SGW firewall

We first introduce the firewall established under SGW.

3.1.1 The data structure of SGW firewall

As shown in Figure 2, the architecture of SGW firewall system consists of E-UTRAN, EPC and SGW-firewall. The former two belong to a 4G system. In this architecture, we add the third which comprises a Manager and edge computers. Edge computers provide firewall services to UEs. All UE packets needed to be filtered by firewalls must be sent to these edge computers for security checking. Those passing the checking will be forwarded to their destinations.

The Manager attached to SGW firewall maintain three tables, including the EC table (Table 1), EC-status table (Table 2) and the UE authentication table (Table 3). Table 1 has four fields, i.e., UE-IP, SGW-IP, EC-IP and Firewall-URL, respectively, keeping IP of UE, IP of SGW, IP of edge computer serving this UE and the firewall-URL of this UE.

Table 2 includes three fields as EC-IP, Number of UE and EC-status, which record IP of an edge computer, the number of UEs currently served by this edge computer and the operaton statuses of the edge computer, repectively. When EC-status=1, it indicates

that the edge computer operates normally. EC-Status=0 represents that currently it fails.

Table 3 contains three fields, i.e., the IMSI, Step and Auth-status, which keep IMSI of UE, current authentication step of the UE and the status of the authentication, respectively. The second field, i.e., step, will be decribed later. When Auth-status=1, indicating that the EPS-AKA operates normally, whereas Auth-status=0 shows that currently authentication has been interrupted owning to the failure of network / EPC component.

Edge computers are managed by Manager. An edge computer creates a Firewall_address table (Table 4) which keeping firewall imformation has two fields, UE-IP and Memory address, recording IP of the UE and address of this UE's firewall in memory, respectively. After the firewall is installed, the edge computer enters the starting address of memory of the firewall into the Firewall_address table for later service.

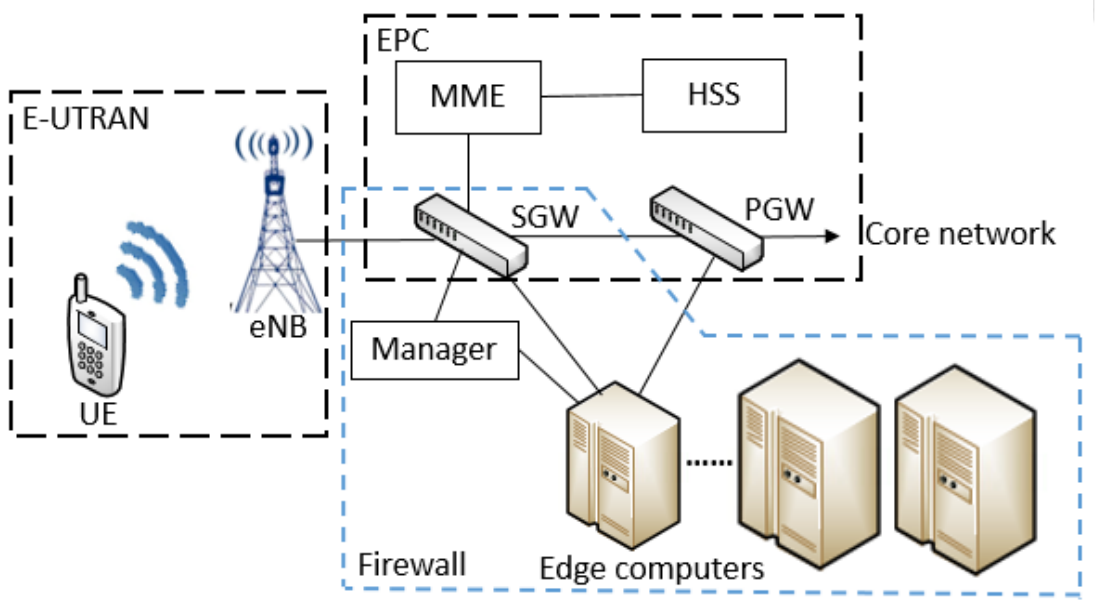

Figure 2. The architecture of a SGW firewall.

Table 1. EC table of manager.

UE-IP	SGW-IP	EC-IP	Firewall-URL

Table 2. EC-status table of manager/Keeper.

EC-IP	Number of UE	EC-status

Table 3. UE authentication table of manager.

IMSI	Step	Auth-status

Table 4. Firewall_address table.

UE-IP	Memory address

3.1.2 The features of SDN controller, Manager and edge computer

The main function of the SDN controller is allocating UEs to the SGWs according to the load of the SGWs. The functions of the Manager are as follows. (1) Selecting an edge computer to service UE based on the edge computer's load; (2) Recording the UE's firewall information in Table 1; (3) Checking to see whether the edge computer operates normally by employing a polling approach. The manager sends a message to each edge computer at a regular time interval and waits for their replies. If it has not received reply from an edge computer before timer times out for three times, it considers that the edge computer fails and EC-status field of Table 2 will be set to 0. (4) Helping MME to record and resume authentication step before and after some network components fail.

In the EPS-AKA procedure, all the Authentication messages passing through MME as shown in Figure 1 will be sent to the manager which continuously records the process step and authentication status in Table 3. When receiving a packet *p*, the edge computer accesses *p*'s IP address (probably source IP or destination IP), looks for the corresponding firewall in memory (Table 4), and filters the *p* with the UE's firewall

policies. If *p* passes the checking, it will be sent to the eNB (inbound) or PGW (outbound). Otherwise, *p* is dropped.

3.1.3 SGW Firewall Establishment Procedure

Figure 3 shows the sequence chart of a SGW-firewall established on an edge computer. The steps are integrated with the EPS-AKA authentication process. After receiving **Attach request** from UE, eNB forwards this message to MME. MME sends an **Authentication command** to the manager (including IMSI, etc. – step 2 of Figure 3). Manager created a tuple in the UE authentication table (Table 3) for the UE according to UE's IMSI. When receiving an EPS-AKA authentication message from eNB, MME informs Manager of this event. Manager then records the EPS-AKA step, in Step field of Table 3. Second, MME sends an **Authentication data request** (including IMSI, SN ID, network type) to HSS (step 3 of Figure 3, also step 3 of EPS-AKA).

HSS then generates AVs (step 4 of EPS-AKA) and looks for the firewall URL in its URL database based on the UE's IMSI carried in this message. If the user had applied for firewall services, HSS will encapsulate the URL in the **Authentication data response** (including AVs, firewall URL, ... – step 4 of Figure 3 and step 5 of EPS-AKA). In the EPS-AKA, the UE authenticates the MME based on the information of its own and that carried in the **User authentication request** (step 7 of EPS-AKA and step5 of Figure 3) and MME authenticates UE with the RES contained in the **User authentication response** sent by UE to MME (step 9 of EPS-AKA and step 6 of Figure 3). The MME transmits a **Firewall-service request** (including UE's IP, firewall-URL) to the SDN controller (step 7 of Figure 3). The controller selects a SGW with the lowest load to serve the UE and sends an **UE-service request** (including the UE's IP, firewall-

URL)) to the SGW (step 8 of Figure 3). The SGW added a tuple for the UE in its flow table, the schema of which is shown in Table 5, recording IP of the UE in the Match Fields (the first field), filling a null in the Instructions field, and then sending an **Edge computer request** (including UE's IP, SGW's IP, firewall- URL, ..., -- step 9 of Figure 3) to the Manager.

Manager then looks up Table 2 (EC-status table), selects an edge computer with minimum load, i.e., the edge computer with the least number in the field of Number of UE, and then fills UE's, SGW's and edge computer's IPs and firewall URL into the EC table (Table 1). The Number of UE field of this edge computer in the EC-status table (Table 2) is increased by one owing to serving the UE. The Manager replies the SGW with an **Edge computer response** (including UE's IP and edge computer's IP -- step 10 of Figure 3), telling SGW the IP of the edge computer. The SGW fills this IP into the Instructions field of the tuple prepared for this UE in the flow table.

After that, all the packets flowing to or from the UE are forwarded to the edge computer. Next, Manager delivers a **Firewall-active command** (including UE's IP and firewall URL, ...) to edge computer (step 11 of Figure 3). On receiving this message, edge computer downloads the firewall settings, records the memory address of the firewall in the Memory address field of Table 4. and starts to execute the firewall for filtering this UE's packets. Those packets passing the filtering will be sent to UE (inbound) or PGW (outbound).

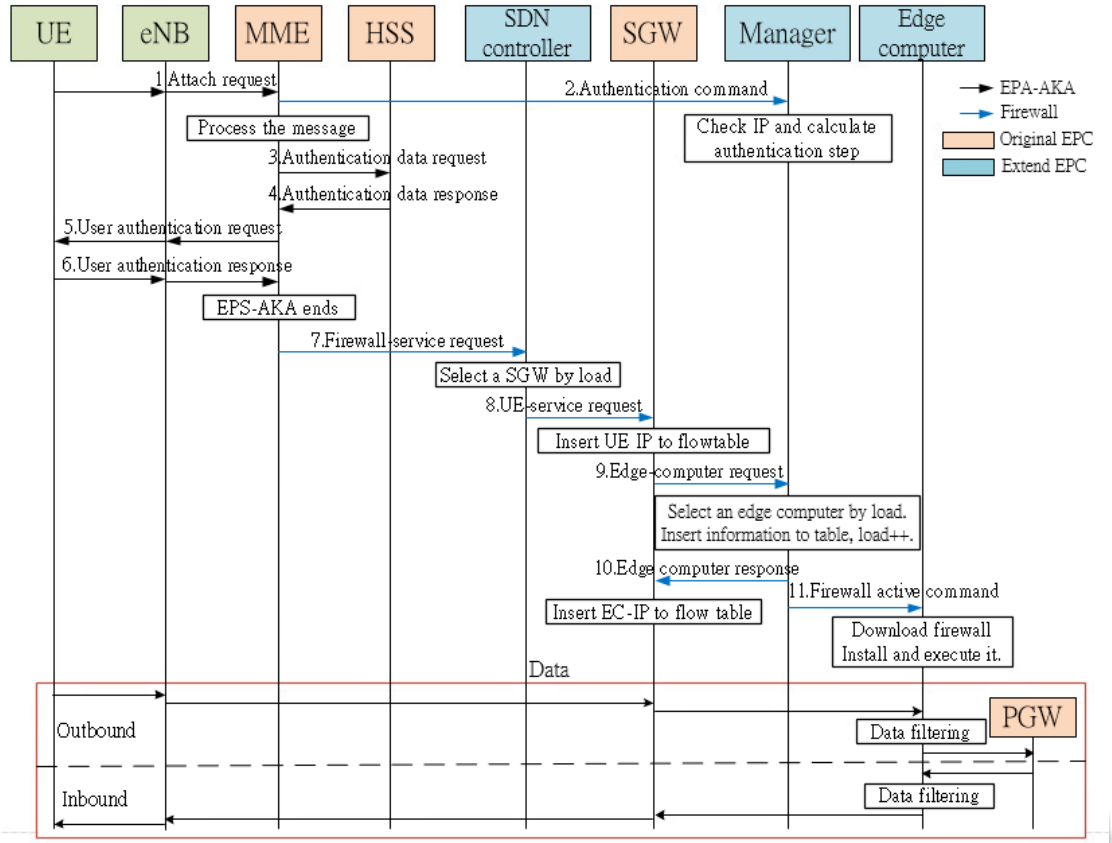

Figure 3. Sequence chart of SGW firewall establishment procedure.

Table 5. Main fields of a flow entry in SGW's flow table.

Match Fields	Priority	Counters	Instructions	Timeouts	Cookie

3.2 eNB firewall

Next, we will introduce the firewall established under on eNB.

3.2.1 The features of eNB firewall

The eNB firewall architecture as shown in Figure 4 consists of eNB, Keeper and edge computer, in which Keeper plays the role similar to that of Manager mentioned above. The difference is that Keeper cannot help MME to record the EPS-AKA steps because eNB is not responsible for UE authentication. In other words, this architecture

does not provide the function of authentication restore when some network entities fail. So Table 3 is ignored in this architecture. But like that in a SGW firewall system, packets are forwarded to edge computers through Keeper. Edge computers are attached to the eNB, rather than SGW, and managed by Keeper. Two tables are used by the Keeper, i.e., EC-status table for keeper which reuses the schema of Table 2, and the EC table for Keeper (Table 6), the schema of which is similar to that of Table 1. But there is no SGW-IP since no SGW is utilized. An edge computer also uses Table 4 to assist finding the firewall location in memory. Each eNB has one Keeper and several edge computers.

Table 6. EC table of Keeper.

UE-IP	EC-IP	Firewall-URL

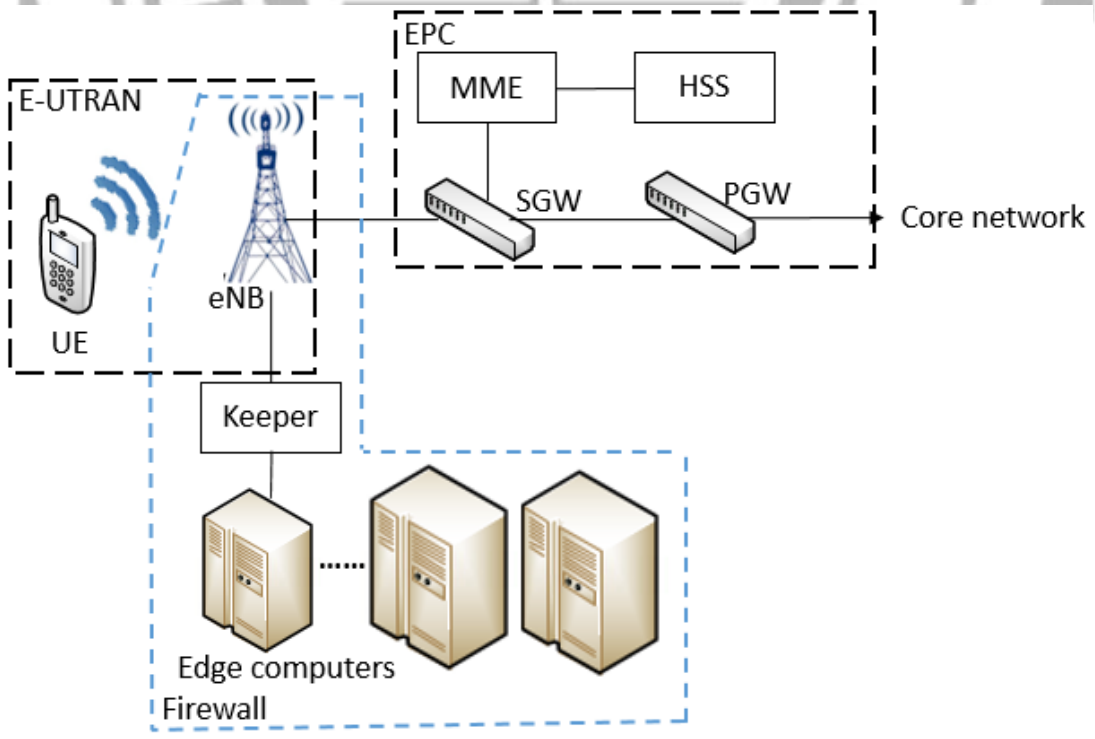

Figure 4. The architecture of an eNB firewall.

3.2.2 eNB Firewall Establishment Procedure

Figure 5 shows the establishment procedure of an eNB firewall. On receiving an **Attach request** (step 1) sent by UE to MME through eNB, MME sends an **Authentication data request** (step 2 of Figure 5 and step 3 of EPS-AKA) to HSS. HSS authenticates the UE and retrieves the UE's Firewall URL from its URL database. After receiving an **Authentication data response** (including Firewall-URL -- step 3 of Figure 5. and step5 of EPS-AKA), MME sends an **User authentication request** to the eNB (step 4 of Figure 5 and step 7 of EPS-AKA). The eNB delivers related authentication parameters of EPS-AKA to UE. After receiving an **User authentication response** (step 5 of Figure 5 and step9 of EPS-AKA) and successfully finishing the EPS-AKA authentication (step 10 of EPS-AKA), MME issues a **Firewall-service request** (including UE IP, firewall URL, ... – step 6 of Figure 5) to inform the eNB to start serving the UE.

The eNB transmits an **Edge computer request** (including UE IP, firewall URL, ... -- step 7 of Figure 5) to Keeper. Keeper selects the edge computer with the minimum load according to "Number of UE" field in EC-status table (Table 2, reused), records the UE's IP, edge compute's IP, firewall URL in the "EC table for Keeper" (Table 6), increases the value of "Number of UE" field of this UE's tuple in EC-status table (Table 2), and sends a **Firewall-active command** (step 8 of Figure 5) to the edge computer. The Edge computer retrieves the UE's firewall settings based on the firewall URL, records the memory address of the firewall in Firewall_address table (Table 4) and starts serving the UE with the UE's own firewall.

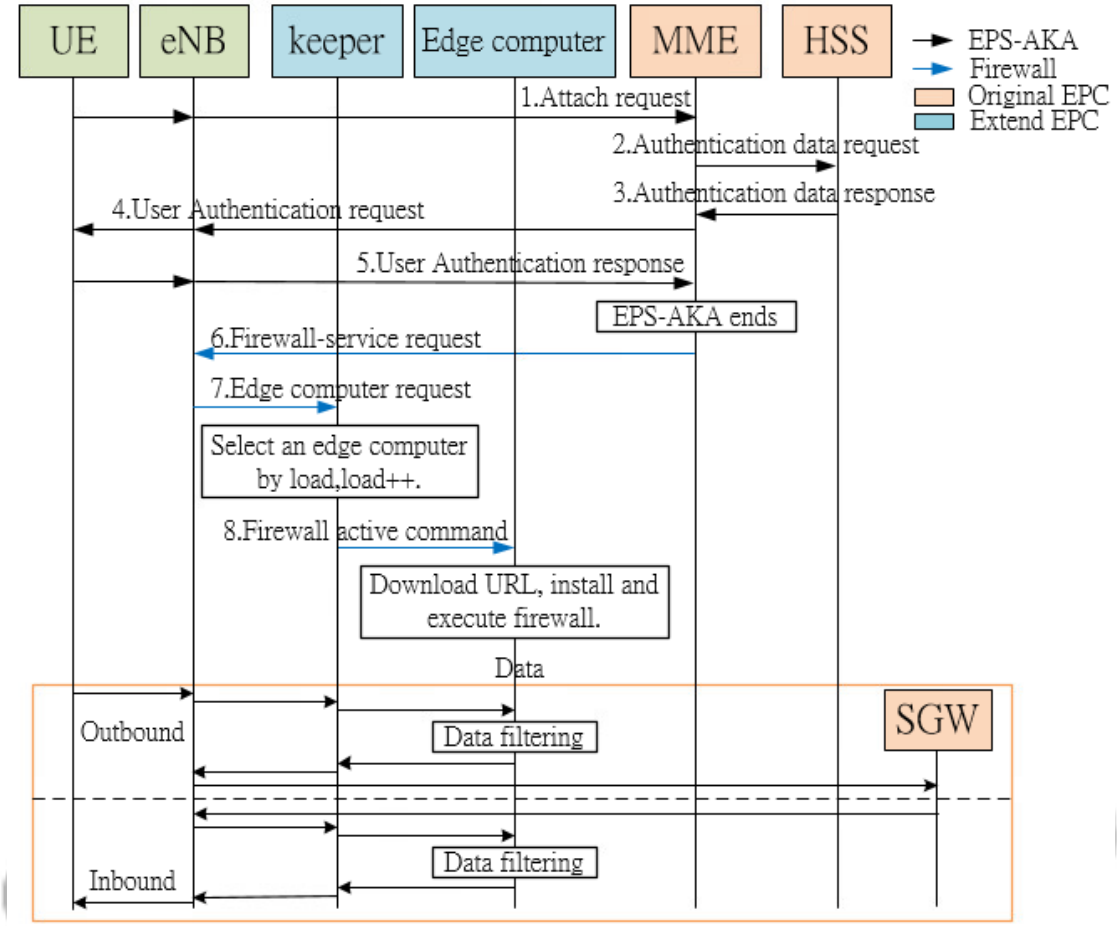

Figure 5. Sequence chart of eNB firewall establishment procedure.

3.3 Network Entity Failure

This section discusses the procedures when SGW and eNB firewalls fail.

3.3.1 SGW Firewall Failure

The SGW firewall failure can be divided into two parts: edge computer failure and SGW failure.

A. Edge computer failure

When an edge computer fails, the takeover procedure is shown in Figure 6.

Basically, the Manager periodically polls heartbeats of all edge computers (step 1) one by one to detect the failure of an edge computer so as to avoid UEs from losing their firewall services. If Manager discovers that an edge computer fails or receives a failure message (step 2) sent by an edge computer before it fails, Manager switches UE's firewall service to other edge computer. It first updates the status field of the edge computer in the EC-status table (Table 2) from 1 to 0 to prevent subsequent UEs from being assigned to this edge computer, selects q alive edge computers, $q \geq 1$, and allocates the firewalls (assuming a total of n firewalls) of the UEs currently served by the failed edge computer to the q edge computers, $q \leq n$, following load balance principles. Assume that K_j UEs' firewalls are assigned to edge computer_j, $K_j \geq 1, 1 \leq j \leq q$.

$$K_j = \left\lceil \frac{\frac{1}{h_j}}{\sum_{i=1}^{q} \frac{1}{h_j}} * n \right\rceil \qquad (1)$$

where h_j represents the number of UEs currently served by edge computer_j.
In theory, if an edge computer serving less UE firewalls will be assigned more UEs' firewalls, and vice versa. For example, as shown in Figure 7, if EC1 fails, the firewalls that serve UE-A, UE-B, UE-C and UE-D need to be migrated to EC2, EC3 and EC4. Because the load of EC4 is the lightest, two firewalls are assigned to it. Each of EC1 and EC2 takes over one firewall.

The Manager sends a **Firewall-active command** (including UE's IP and firewall-URL) -- step 3 to each of the q edge computers, and transmits an **Update message** (including UE's IP and new edge computer's IP) -- step4 to inform SGW that these K_j UEs have been assigned to edge computer_j for all js, $1 \leq j \leq q$. After receiving the command, edge computer_j downloads these K_j UEs' firewall settings, records IP address of these UEs and memory addresses of these firewalls in UE-IP and Memory-

address fields of the Firewall_address table (Table 4), respectively, and then starts executing those firewalls. After that, packets sent to or by UE_i will be forwarded to its new edge computer to perform the firewall filtering. All packets passing the filtering will be delivered to SGW or PGW. The failed edge computer will be removed and the tuples created for this edge computer in Table 2 and 6 will be deleted.

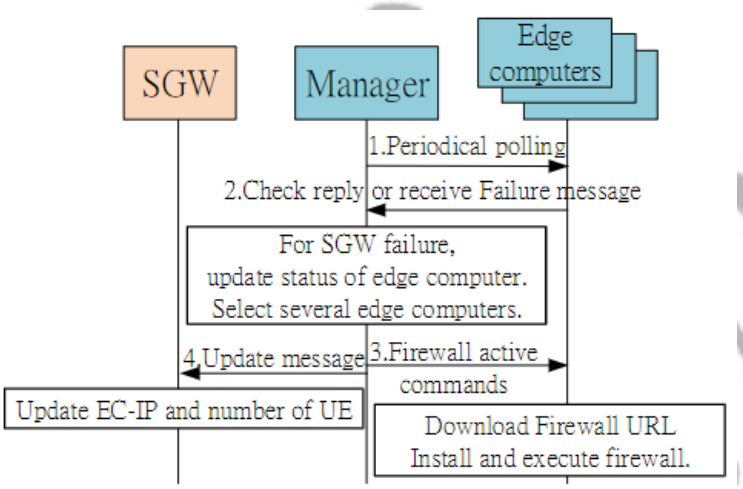

Figure 6. The sequence chart of edge computer failure in SGW firewall.

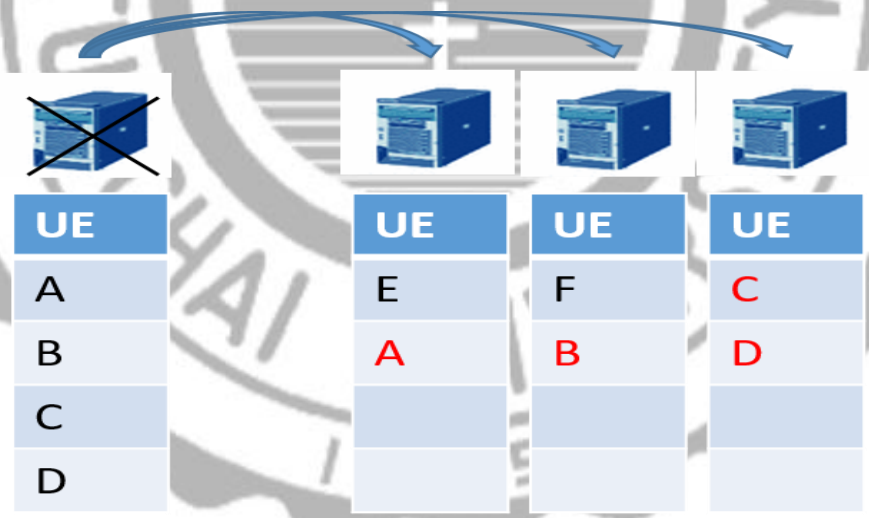

Figure 7. After an edge computer fails, all UEs served by it are allocated to other edge computers according to the loads of these edge computers.

B. SGW failure

The process developed for SGW failure is similar to that of edge computer failure.

As shown in Figure 8, the SDN controller periodically checks status of a SGW -- step 1. If the controller discovers that SGW fails or receives the failure message sent by a SGW before this SGW fails -- step 2, the SGW-failure process will be invoked to prevent some UEs served by this SGW, e.g., a total of n UEs, from losing their SGW services for a long time. In this process, q SGWs are selected, and assigned the n UEs to the q SGWs following Eq.1 for balancing these SGWs' loads. The semantics of this equation is that SGW_j (instead of edge computer) takes over additional K_j UEs and h_j represents the number of UEs currently served by SGW_j. After the assignment, the SDN controller sends a **UE-service request** (step3) to each of the q SGWs, and SGW_j creates a tuple (flow entry) for each of the K_j UEs assigned to it in its flow table (see Table 5). Then SGW_j starts serving these UEs, and sends an **Update message** (step4) to Manager. Manager will update SGW-IP field of the corresponding tuples of these K_j UEs in EC table (Table 1) with IP of SGW_j. These UEs' EC-IPs remain unchanged for all js, $1 \leq j \leq q$.

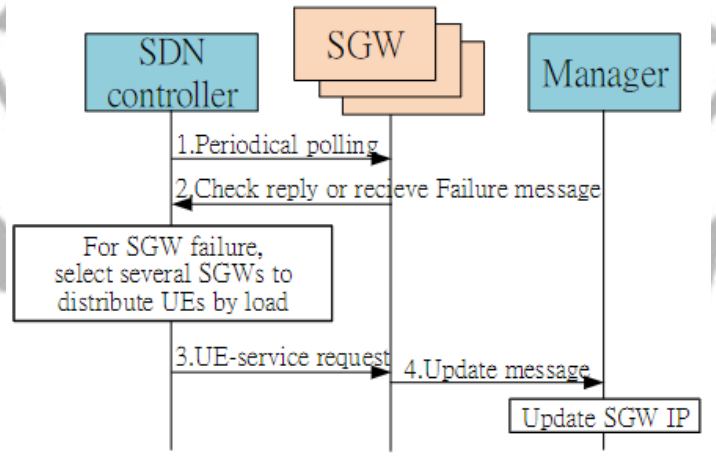

Figure 8. The sequence chart of SGW failure in SGW firewall.

3.3.2 Failure of edge computer in eNB firewall

Figure 9 shows the procedure developed for edge-computer failure when eNB

firewall is employed. The Keeper periodically checks heartbeats of all edge computers by polling -- step 1. If Keeper discovers that an edge computer fails or receives a **failure message** sent by an edge computer before this edge computer fails -- step2, Keeper changes the EC-status field of these UEs currently served by the failed edge computer, e.g., a total of n UEs, in the EC-status table from 1 to 0, and selects q edge computers from its edge-computer pool. Keeper then follows the load balance principles, i.e., Eq.1, to assign firewalls of K_j UEs served by the failed edge computer to edge computer$_j$ for all js, $1 \leq j \leq q$, and sends a **Firewall-active command** (including UE's IP, firewall-URL -- step3) to each of the q edge computers, After receiving the command, edge computer$_j$ downloads the firewalls of these UEs, records the memory addresses of these UEs' firewalls in the field of Memory-address in the Firewall_address table and starts these firewall services. After that, these UEs' packets will be forwarded to their new edge computers for filtering.

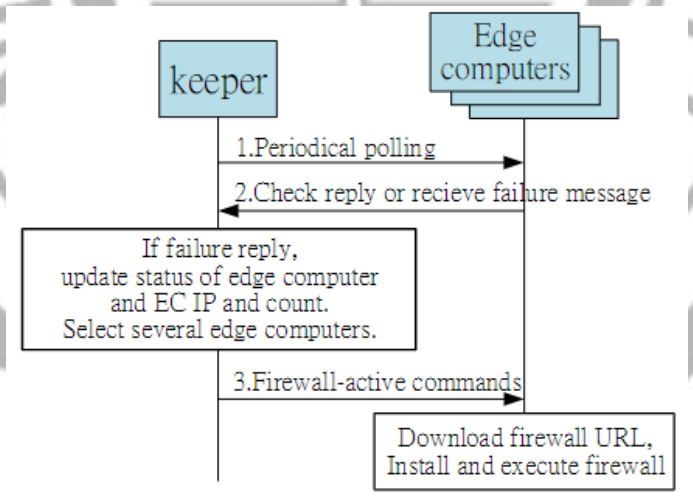

Figure 9. The sequence chart of edge computer failure in an eNB firewall.

Chapter 4 Firewall Migration

This section discusses how to move UE's firewall to the next location when the UE hands over. There are 4 cases, including Intra-MME, Inter-MME, Intra-SGW and Inter-SGW handover. In the Intra-SGW handover, this UE's firewall service is still provided by the original SGW, Manager and edge computer. Therefore, there is no need firewall migration is not needed. The three remaining cases are discussed below.

4.1 Intra-MME handover (Intra eNB X2 handover)

When SGW firewall is employed, Intra-MME handover is in fact Intra-SGW handover since the SGW still serves the UE. That means Intra-MME handover is reduced to Inter-eNB handover. The procedure is shown in Figure 10. When the eNB discovers that the RSRP of an UE is lower than its predefined threshold, the handover process will be triggered. After handover, UE still connects to the same EPC. S-eNB sends a **Handover request** (including C-eNB IP and UE's IP) via the X2 interface established between S-eNB and T-eNB to the T-eNB (step 1) where C-eNB is the eNB currently serving corresponding node (CN). T-eNB delivers a **Load check** message to T-Keeper. T-keeper inquires the loads of all edge computers (step 2), and determines whether or not it can accept the new firewall mission according to the Number of UE field in EC-status table. If none of the edge computer has space to accommodate this new firewall, T-Keeper sends a **Connect-reject** to T-eNB (step 3, but not shown), and T-eNB replies a **handover request NACK** to S-eNB. S-eNB will contact other eNB for handover. The procedure shown in Figure 10 would start from step 1 in which S-eNB sends a **Handover request** to other T-eNB.

However, if at least one edge computer's load does not exceed its upper limit, the

Connect-permission will be sent back to T-eNB (step 3), and the T-eNB will establish an optimized path with C-eNB according to the C-eNB IP carried in **Handover request**. Before the handover finishes, User data sent by CN to UE via the S-eNB is temporarily saved in the buffer of S-eNB. T-eNB issues a **Handover request ACK** to S-eNB (step 4), and then S-eNB delivers an **eNB-firewall status** (including the PDCP SN, HFN and firewall URL) to T-eNB (step 5) where PDCP SN and HFN stand for SN ID and Hyper Frame Number in PDCP layer. After that, the eNB firewall establishment procedure will be activated. Meanwhile, S-eNB sends the buffered data to T-eNB through X2 interface, and T-eNB notifies MME (step 6) with a **Path switch request** (including TAI, and ECGI, the same as Binding update) about the fact that UE is handing over to eNB where TAI and ECGI stand for, Tracking Area Id and E-UTRAN CGI and informs the SGW with **Modify bearer request** (including the eNB's IP, TEIDs) about new address of eNB and TEID of user plane (step 7) where TEID standing for Tunnel Endpoint IDentifier. The SGW updates the instructions field of the UE's flow entry in its flow table with the new eNB IP. After the update, a **Modify bearer RSP** is sent back to MME (step 8). The MME issues a **Path switch ACK** (including TEID) (step 9) to notify the T-eNB that the path switch has been completed.

T-eNB returns an **UE context release** (including the UE's IP) telling S-eNB to release the resources reserved for serving this UE (step 10). S-eNB transmits a **Delete command** (including the UE's IP) to S-Keeper (step 11). S-Keeper then deletes the information about the UE from the EC table and the EC-status table, and issues a **Delete command** (including the UE's IP) to the S-edge computer (step 12) to remove this UE's firewalls settings and the data in the Firewall_address table.

On the other hand, if there is no X2 interface between S-eNB and T-eNB, all messages delivered between them go through MME, i.e., via S1-C interface.

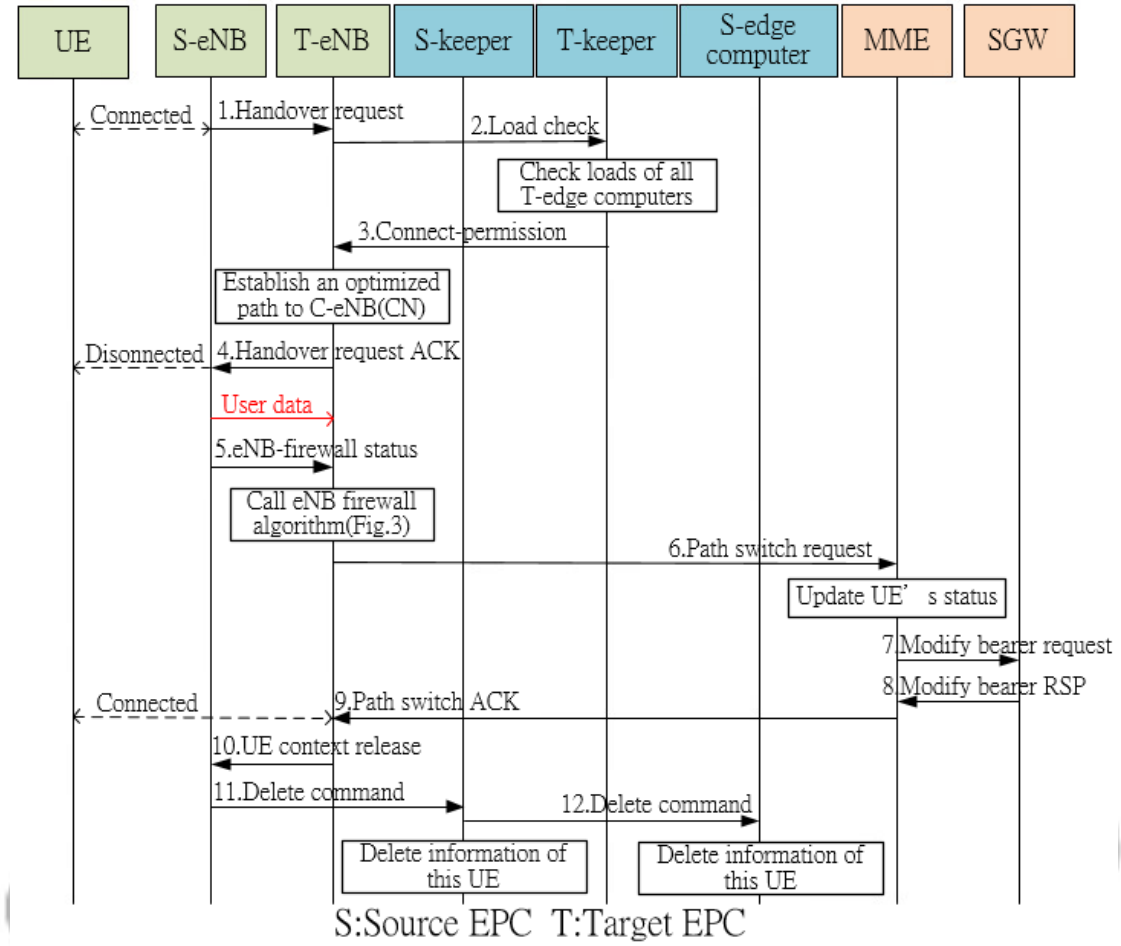

Figure 10. The sequence chart of Intra-MME handover.

4.2 Inter-MME handover (X2 handover)

The Inter-MME handover procedure is shown in Figure 11. When the eNB detects that the RSRP of an UE is lower than its predefined threshold, the handover procedure will be triggered. The four steps between **Handover request** (step 1) and **Connect-permission** (step 3) including "check loads" of all T-edge computers are the same as the four steps between step 1 and step 3 of the Intra-MME sequence chart illustrated in Figure 10. Because UE hands over to T-EPC, which will authenticate the UE with the help of H-EPC (step 4 and step 5). After the authentication, H-HSS replies an **Authentication data RSP** to T-MME through T-HSS (step 6) and T-MME passes **User authentication request** to T-eNB (step7). T-eNB then do EPS-AKA and establishes an

optimized path between it and CN to maintain the connection between CN and UE. After receiving a **Handover request ACK** (step8) from T-eNB, S-eNB sends the buffered data received from CN to T-eNB through X2 interface. After UE successfully hands over to T-eNB, User data is sent to UE by T-eNB.

Next, S-eNB sends a **Firewall information request** (including UE's IP) to S-Keeper to request the firewall URL of the UE (step9). After looking up the firewall URL in the EC table, S-Keeper returns a **Firewall information RSP** (including the UE's IP and firewall URL -- step10). S-eNB sends **eNB-firewall status** (including PDCP SN, HFN, firewall URL) to T-eNB (step11). T-eNB then establishes a firewall on one of its edge computers for this UE. The following steps from step12 to step14 are similar to step 10 to step 12 of the Intra-MME shown in Figure 10. We do not redundantly describe them.

In addition, similar to that of Intra-MME, if there is no X2 interface between S-eNB and T-eNB, all the messages transferred between them will go through S-eNB – S-MME – T-MME – T-eNB connection.

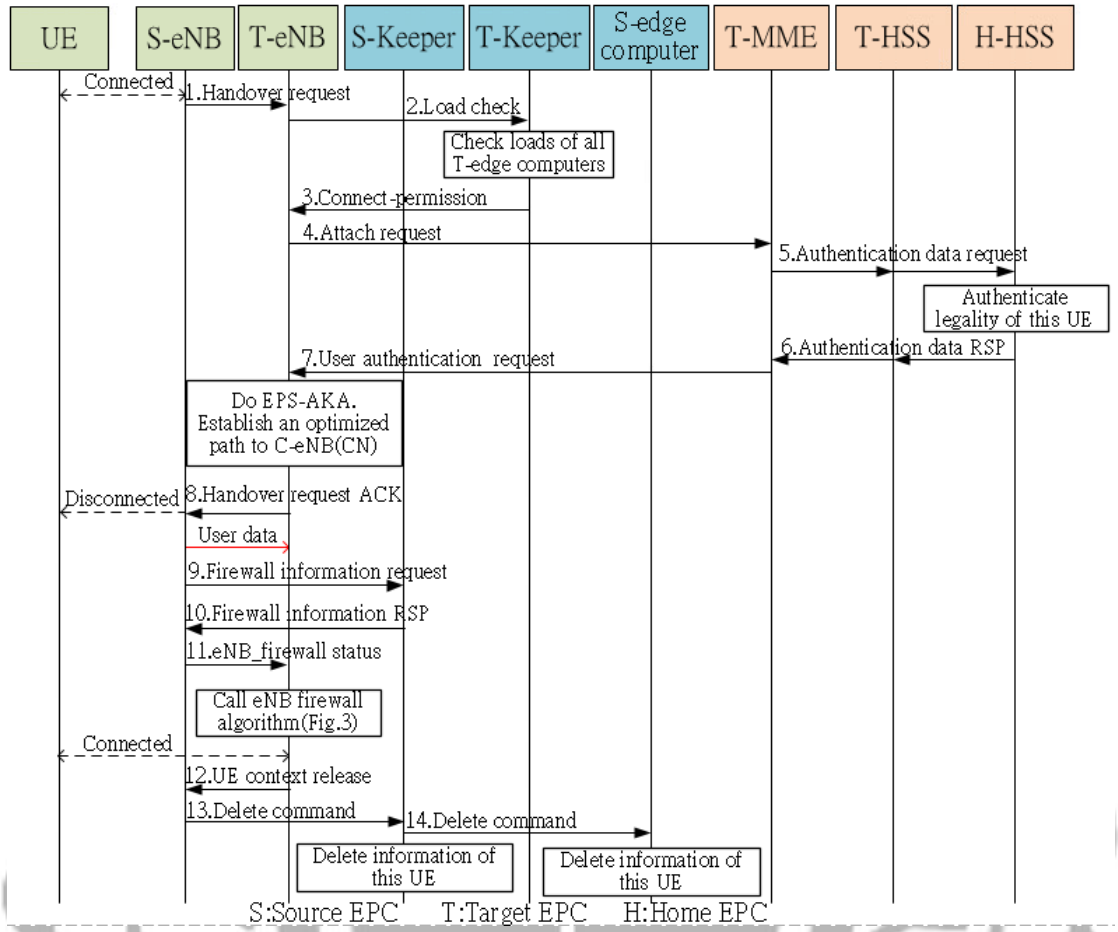

Figure 11. The sequence chart of Inter-MME handover.

4.3 Inter-SGW handover (S10 handover)

Figure 12 shows the sequence chart of Inter-SGW handover. When S-eNB needs to hand over, S-eNB sends a **Handover request** (including C-eNB's IP, ...) to S-MME. S-MME passes it to T-MME (step 1) via S10 interface, and T-MME sends it to T-eNB (step 2). After agreeing to handover, T-eNB sends a **Handover request ACK** to T-MME (step 3) through S1 interface. T-MME transmits an **Authentication data request** to the UE's Home network (H-HSS) through T-HSS for authenticating UE (EPS-AKA) and requesting firewall data (step 4). An optimized path between T-eNB and CN according to C-eNB's IP is also established. When receiving a **Handover request ACK** from T-MME (step 5), S-MME sends the **Handover command** (step 6) to S-eNB. The

User data sent by CN to UE through S-eNB are temporarily saved in S-eNB. After the establishment of a bidirectional channel between S-NB and T- eNB, S-eNB sends the User data to T-eNB via S-SGW and T-SGW. When receiving a **eNB status transfer** (including PDCP SN and HFN) sent by S-eNB via S-MME (step 7), T-MME forwards it to T-eNB (step 8). Then T-MME builds a firewall based on the SGW-firewall establishment procedure (Figure 3). T-MME sends an **UE context release** to S-eNB through S-MME (step 9)to release resources reserved for serving this UE. S-MME sends a **Delete command** to the S-Manager (step 10) and the S-SGW (step 11). S-Manager then deletes the information about this UE from its EC table (Table 1), and the UE Authentication table (Table 3). It also decreases the No of UE field of the edge computer that serves this UE in its EC-status table (Table 2) by one due to the UE's handover. The S-SGW deletes the tuple/entry of this UE from its flow table. S-Manager also transmits a **Delete command** (step 12) to S-edge computer to remove this UE's firewall and the tuple of this UE in the Firewall_address table.

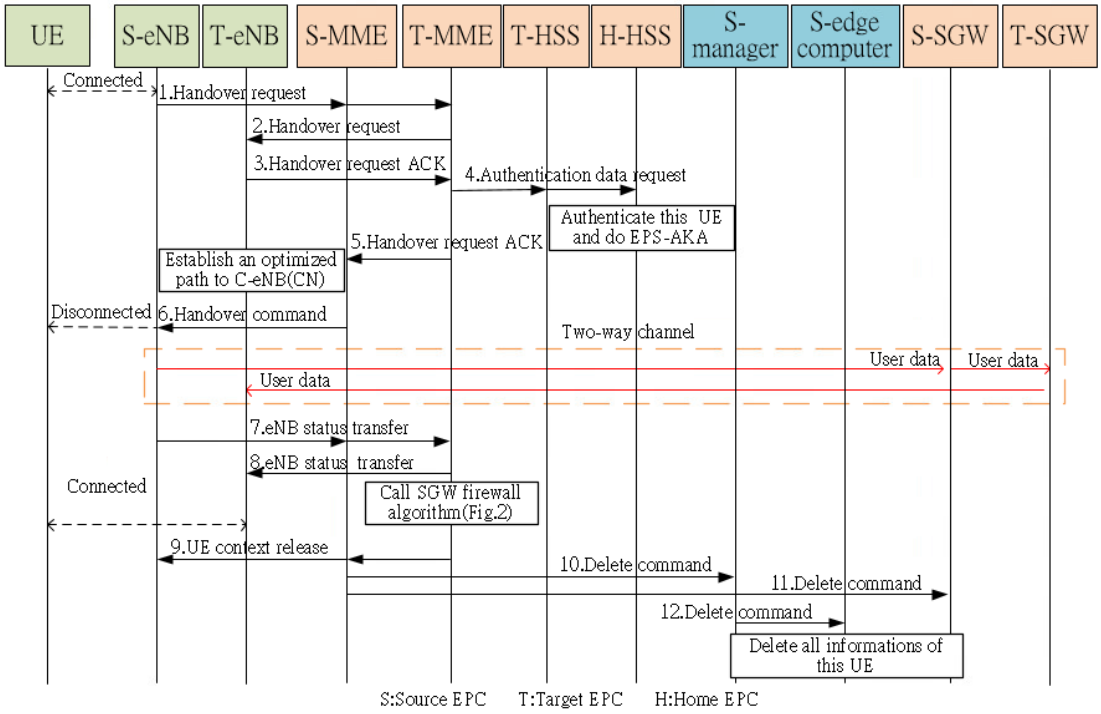

Figure 12. The sequence chart of Inter-SGW handover.

4.4 Untrusted Case

If the two EPCs, e.g., Q and R, are owned by the two untrusted telecom operators, when UE hands over between the two MMEs of Q and R, there is no S10 interface connecting the two MMEs or no X2 interface bridging the two eNBs. So, they need a third party C to deliver messages exchanged between the two EPCs. If both telecom operators have individually signed a contract with C in advance for collecting the information about the two operators' base stations and providing network services to their users, when UE of Q (or R) needs to hand over, UE's serving eNB, i.e., PMAG, can extracts the required information, e.g., the firewall URL or eNB status, from server of C. Figure 13 shows a part of network entities drawn from Figure 12 (Inter-SGW handover). Due to untrusted relationship between Q and R, S-MME needs to send a message, for example, a Handover request to C. C will pass this message to T-MME. The Handover request ACK, Firewall information request, etc. are the same. All are transferred to their destinations via C.

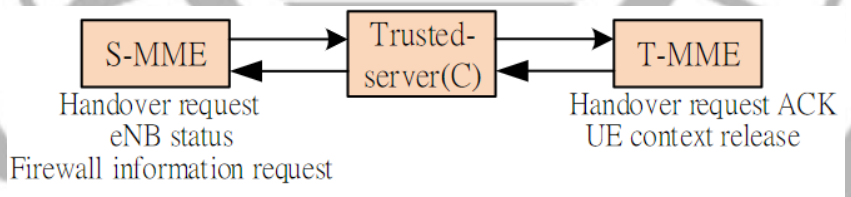

Figure 13. Messages transferred between two untrusted operators.

Chapter 5 Simulation and Discussion

In this chapter, we simulate our systems with the Mininet on ubuntu 14.04. The evaluated schemes include SGW firewall, eNB firewall, an EPC without a firewall (denoted by No-firewall) and the scheme using OpenvSwitch (denoted by OvS). In the OvS scheme, OpenvSwitch follows its processing rules stored in its flow table to drop malicious packets based on IPs of those packets flowing through this OpenvSwitch so as to achieve a simple firewall mechanism.

The evaluated metrics include Round Trip Times (RTT) defined as the time period from the time when source node sends a packet to destination node to the time when the source node receives reply from the destination node, drop rates defined as the number of packets dropped on the way to their destinations over the total number of packet sent by the source node, and throughputs defined as the number of bits received by the destination node per second. Three experiments are performed. The first evaluated the RTTs of the four schemes. The second (the third) measures their drop rates (throughputs).

Experiments are conducted given different bandwidths on different data rates.

5.1 Simulation setup

Mininet is a network emulator that provides virtual hosts, switches and controller to establish a virtual network system in which switches support the Openflow protocol. We used Wireshark, which is a network sniffer, to observe network statuses and behaviors. Packet transmission is implemented by using ping commands. The default payload of a ping packet in Mininet is 1500 bytes, i.e., a ping packet is 1514 bytes long in which 14 bytes is the length of ICMP header. Before our experiments, we need to

adjust the default size of a Maximum Transmission Unit (MTU) from 1500 bytes to 2020 bytes to generate different data rates. One of the reasons is that Mininet autonomously add an Ethernet header, i.e., which is also 14 bytes long, to a data packet. To avoid exceeding the packet length we choose, i.e., 2048 bytes, 28 bytes are first reduced from a 2048-bytes data packet. Consequently, a data packet will be 2034 (=2048-28+14) bytes long. In summary, as shown in Figure 14, a packet of 2048 bytes is divided into 2020 bytes and 28 bytes where 28 bytes are a part of the payload of ICMP packet, meaning that the 2048-bytes packet reproduces a small ICMP packet of 90 bytes (in which 14 bytes of ICMP header + 28 bytes of ICMP data + …) and an IPv4 data packet of 2034 (2020+14) bytes. The MTU settings for different data rates are listed in Table 7.

Also, during the ping process, RTT starts its counting when an ICMP is sent, and stops the counting when the ICMP reply packet header arrived at the source node. So the counted RTT time is actually a little shorter than its theoretical value due to ignoring the transmission time of the payload of the ICMP reply packet since the payload is small. Also, data rate is calculated only based on IPv4 packet which is 2034 bytes in length.

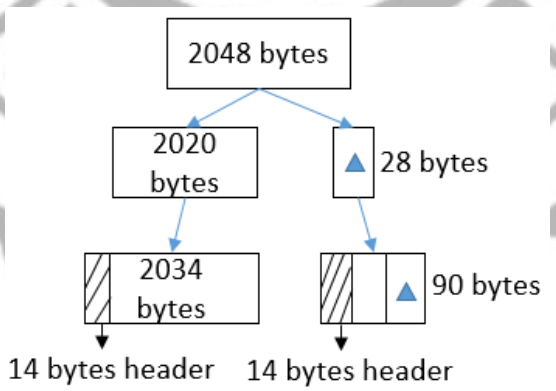

Figure 14. A packet divided into two parts by MTU setting

Table 7. MTU settings for on different data rates (Mbps : Mega bits per second).

Packet rate (Kbps)	16	40	80	160	320s	500
MTU (bytes)	2020	5100	10220	20460	40940	63980

During the experiments, packets are transmitted from UE to eNB. Bandwidth of the link connecting two arbitrary EPC entities is set to 800Mbps. The processing time individually consumed by an edge computer and a Keeper is listed in Table 8.

Table 8. The processing time of different network entities

Item	Time (ms)
The processing time of an edge computer in SGW firewall (eNB firewall).	5
In eNB firewall, the EC table look-up time consumed by Keeper	3

The real SGW firewall topology is shown in Figure 2. The simulation topology of the SGW firewall is illustrated in Figure 15. The UE is set to host1, eNB node and entities in EPC are represented by openflow switches (denoted by OvS1~OvS4). When SGW and edge computer deliver data packets to each other, these packets do not pass through Manager. The destination data packet network denoted by “Other core network”, is represented by host2. About the real eNB firewall, the topology is shown in Figure 4. The simulation topology of eNB firewall is shown in Figure 16.

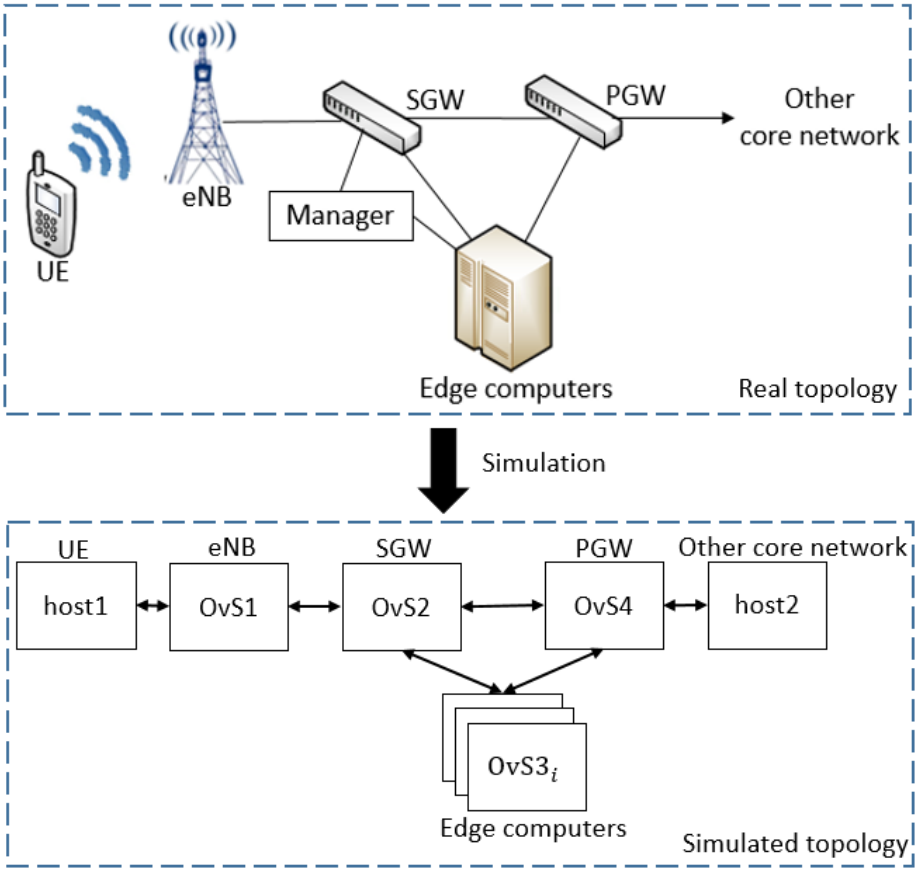

Figure 15. A simulated SGW firewall topology.

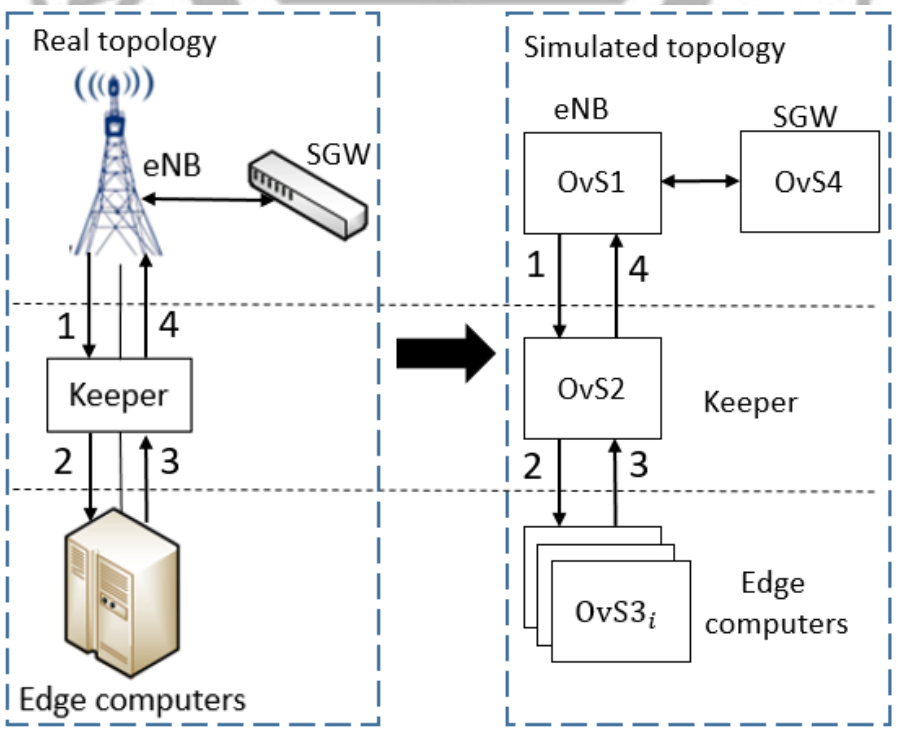

Figure 16. A simulated eNB firewall topology.

5.2 Round Trip Time

Before the first experiment, we send 100 packets from host1 to host2 shown in Figure 15 and measure their RTTs. The results are shown in Figure 17, in which RTT of the first packet (Packet-In) is 170ms, because this packet needs to be processed by the controller and waits for the controller to send flow-mod to a switch for installing processing rules into the flow table of this switch. The RTTs of the subsequent packets retain between 20 and 23ms.

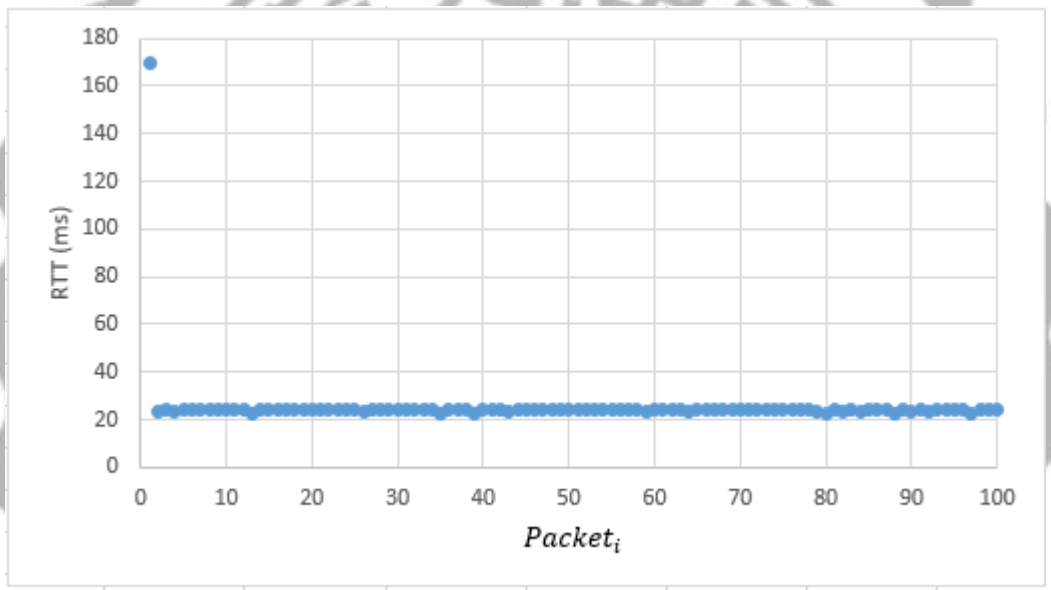

Figure 17. RTT of 100 packets and x-axis is packet ID.

We assume that this firewall only checks packet *p*'s header and then blocks *p* if *p* is a malicious packet. So the delay time for edge computer to filter packets does not significantly increase compared to that of No-firewall. Figure 18 shows the RTTs of the No-firewall given it different data rates, including 16, 40, 80, 160, 320 and 500Kbps, and different bandwidths, ranging between 1 and 64 Mbps. Figure 19 (Figure 20) illustrates the RTTs of SGW firewall (eNB firewall) given the experimental specifications the same as those of Figure 18. Figure 21 shows RTT of OvS scheme. Its RTTs are a little higher than those of No-firewall. The differences range between 0.5ms

and 2ms. Among the four figures, No-firewall's RTTs are the shortest. But it provides no security functions to protect its network system.

With the eNB firewall, the Keeper needs to spend some amount of time to check the EC table. So its RTT is the longest. When the bandwidth increases, the RTTs of the four schemes decrease sharply. With a firewall, packet transmission delay is almost conducted by components' data processing. Table 9 shows the percentages of processing delays resulting from different components of the corresponding firewall systems when data rate is 16Kbps and bandwidth is 64Mbps. It can be seen that a large proportion of RTTs is actually consumed by Keeper, edge computers and SGW. When the bandwidth increases, the amount of RTT reduces. When bandwidth is higher than 32 Mbps, RTTs approach their components' total processing delay. So in the future 5G environment, shortening components' processing delays is an important issue worth to study.

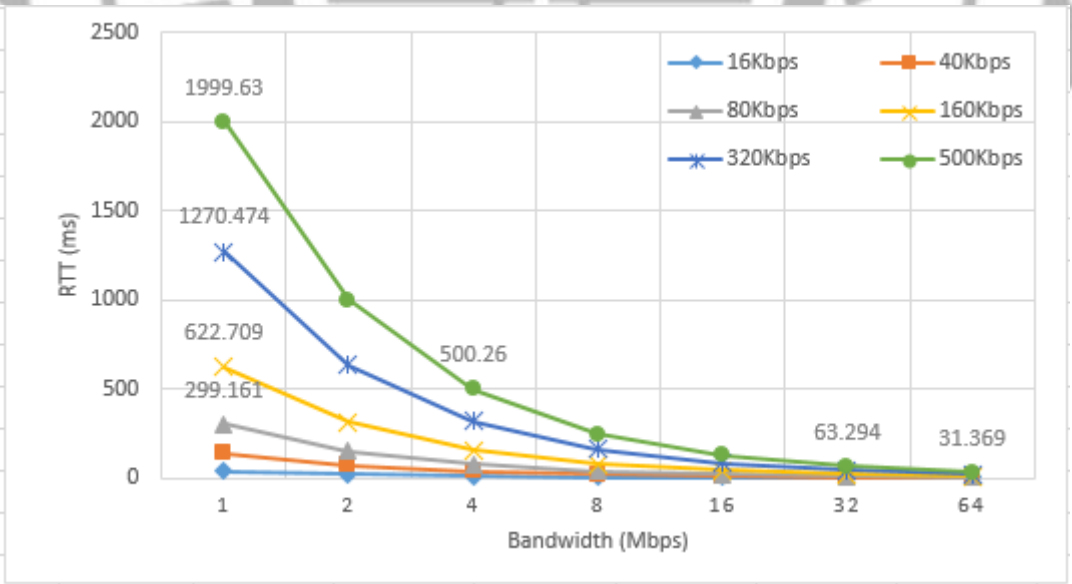

Figure 18. RTTs for forwarding packets through No-firewall.

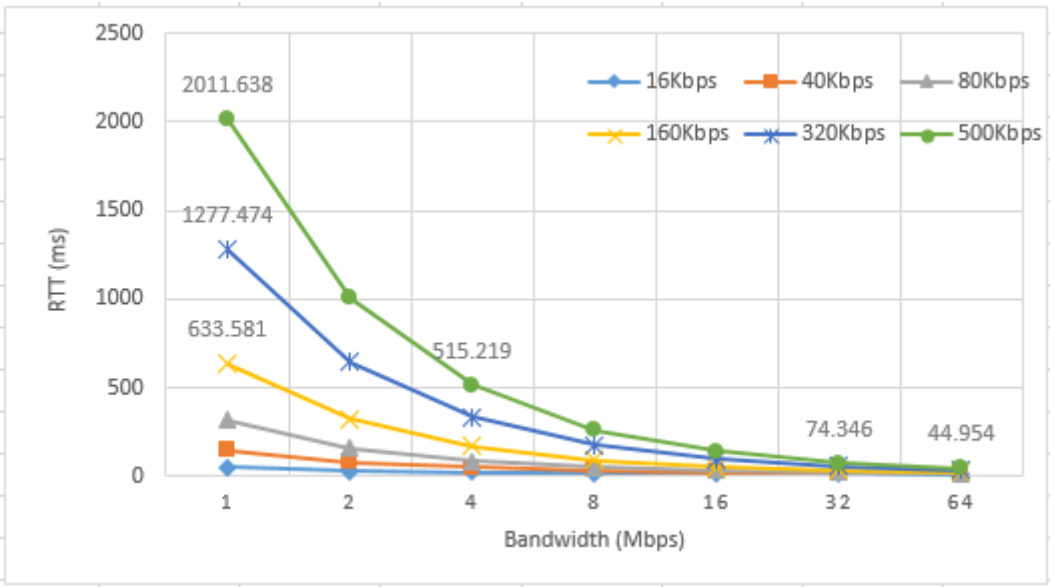

Figure 19. RTTs for forwarding packets via an SGW firewall.

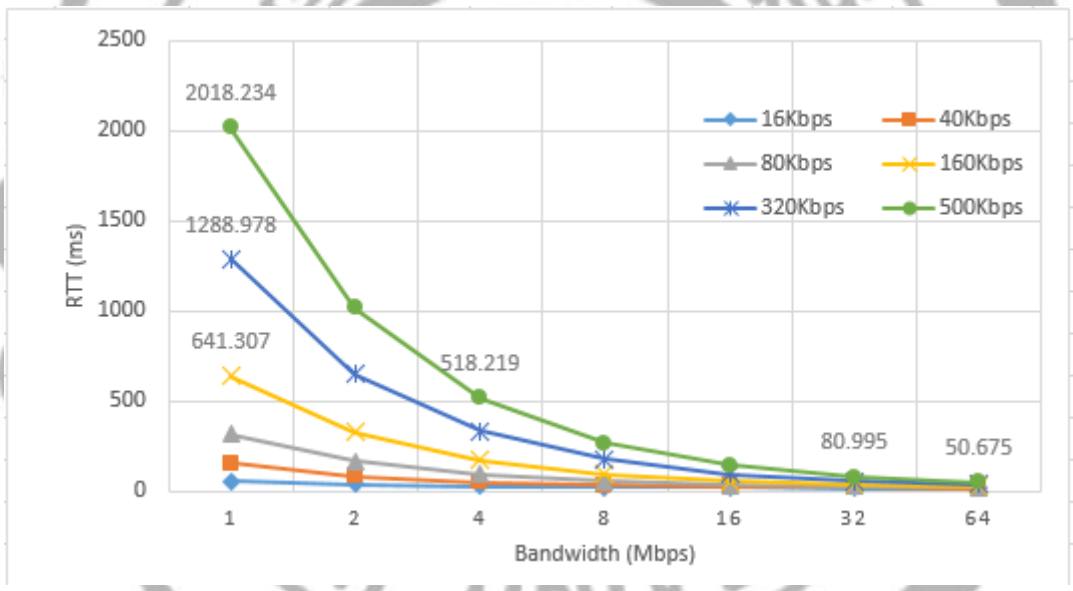

Figure 20. RTTs for forwarding packets via an eNB firewall.

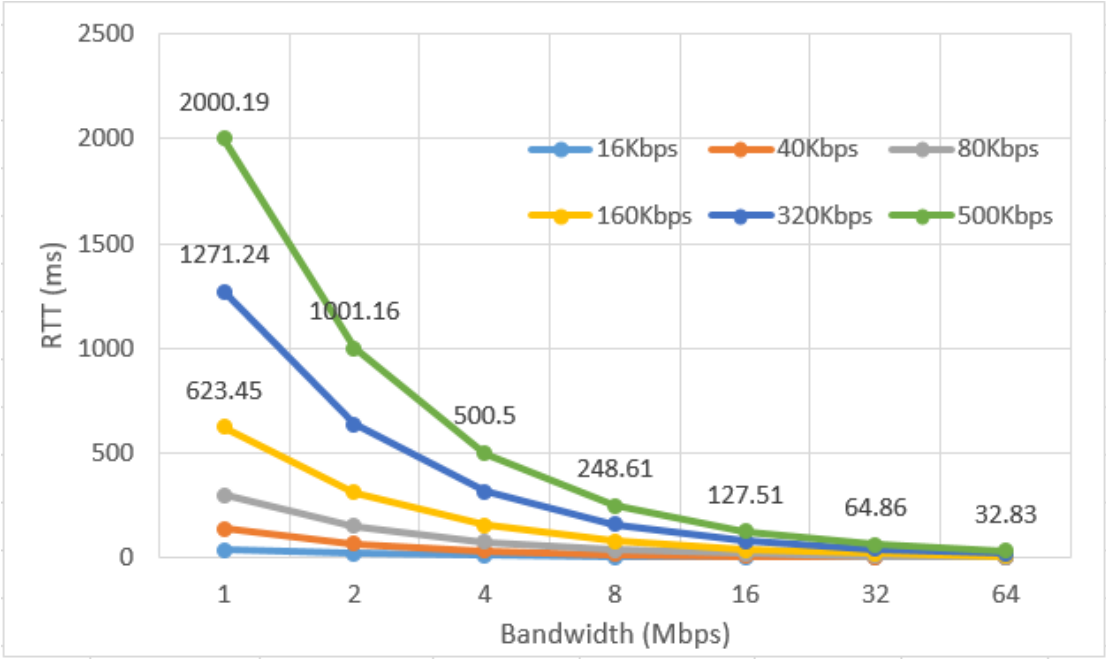

Figure 21. RTTs for forwarding packets in an OvS scheme.

Table 9. RTTs and percentages of function delays on Data rate =16Kbps and Bandwidth = 64Mbps.

Item	SGW firewall	eNB firewall	OvS
RTT (ms)	13.28	19.34	3.24
Delay conducted by Keeper and edge computer (ms)	10 (edge computer)	16 (Keeper+ edge computer)	-
Percentage of RTT delay conducted by Keeper and edge computer (%)	75.3(=10/13.28)	82.7(=16/19.34)	-

5.3 Drop rates

In the second experiment, we measure drop rates for the four tested schemes.

The experimental results on different bandwidths are shown in Figure 22, in which the bandwidths are individually equal to their data rates. Given low bandwidths, such as 1Mbps, 2Mbps, ... and 16Mbps, the amount of delivered data per unit of time is small, so the drop rate is almost 0. When bandwidth ranges between 32Mbps and 128Mbps,

due to the processing delay, a small amount of packets is lost. When bandwidth is higher than 256Mbps, the drop rates increase. This means some packets have waited for their services for a long time in the edge computer, causing that a large amount of packets which would like to enter the network's packet queue is dropped. On the other hand, because of the long processing delay of Keeper (3ms), edge computer (5ms), drop rates of the eNB firewall are higher than those of the other three schemes. The SGW firewall ranks the second.

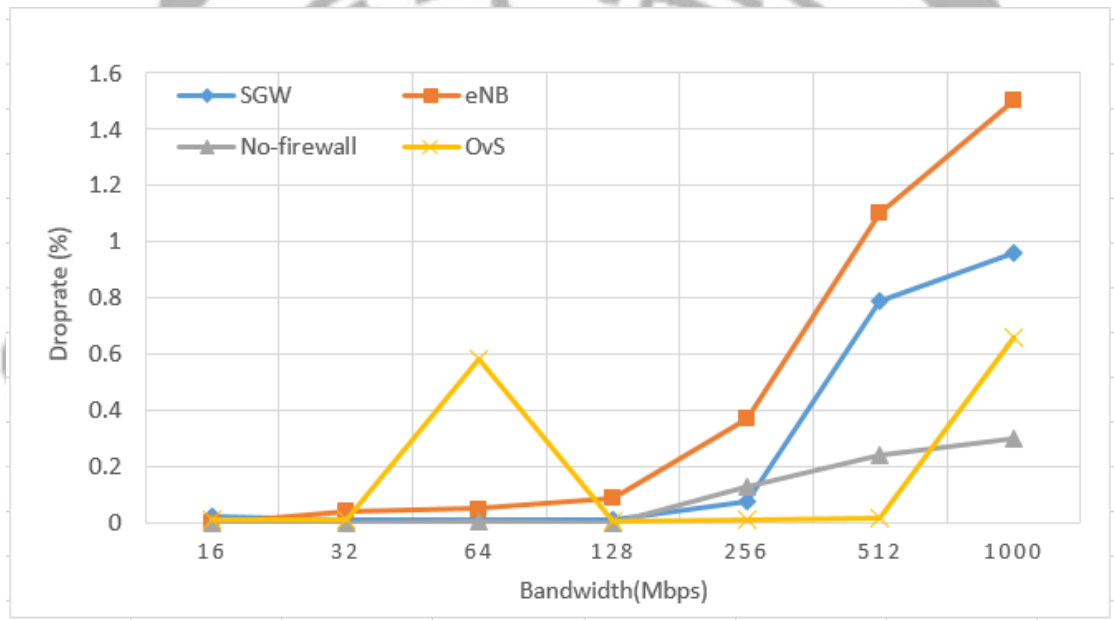

Figure 22. Drop rates for different tested schemes on data rate=bandwidth where bandwidth is 16, 32 …. 1,000 Mbps.

5.4 Throughputs

In the third experiment, we measure throughputs of the four tested schemes given different bandwidths. The measurement is performed in two cases. In the first case, there is only one source node and one destination node, denoted by 1 to 1. In the second case, these are 4 source nodes and 1 destination node, denoted by 4 to 1.

5.4.1 1 to 1

Figure 23 shows throughputs given different bandwidths, like 2Mbps, 8Mbps, ... and 1,000Mbps. It can be seen that the difference among the four schemes is limited. The throughputs of the eNB firewall (No-firewall) is the lowest (highest) because its network components' total processing delays are the longest (lowest).

Next, the processing delays of the edge computer are individually set to 5ms, 50ms, 250ms or 500ms. Those of Keeper in the eNB firewall scheme are set to 3ms, 30ms, 150ms or 300ms. Figure 24 shows the experimental results on bandwidths of 64 Mbps, 256 Mbps, and 1,000 Mbps, denoted by X-64, X-256 and X-1000, respectively, where X may be SGW firewall or eNB firewall. The processing delay of edge computer (Keeper) is still 5 (3) ms. It means that this time, only SGW and eNB firewalls are compared. Throughputs of these two firewalls significantly decrease when their network components' processing delays vary from 5ms to 50ms. When processing delay is 250ms, their throughputs are all less than 100Mbps, and SGW-1000's throughput is about 17.3Mbps. When processing delay is 500ms, throughputs of the eNB-1000 is less than 5Mbps. The processing time of the eNB firewall is about 1.6 (=(5+3)/5) times that of the SGW firewall where 3ms is the processing delay of the Keeper (there is no delay on Manager in the SGW firewall when data packets are sent), resulting in low throughputs.

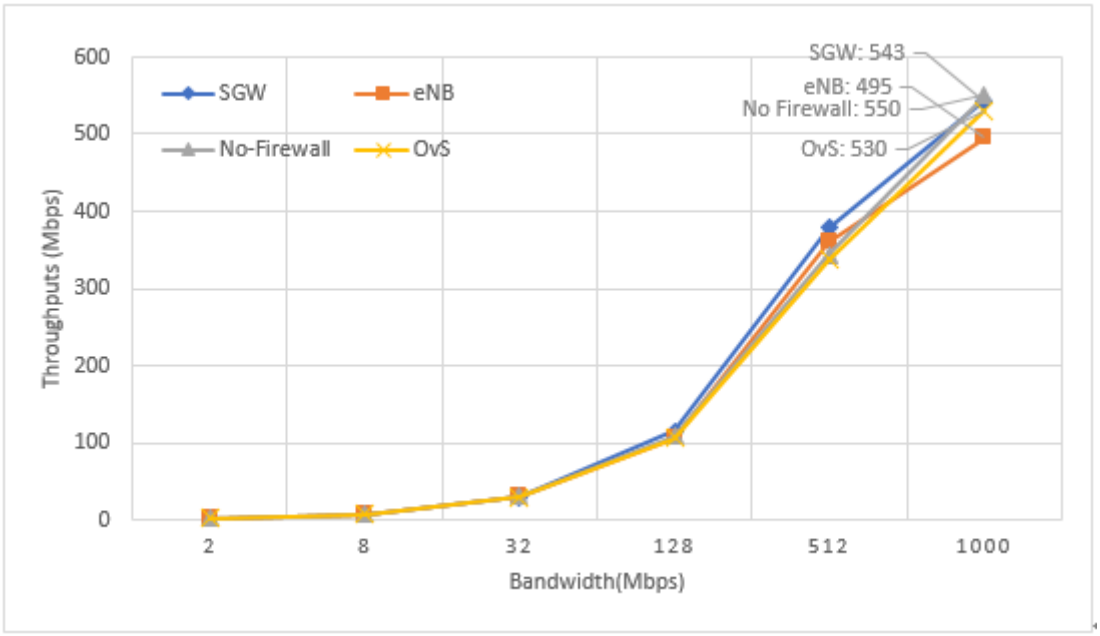

Figure 23. Throughputs of 1 to 1 on different bandwidths when the processing delay of an edge computer (Keeper) is 5 (3)ms

.

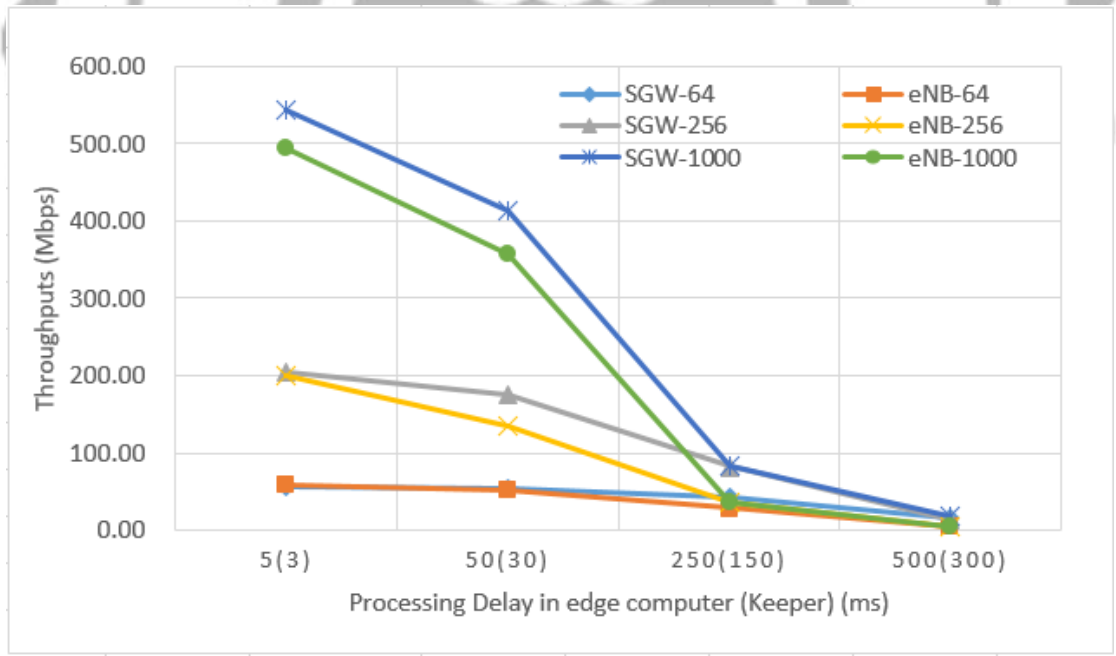

Figure 24. Throughputs of 1 to 1 on different processing delays of edge computer and Keeper where x(y) in x-axis represents an edge computer's and Keeper's processing delays are x and y, respectively.

5.4.2 4 to 1

Figure 25 shows the throughputs when four hosts transmit packets to one server at the same time on different bandwidths, ranging from 64 Mbps to 1,000 Mbps. Note that throughputs of the four tested schemes are not individually higher than 110Mbps, owing to the processing delay of SDN controller and switches, even though the available bandwidth is equal to or higher than 256 Mbps. The other reason is packet contention and collision since packets from different sources need to compete the channel of their common wireless environment.

On the other hand, Figure 26 shows that when processing delays are higher, the overall throughputs sharply reduce. The throughputs of SGW-1,000 on 500ms of processing delay is 6Mbps, and that of the eNB-1,000 is 3Mbps, indicating that edge computer's processing delay has a great impact on performance of the overall network.

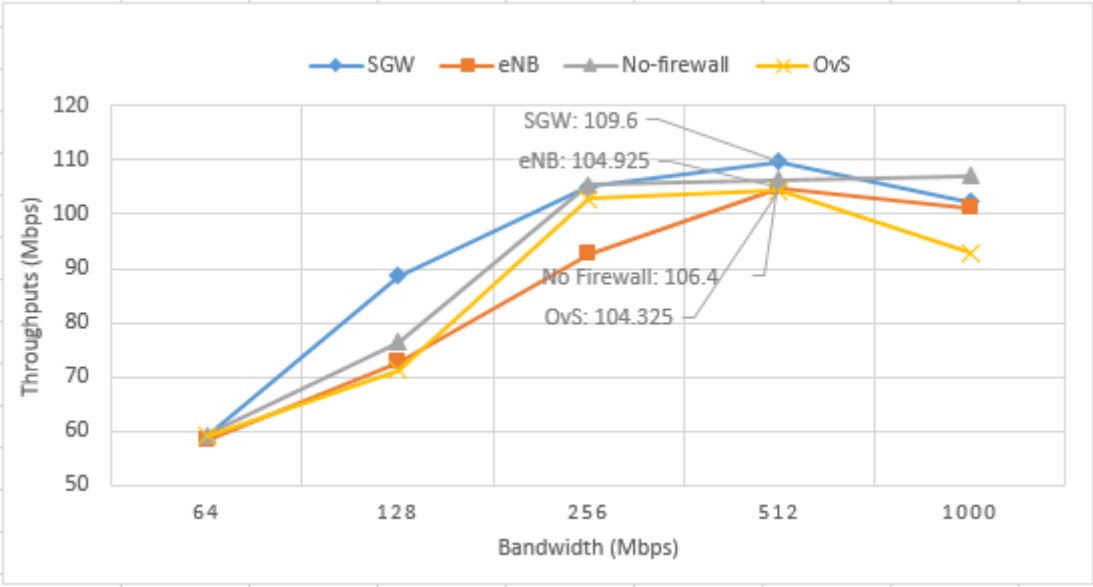

Figure 25. Throughputs of 4 to 1 on different bandwidths when the processing delay of an edge computer/ Keeper is 5/ 3ms.

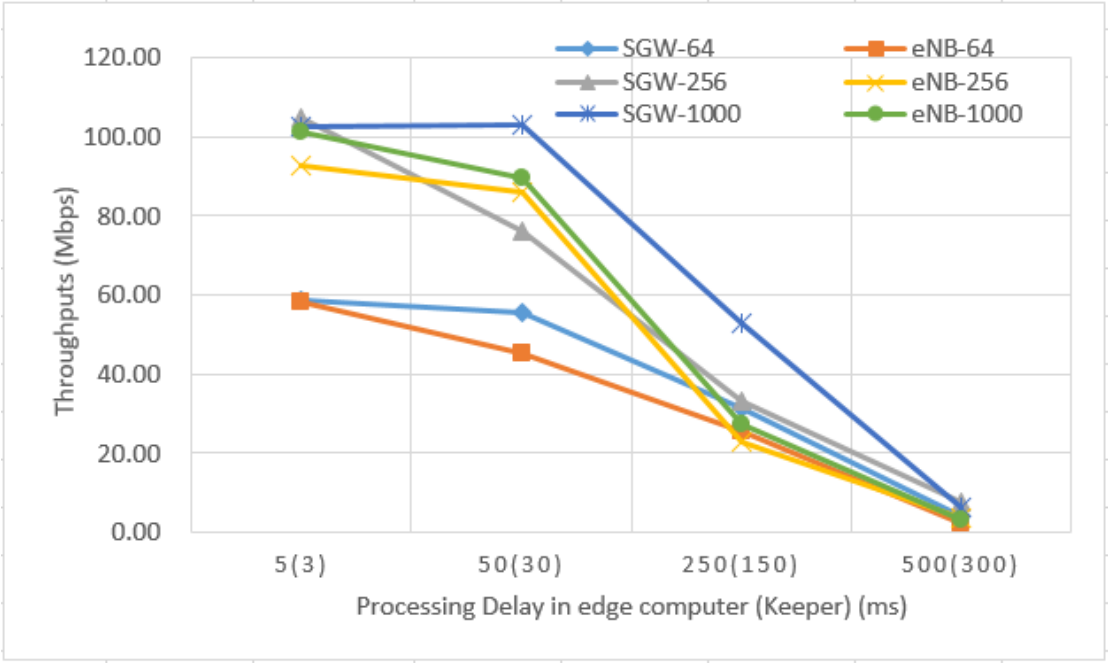

Figure 26. Throughputs of 4 to 1 on different function delays of edge computer and Keeper where x(y) in x-axis represents an edge computer's processing delay are x and y, respectively.

5.5 Costs of packets delivered

In this section, we count the amounts of packets delivered by our two schemes, i.e., SGW firewall and eNB firewall.

5.5.1 SGW/eNB firewall-EPS-AKA

In Table 10, items (a) and (d) in the second column only count the numbers of packets sent by the EPS-AKA which as shown in Figure 1 is 9 packets. In the SGW firewall, we need to record authentication command (step 2 of Figure 3), so it sends one more packet (a total of 10) than eNB firewall does (only 9 packets).

(1) Item (a) EPS-AKA in SGW firewall (10 packets):

The packets sent for EPS-AKA authentication in a SGW firewall include the messages delivered in step 1 (2 packets), step 3, step 5, step 7 (2 packets), step 9 (2

packets), step 11 of Figure 1 and step 2 of Figure 3. The cost is

3T(UE, eNB)+4T(eNB, MME)+2T(MME, HSS)+T(MME, Manager)

(2) Item (d) EPS-AKA in eNB firewall (9 packets):

According to Figures 1 and 5, the packets sent by EPS-AKA in an eNB firewall include the packets delivered in step 1 (2 packets), step 3, step 5, step 7 (2 packets), step 9 (2 packets), step 11 of Figure 1, excluding step 2 of Figure 3. The cost is

3T(UE, eNB)+4T(eNB, MME)+2T(MME, HSS)

Further, both items (b) and (e) integrate the steps of EPS-AKA and the steps of firewall establishment. That is, firewall is established when EPS-AKA is performed.

(3) Item (b) EPS-AKA + Firewall (integrated, 15 packets):

Due to the integration, EPS-AKA in a SGW firewall delivers 10 packets and the establishment of SGW firewall sends 5 packets (step 7 ~ step 11 of Figure 3), the total number of packet sent is 15. They are

3T(UE, eNB)+4T(eNB, MME)+2T(MME, HSS)+T(MME, Manager)+T(MME, SDN controller)+T(SDN controller, SGW) +2T(SGW, Manager)+T(Manager, edge computer)

(4) Item (e) EPS-AKA + Firewall (integrated, 12 packets)

The EPS-AKA in an eNB firewall delivers 9 packets and the establishment of eNB firewall transmits 3 packets (step 6 ~ step 8 of Figure 5). The total number of packets sent is 12. They are

3T(UE, eNB)+5T(eNB, MME)+2T(MME, HSS)+T(eNB, Keeper) +T(Keeper, edge computer)

(5) Item (c) EPS-AKA, Firewall (individually, 19 packets):

In item (c), EPS-AKA authentication process and the establishment of a SGW firewall are individually performed. It is the case that EPS-AKA is finished, transmitting 10 packets. UE sometime later requests firewall service. UE needs to execute steps 1 (2 packets), 3 and 4 and step 7 ~ step 11 of Figure 3, skipping steps 2, 5 and 6 of Figure 3, totally consuming 9 packets. So a total of 19 (=10+4+5) packets is

delivered. They are

4T(UE, eNB)+5T(eNB, MME)+4T(MME, HSS)+T(MME, Manager)
+T(MME, SDN controller)+T(SDN controller, SGW)
+2T(SGW, Manager)+T(Manager, edge computer)

(6) Item (f) EPS-AKA, Firewall (individually, 16 packets):

In the eNB firewall, EPS-AKA transmits 9 packets. After that UE requests firewall service, thus reperforming steps 1 (2 packets), 2 and 3 and step 6 ~ step 8 of Figure 5 to build its firewall, totally consuming 7 packets. So a total of 16 packets is transmitted. They are

4T(UE, eNB)+6T(eNB, MME)+4T(MME, HSS)+T(eNB, Keeper)
+T(Keeper, edge computer)

Table 10. The amount of packets delivered for performing EPS-AKA and/or establishment of a SGW or an eNB firewall.

Firewall	Components	No. of message sent
SGW	(a)EPS-AKA	10
	(b)EPS-AKA+ Firewall (integrated)	15
	(c)EPS-AKA Firewall (individually)	19
eNB	(d)EPS-AKA	9
	(e)EPS-AKA+ Firewall (integrated)	12
	(f)EPS-AKA Firewall (individually)	16

5.5.2 Costs for UE handover.

This section discusses the costs for UE handover in different environments.

A. Intra-MME

Table 11 shows the amounts of packets delivered for UE handover in different handover environments. In fact, Intra-MME means that UE does not change its serving EPC. No EPS-AKA authentication process is required. As shown in Figure 10, from

step 1 to step 12, totally 12 packets are sent for MME handover, plus step 7 and step 8 in Figure 5 for firewall establishment. So intra-MME requires 14 packets. They are

4T(S-eNB, T-eNB)+2T(T-eNB, T-Keeper)+2T(T-eNB, MME)
+2T(MME, SGW)+T(T-eNB, T-Keeper)+T(T-Keeper, T-edge computer)
+T(S-eNB, S-Keeper)+T(S-Keeper, S-edge computer)

B. Inter-MME

In the Inter-MME environment, if source and target EPC may belong to the same or different communication operator. For the former, no EPS-AKA is required, i.e., without EPS-AKA. The latter is one with EPS-AKA.

(a) Without EPS-AKA: 18 packets

As shown in Figure 11, from step 1 to step 14, there are totally 16 packets for UE handover since Authentication data request (Authentication data response) is passed to H-HSS (to T-MME) through T-HSS totally consuming 2 extra packets by these two signaling messages for firewall establishment request. Also, 2 packets, i.e., step 7 and step 8 in Figure 5 are delivered. According to Figures 5 (2 packets), and 11 (16 packets), they are

4T(S-eNB, T-eNB)+3T(T-eNB, T-Keeper)+2T(T-eNB, T-MME)+2T(T-MME, T-HSS)+2T(T-HSS, H-HSS)+3T(S-eNB, S-Keeper) +T(T-Keeper, T-edge computer)
+T(S-Keeper, S-edge computer)

(b) With EPS-AKA: 23 packets

If EPS-AKA is required, 5 steps from step 7 to step 11 in Figure 1 have to be performed (both step 7 and step 9 deliver 2 packets) after UE requests firewall service. So inter-MME consumes 23 (=18+5) packets for UE handover. They are

4T(S-eNB, T-eNB)+3T(T-eNB, T-Keeper)+5T(T-eNB, T-MME)+2T(T-MME, T-HSS)+2T(T-HSS, H-HSS)+3T(S-eNB, S-Keeper)+2T(UE, T-eNB)
+T(T-Keeper, T-edge computer)+T(S-Keeper, S-edge computer)

C. Inter-SGW

There are also two cases for Inter-SGW, i.e., with EPS-AKA and without EPS-

AKA.

(a) Without EPS-AKA: 23 packets

In the Inter-SGW environment, if EPS-AKA process is not required, there are 23 packets for UE handover, including step 1 to Step 12 in Figure 12, requiring a total of 16 packets since 4 packets in steps 1, 4, 7 and 9 are individually reproduced and delivered, plus 5 packets delivered in step 7 to step 11 in Figure 3. Also, a packet, like the Authentication Data Response shown in Figure 1 (step 5) is required to carry firewall-URL from H-HSS to T-HSS and T-HSS to T-MME, consuming 2 extra packets. So a total of 23 (=16+5+2) packets is required. According to Figure 1 (2 packets), 3 (5 packets), and 12 (16 packets), they are

4T(S-eNB, S-MME)+4T(S-MME, T-MME)+3T(T-eNB, T-MME)+2T(T-MME, T-HSS)+2T(T-HSS, H-HSS)+T(MME, SDN controller)+T(SDN controller, SGW) +2T(SGW, Manager)+T(Manager, edge computer)+T(S-MME, SManager) +T(S-MME, S-SGW)+T(S-Manager, S-edge computer)

(b) With EPS-AKA: 28 packets

If EPS-AKA is required, 5 steps from step 7 to step 11 in Figure 1 have to be performed, consuming 5 packets (each of step 7 and step 9 delivers 2 packets). There are a total of 28 (=16+5+2+5) packets. According to Figure 1 (2+5 packets), 3 (5 packets) and 12 (16 packets), they are

4T(S-eNB, S-MME)+4T(S-MME, T-MME)+6T(T-eNB, T-MME)+2T(T-MME, T-HSS)+2T(T-HSS, H-HSS)+2T(UE, T-eNB)+T(MME, SDN controller)+T(SDN controller, SGW)+2T(SGW, Manager)+T(Manager, edge computer)+T(S-MME, SManager)+T(S-MME, S-SGW)+T(S-Manager, S-edge computer)

Table 11. The amount of packets delivered for UE handover when SGW and eNB firewall are individually employed.

Item	No. of packets sent
Intra - MME	14 (without EPS-AKA)
Inter - MME	(a) 18 (without EPS-AKA)
	(b) 23 (with EPS-AKA)

Inter - SGW	(a) 23 (without EPS-AKA)
	(b) 28(with EPS-AKA)

Chapter 6 Conclusions and future studies

This paper proposes a function migration architecture which is performed by edge computers and which provides a mechanism for those functions or applications required by UEs to follow up the moving path of the mobile node to continue serving the UE. In fact, this architecture can apply to a 5G network system when it is available in the near future.

In this study, firewalls are utilized as an example. A firewall employed by an eNB or a SGW for packet filtering can improve network security level. Basically, an IDS/IPS can be utilized in the same method. When network entities fail, other components of the same function will take over for it so the provided network services will not be interrupted. Procedures for functional migration are also proposed for different environments helping UE's firewall to migrate to the target node when UE hands over.

Our experimental results show that the RTTs of our two schemes are small. Drop rate of eNB firewall is 1.2 (=1.5%-0.3%)% higher than that of No-firewall on Bandwidth = 1000Mbps (see Figure 22). Their throughputs are similar to those of No-firewall (see Figure 23). But our schemes provide higher security and more flexible employment when UE hands over to other eNB. However, from Figures 24 and 26, we can see that lower processing delays of our two schemes, i.e., SGW firewall and eNB firewall are essential to their performance since longer processing delays seriously impact system throughputs. As shown in Tables 10, the EPS-AKA and establishment of Firewall are individually performed, four more packets are required than that of EPS-AKA integrating firewall establishment, i.e., SGW firewall and eNB firewall. In Table 11, handover with EPS-AKA consumes four (five) more packets than without EPS-AKA in Inter-SGW (Inter-MME).

In the future, experiments for handover and entity failure mechanisms will be performed. We expect that feature migration mechanism can be applied to other 5G virtual functions, such as language translation and video streaming, allowing users to enjoy full 5G functions anywhere and anytime. We also like to derive the behavior model and reliability model for our two schemes so that users can realize their behaviors and reliabilities before using them. These constitute our future study.

Reference

[1] A. Ijaz, L. Zhang, M. Grau, A. Mohamed, S. Vural, A.U. Quddus, M.A. Imran, C.H. Foh and R.Tafazolli, "Enabling Massive IoT in 5G and Beyond Systems: PHY Radio Frame Design Considerations," *IEEE Access*, vol. 4, 2016, pp.3322-3339.

[2] J.M. Khurpade, D. Rao and P.D. Sanghavi, "A Survey on IOT and 5G Network," *International Conference on Smart City and Emerging Technology*, 2018, pp.1-3.

[3] D. Fang, Y. Qian and R.Q. Hu, "Security for 5G Mobile Wireless Networks," *IEEE Access*, Vol.6, 2018, pp.4850-4874.

[4] S. Vij and A. Jain, "5G: Evolution of a secure mobile technology," *International Conference on Computing for Sustainable Global Development*, 2016, pp.2192-2196.

[5] X. Liang and X.Qiu, "A software defined security architecture for SDN-based 5G network," *IEEE International Conference on Network Infrastructure and Digital Content*, 2016, pp.17-21.

[6] J. Zhao, S. Guo, K. Zheng, X. Niu and Y. Jiang, "An active defense model for Web Accessing DoS attack," *IEEE International Conference on Information Theory and Information Security*, 2010, pp314-318.

[7] B.A.A. Nunes, M. Mendonca, X.N. Nguyen, K. Obraczka and T. Turletti, "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," *IEEE Communications Surveys & Tutorials*, Vol.16, Issue 3, Third Quarter, 2014, pp.1617-1634.

[8] D. Kreutz, F.M. V.Ramos, P.E. Veríssimo, C.E. Rothenberg, S. Azodolmolky and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proceedings of the IEEE*, Vol.103, Issue 1, Jan. 2015, pp.14-76.

[9] C.J. Bernardos, A.L. Oliva, P. Serrano, A. Banchs, L.M. Contreras, H. Jin and A.J.C.

ZÚÑIGA, “ An architecture for software defined wireless networking,” *IEEE Wireless Communications*, Vol. 21, Issue 3, June. 2014, pp.52-61.

[10] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker and J. Turner, “OpenFlow: Enabling Innovation in Campus Networks,” *ACM SIGCOMM Computer Communication Review*, Vol.38, Number 2, April 2008, pp.69-74.

[11] Open Network Foundation https://www.opennetworking.org/, March 2019

[12]A. Basta, W. Kellerer, M. Hoffmann, K. Hoffmann and E.D. Schmidt, “A Virtual SDN-enabled LTE EPC Architecture: a case study for S-/P-Gateways functions,” *IEEE SDN for Future Networks and Services*, 2013, pp.1-7.

[13] M. Naeem, W. Ejaz, L. Karim, S.H. Ahmed, A. Anpalagan, M. Jo, and H. Song, ”Distributed Gateway Selection for M2M Communication in Cognitive 5G Networks,” *IEEE Network*, Vol.31, Issue 6, November/December 2017, pp.94-100.

[14] L. Velasco and M. Ruiz, “Flexible Fog Computing and Telecom Architecture for 5G Networks,” *International Conference on Transparent Optical Networks*, 2018, pp.1-4.

[15] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher and V. Young, “Mobile Edge Computing A key technology towards 5G,” *ETSI White Paper*, No. 11, September 2015, pp.1-16.

[16] X. Ge ,J. Ye ,Y. Yang and Q. Li, “User Mobility Evaluation for 5G Small Cell Networks Based on Individual Mobility Model,” *IEEE Journal on Selected Areas in Communications*, Vol.34, Issue 3, March 2016, pp.528-541.

[17] A. Arins, “Firewall as a service in SDN OpenFlow network,” *IEEE Workshop on Advances in Information, Electronic and Electrical Engineering*, Nov. 2015, pp.1-5.

[18] N. Zope, S. Pawar and Z. Saquib, “Firewall and load balancing as an application of SDN,” *IEEE Conference on Advances in Signal Processing*, 2016, pp.354-359.

[19] S.V. Morzhov and M.A. Nikitinskiy, “Development and Research of the PreFirewall Network Application for Floodlight SDN Controller,” *Moscow Workshop on Electronic and Networking Technologies*, 2018, pp.1-4.

[20] S. Asadollahi, B. Goswami, A.S. Raoufy and H.G.J. Domingos, “Scalability of software defined network on floodlight controller using OFNet,” *International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques*, 2017, pp.1-5.

[21] N. Gray, C. Lorenz and A. Müssig, “A priori state synchronization for fast failover of stateful firewall VNFs,” *IEEE NetSays*, March 2017, pp.1-6.

[22] S. Fichera, M. Gharbaoui, P. Castoldi, B. Martini and A. Manzalini, “On Experimenting 5G: Testbed Set-up for SDN Orchestration across Network Cloud and IoT domains,” *IEEE Conference on Network Softwarization*, 2017, pp.1-6.

[23] K. Zhang, S. Leng, Y. He, S. Maharjan and Y. Zhang, “Cooperative Content Caching in 5G Networks with Mobile Edge Computing,” *IEEE Wireless Communications*, Vol.25, Issue 3, June 2018, pp.80-87.

[24] P. Bellavista, A. Zanni and M. Solimando, “A Migration-enhanced Edge Computing Support for Mobile Devices in Hostile Environments,” *International Wireless Communications and Mobile Computing Conference*, 2017, pp.957-962.

[25] LibSVM, [online] Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm, March 2019

[26] Z. Guoa, R. Liub, Y. Xuc, A. Gushchind, A. Walide and H.J. Chaoc, “STAR: Preventing flow-table overflow in software-defined networks,” *Computer Networks*, Vol.125, 2017, pp.15-25.

[27] J. Liu, Y. Shi, L. Zhao, Y. Cao, W. Sun and N. Kato, “Joint Placement of Controllers and Gateways in SDN-Enabled 5G-Satellite Integrated Network,” *IEEE Journal on Selected Areas in Communications*, Vol.36, Issue 2, Feb. 2018, pp.221-232.