Instant Messaging Application for 5G Core Network

Fan-Hsun Tseng(✉), Tung-Yi Wu, I-Lung Chang, Chia-Chen Hsu, and Yi-Cen Chen

Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 70101, Taiwan
skittles2567@gmail.com

Abstract. Instant messaging (IM) has been widely used for many years since Internet technology developed and all-IP network architecture proposed. 5G mobile network launched worldwide a few years ago for pursuing ultra-low latency and high speed data transmission. In this paper, an IM application is developed in an open source 5G core (5GC) network framework, i.e. the free5GC. To validate the developed IM application operates properly in free5GC, we utilized the Tshark to examine and capture packets. Experimental results showed that the messages of the developed IM application passed through the free5GC network.

Keywords: 5G · Core Network · Instant Messaging

1 Introduction

The fifth-generation mobile networks (5G) is the latest generation of mobile communications technology with the purposes of high data rates, reduced latency, energy savings, reduced costs, increased system capacity, and large-scale device connectivity [1]. Due to the overwhelming network demands along with rapid development of network, Internet service providers worldwide upgrade the access network and core network to 5G. The cost of purchasing and upgrading network hardware has increased dramatically to meet the different functions of information system, such as computing, storage, networking, security. Therefore, some technologies such as Network Function Virtualization (NFV) [2, 3] and Software-Defined Networking (SDN) [4] are flourishing. With these technologies, the cost of investment become lower compared to the dedicated hardware provided by the supplier. In addition, NFV can automate a variety of management tasks [5] to reduce human errors, shorten deployment time, and provide better network productivity and efficiency. The above conditions can noticeably reduce the procurement and operating costs.

International Mobile Communications (IMT) standardized the IMT-2020 for pursuing 5G standard. In IMT-2020, the three ultimate goals of 5G mobile network are Enhanced Mobile Broadband (eMBB), Ultra-Reliable Low Latency Communications (uRLLC) and Massive Machine-Type Communications (mMTC). However, these three characteristics belong a trade-off problem and are partially contradictory. For instance,

Y. Chen et al. (Eds.): BICT 2023, LNICST 512, pp. 110–121, 2023.
https://doi.org/10.1007/978-3-031-43135-7_11

mMTC services require low energy consumption and ultra-low latency [6] but may not need high transmission data rate.

A mobile communication network system is composed of core network (CN), backbone network, Radio Access Network (RAN), and user equipment (UE), and so on. The core network is a network system consisting of various communication systems, shared by all users, and is responsible for transmitting backbone data to achieve large data transmission. The core network for 5G is known as the Next Generation Core (NGC). The 5G core network framework in this paper is based on the above-mentioned virtualized network technology. In the 21st century, instant messaging (IM) [7] has long become an indispensable tool in our life. Instant messaging is a system, application, service for communicating over the Internet, and the content transmitted can be text, files, and audio/video. Unlike the early days of email service, instant messaging emphasizes that communication between two parties is real-time. In recent years, IM service is no longer limited to text service, IM becomes more popular due to the rapid development of Voice over IP (VoIP) and video conferencing services [8].

In this paper, we propose and design an IM application with Python programming language. The designed IM application is implemented in an open source 5G core network project, i.e. the open-source project free 5G core network (free5GC throughout the paper). The designed IM application is implemented in the free5GC platform. Then, the designed IM application is validated its functionality through capturing packets by using Tshark. Experimental results showed and proved that the designed IM application works correctly in the free5GC platform. Further IM applications such as VoIP and video conferencing services can be founded on the proposed IM application in this paper.

The rest of the paper is organized as follows. Section 2 discusses core network applications and related works of instant messaging in 5G. In Sect. 3, the system model of free5GC and its installation are introduced. The design of the proposed instant messaging application is presented in Sect. 4. Finally, Sect. 5 concludes this work and provides its future directions.

2 Background and Related Work

There are existing three open source 5G core network projects, i.e., the free5GC [9] hosted by the Communication Service/Software Laboratory in National Yang Ming Chiao Tung University in Taiwan, the Open5GS [10] developed in Korea, and the Open Mobile Evolved Core (OMEC) [11] developed by the Open Networking Foundation (ONF). The three open source 5G core networks are compared as follows. The free5GC project is stated that it was the first open-source 5G core network in the world. The free5GC supports for setting up physical networks and low hardware requirements. The characteristic of Open5GS is that it can choose either 4G core-based configuration or 5G core-based configuration in the core network architecture, but the Open5GS needs higher hardware requirements than the free5GC. The particularity of OMEC project is that it can be integrated with other ONF projects, but suffers from the highest hardware requirements with compared to the free5GC and Open5GS. After evaluating the hardware requirement and system capability of the three open 5G core network projects, the proposed IM application is implemented in the free5GC platform.

The technology of network virtualization brings many advantages to the research field of 5G core networks, but may also results in security concerns [12]. In [13], the authors analyzed the 5G proprietary mobile network architecture based on the 3rd Generation Partnership Project (3GPP) technical specifications. They explored three different scenarios, and the establishment of a core network focus on private can effectively increase information security. In [14], the authors described the Open Network Automation Platform (ONAP) architecture supporting network slicing in 5G communication systems and its new capabilities for creating network slices. The ONAP not only manages and controls complex infrastructures based on network virtualization and software-defined networks but also slices the core network into multiple virtual core networks for supporting different functions. Each network slice is able to operate independently. In [15], the authors considered that the 5G system operators cannot know whether the methods of network traffic delivery such as switching and branching are performed correctly. Therefore, a 5G core network traffic monitoring system was proposed that allows operators to easily identify switching and branching traffic using traffic monitoring messages delivered by 5G core network functions. In addition to establish a secure private 5G core network environment, this paper also concentrates on developing a real-time communication application that can operate in this environment.

In [16], the authors believed that client-based IM applications have some drawbacks, such as scalability, single point of failure (SPOF), and vulnerability to Distributed Denial-of-Service (DDoS) attacks. As a result, they proposed a semantic point-to-point (P2P) approach to build an IM application. The designed P2P-based IM is a simpler design approach with the shortcoming of insufficient security. However, using the designed approach to implement on private networks can solved this problem with some restrictions. Due to the different platforms between users, private networks may suffer from information barriers after successful setup. Therefore, the authors in [17] proposed a technique to implement a cross-platform real-time communication system through HTTP, XMPP and TCP protocols, and through frameworks such as SSH, DWR and ExtJS. In addition, the authors studied the quality of service (QoS) mechanism when multiple terminals are accessed by multiple users at the same time.

In [18], the authors stated that IM communication is widely used but users' topics are diverse. In order to assist users in capturing the topics and contents of IM communication without reading all messages, they proposed a new method for detecting topics of instant messaging. The concept is suitable for certain situations, such as useless words keep appearing, the instant messages are very short, or multiple languages are used. In [19], the authors discussed the expansion and extension of IM technologies in Internet of Things (IoT). IoT IM aims to allow users communicate with each other through mobile phones or terminal equipment no matter where they are. Then, the authors proposed some key technologies to build a powerful real-time communication framework that can handle a large scale IoT platform. Furthermore, they have developed some applications based on above-mentioned technologies, e.g. the smart home system.

3 Experiment Environment Setup

In this section, the experimental environment of core network is introduced. The core network is implemented on a virtual machine and the used OS version is Ubuntu 20.04.3 LTS. In the beginning, we create the first virtual machine as a carrier for the core network (CN) and make sure it can connect to the external network. The ping command is applied to contact the Google server, and waits for the responded packets sent from Google server, which is shown as Fig. 1. Finally, we can see that the virtual machine has successfully connected to external network.

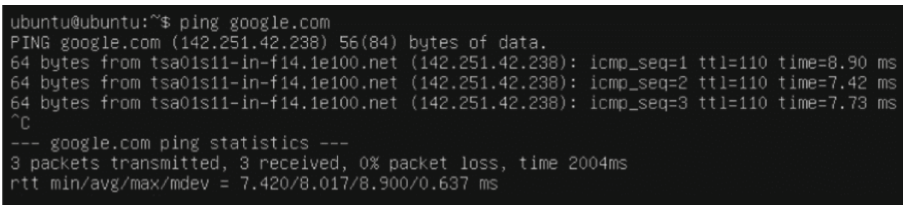

Fig. 1. Verify the status of network connection

After that, two virtual machines are created as a user equipment (UE) and a Next Generation Node B (gNB). Next, modify the host name and the local network IP address to facilitate instance identification during experiment process. The IP addresses of core network, user equipment and gNB are 192.168.56.101, 192.168.56.102 and 192.168.56.103, respectively. In addition, they are abbreviated as CN, UE102, and UE103 throughout experiments in this paper. Lastly, to validate the UE102 and UE103 are connected each other through core network rather than external network, the external network cards of them are disabled.

To construct the core network environment, the free5GC is downloaded from GitHub [20] with version 3.0.5. Note that some required packages should be downloaded and installed according to the readme document. The network parameters such as IP addresses of components in core network should be modified. Then, download the customized Linux kernel module as well as the gtp5g to handle packets. It is a core module developed for the Linux OS kernel to handle packets for the N4 interface which defined by 3GPP. After that, the core network can be executed and launched. Note that the connection status of core network and the configurations of network interface can be examined by using commands. Once the core network has been successfully established, we can set up the UE102 and UE103 in core network.

With respect to UE, the UERANISM project is downloaded and installed in the UE102. The UERANISM project is used to simulate UE and base stations (BSs). Before connecting to core network, execute the script in the virtual machine of core network and add set the IP address of UE102 as a subscriber. Then, execute the script of UERANISM project to boot user device and BS, and check the connection between cloud network and core network. A successful connection means that the overall core network environment has been successfully built up. Last, set up and check the UE103 by following the same above-mentioned steps to perform experiment environment.

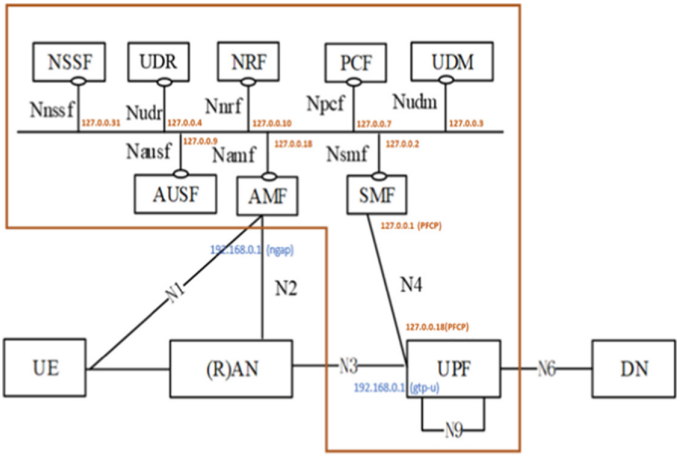

Fig. 2. The framework of free5GC

The framework and setting of the implemented free5GC is illustrated in Fig. 2. The default components of the free5GC are marked and framed in the orange lines. There are 9 components, i.e., the Network Slice Selection Function (NSSF), Unified Data Repository (UDR), Network Function Repository Function (NRF), Policy Control Function (PCF), Unified Data Management (UDM), Authentication Server Function (AUSF), Access and Mobility Management Function (AMF), Session Management Function (SMF), and User Plane Function (UPF). The functionality of these components is introduced as follows.

The NSSF is responsible for selecting a slice collection for serving users. On the basis of subscriber's data and location, the NSSF selects the AMF for users. The UDR aims to store subscription-related data, e.g., user contract, static contract, and policy data. The NRF is responsible for registering, managing, and checking the status of network. The PCF provides control and implementation of all network policies in control plane. The UDM is responsible for data management, such as identity authentication and certification repository. The UDM selects authentication method based on the subscriber identity and the configured policy, and calculates authentication data and key data as needed. The AUSF certificates the access authentication of subscribers from 3GPP and non-3GPP users.

In experiments, the settings of three components are modified, i.e., the AMF, SMF, and UPF. The AMF manages and authorizes users to access core network, such as the management of registration, connection, and reachability. The configuration of AMF is captured in Fig. 3. The IP address of the ngapIpList is changed from 127.0.0.1 to 192.168.56.101, which is marked with red line in Fig. 3. The SMF component establishes, modifies and release user sessions, and allocates and manages users' IPs. In addition, the SMF provides functions of charging and roaming. The configuration of SMF is captured in Fig. 4. The interface IP address is set to 192.168.56.101, which is marked with red line in Fig. 4. Lastly, the UPF component provides functions of routing, forwarding, packets

transmission, and the users' QoS management. The configuration of UPF is captured in Fig. 5. The IP address of the gtpu is set to 192.168.56.101.

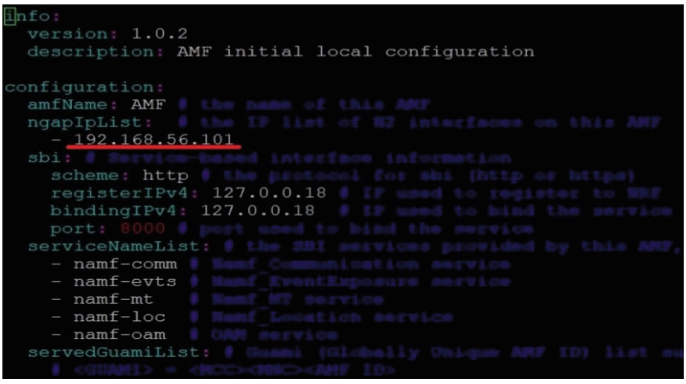

Fig. 3. Configuration of AMF

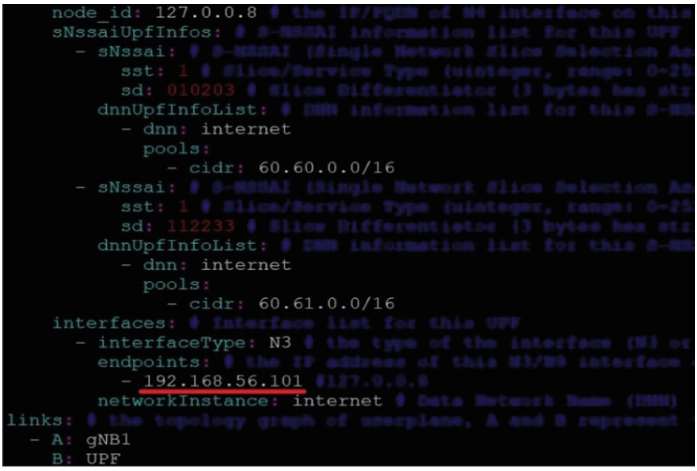

Fig. 4. Configuration of SMF

The network architecture of the experiment is captured in Fig. 6. We add the users as well as UE 102 and UE 103 and base stations to the free5GC framework. Note that to simplify the diagram, we illustrate the experiment architecture with core network and the UE102 only but there is another UE103 in practical experiment. The difference between UE102 and UE103 is IP addresses only.

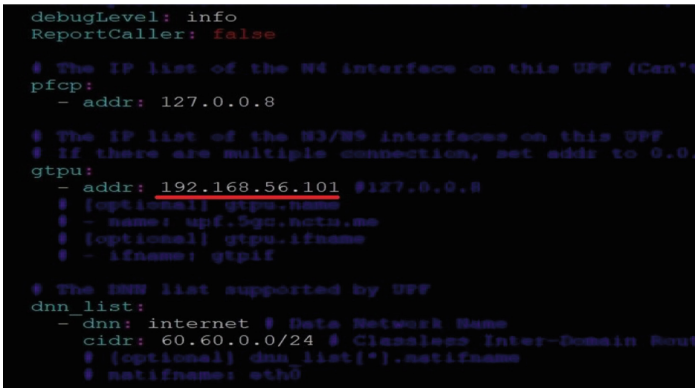

Fig. 5. Configuration of UPF

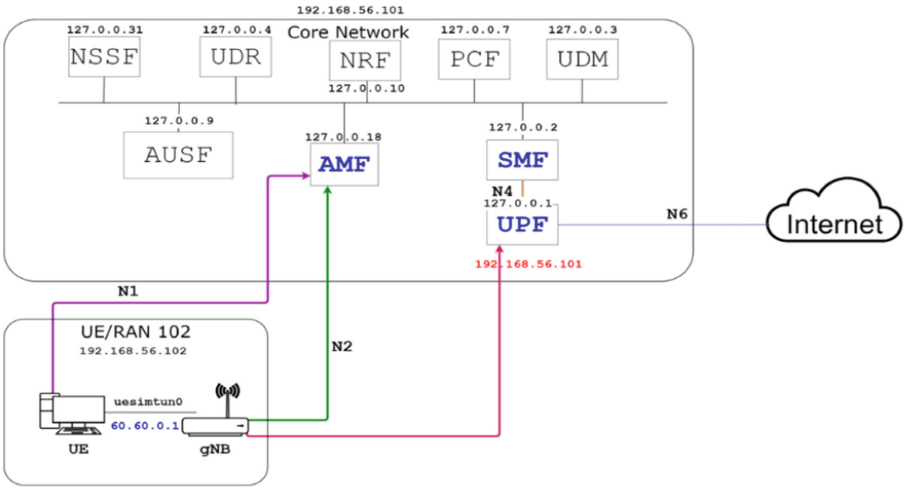

Fig. 6. The network architecture of experiment

4 Instant Messaging Application Deployment

In this section, the designed IM application is introduced. The IM application is developed by Python. The function of the program is to create a socket on both UEs so that they can communicate with each other through this socket communications. The operation steps of the IM application are described as follows.

1. Both UEs execute one after another and establish sockets on both UEs.
2. Enter one of the pre-defined mode codes on one side to activate IM types, e.g., "**m**" for text messages and "**f**" for file transmission. Then, the communication starts.

a). If text message mode "**m**" is selected, then UEs can start sending text messages. In Fig. 7, the top side is the sender and bottom side is the receiver. The sender sends "testing" message to receiver, then the receiver receives the message sent from sender with the beginning of "Incoming message".

b). If file transmission mode "**f**" is selected, then files can be transmitted through the designed IM application. In Fig. 8, the top side is the sender and bottom side is the receiver. The sender enters a file name "pi_200K.txt" but there is no such file, hence the file transmission is unsuccessful. The sender enters a file name "pi_204K.txt", and then the receiver receives a message "File transmission complete" represents that the file transmission is completed.

3. UEs can stop text or file mode by entering "stop", then the process of IM backs to step 2.
4. If any UE wants to terminate the IM application, they can enter "exit" while the IM application stage is in step 2.

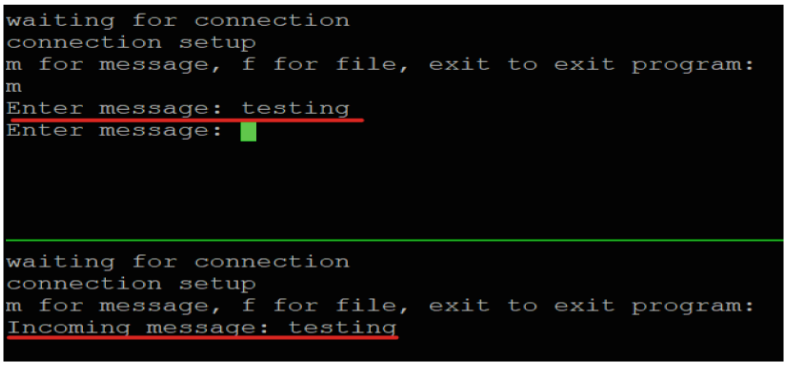

Fig. 7. Text mode of the IM

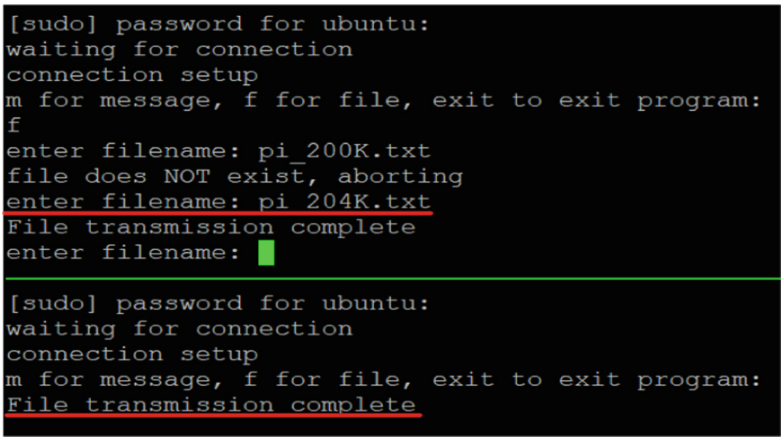

Fig. 8. File mode of the IM

In this paper, we utilized Tshark [21] to examine the packets of the proposed IM application pass through free5GC or not. The major goal is to confirm that the packets of IM application have passed through the 5G core network. Tshark is a packet analyzer tool in command line mode, which is similar to the tcpdump command. Tshark is developed by Wireshark without a Graphical User Interface (GUI).

The validation process is described as follows. Firstly, execute the IM application and make sure the transmission status works correctly. Then, access CN, UE102 and UE103 by using Secure Shell Protocol (SSH) and then monitor the network interfaces of three components by Tshark. Finally, execute IM application to transmit messages and then monitor the components through Tshark. The expected packet flow is illustrated in Fig. 9.

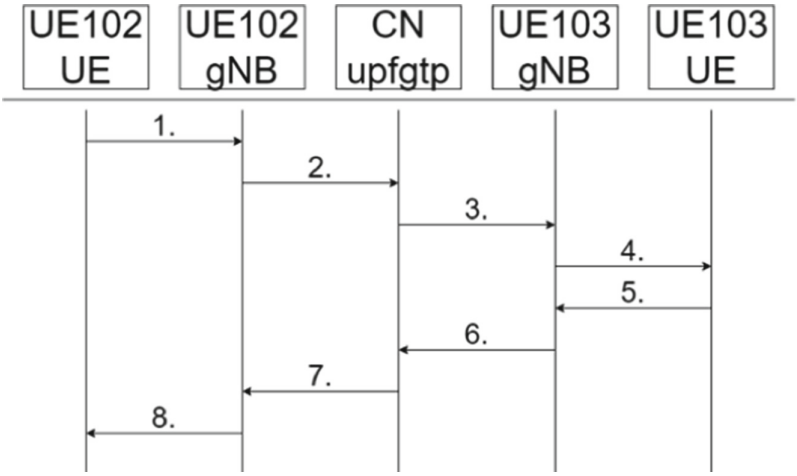

Fig. 9. Expected packet flow of IM message

Three network interfaces are monitored, i.e., the upfgtp in CN, the uesimtun0 tunnel of the UE102, and the uesimtun0 tunnel of the UE103. In all experiments, the IM communication initialized at UE102 sends a message to UE103. The packets captured in CN, UE102, and UE103 are shown as Figs. 10, 11, and 12, respectively.

The beginning field of the numbered sequences is the timestamp of a packet. According to these timestamps, it can be observed that the UE102 sends packets at the earliest on 07:50:44.338431664, which is shown in Fig. 10. Then, the UE 103 receives packets on 07:50:44.403228908, which is shown in Fig. 11. After that, the CN receives packets on 07:50:44.430627594, which is shown in Fig. 12. The results validate that the packets of the designed IM application passed through CN from UE102 to UE103 correctly. However, we found that the packets flow is not as expected.

The practical packet flow of the proposed IM is shown as Fig. 13. Note that the external network interfaces of UE102 and UE103 have been disabled to guarantee that the IM application works correctly in the free5GC. Although the practical packet flow does not match our expected packet flow, the packets of the IM application passed through CN based on the result of capturing packets in Fig. 12.

```
ubuntu@ue102:~$ sudo tshark -i uesimtun0 -t a
Running as user "root" and group "root". This could be dangerous.
Capturing on 'uesimtun0'
    1 07:50:44.338431664    60.60.0.1 → 60.60.0.2    TCP 59 60142 → 6000
0 [PSH, ACK] Seq=1 Ack=1 Win=510 Len=7 TSval=255852167 TSecr=2282608875
    2 07:50:44.339060211    10.0.2.15 → 60.60.0.1    ICMP 87 Redirect
         (Redirect for host)
    3 07:50:44.340207777    60.60.0.2 → 60.60.0.1    TCP 52 60000 → 6014
2 [ACK] Seq=1 Ack=8 Win=506 Len=0 TSval=2282631900 TSecr=255852167
```

Fig. 10. The monitored packets in UE102

```
ubuntu@ue103:~/socket$ sudo tshark -i uesimtun0 -t a
Running as user "root" and group "root". This could be dangerous.
Capturing on 'uesimtun0'
    1 07:50:44.403228908    60.60.0.1 → 60.60.0.2    TCP 59 60142 → 6000
0 [PSH, ACK] Seq=1 Ack=1 Win=510 Len=7 TSval=255852167 TSecr=2282608875
    2 07:50:44.403238973    60.60.0.2 → 60.60.0.1    TCP 52 60000 → 6014
2 [ACK] Seq=1 Ack=8 Win=506 Len=0 TSval=2282631900 TSecr=255852167
    3 07:50:44.404229436    10.0.2.15 → 60.60.0.2    ICMP 80 Redirect
         (Redirect for host)
```

Fig. 11. The monitored packets in UE103

```
ubuntu@free5gc:~$ sudo tshark -i upfgtp -t a
Running as user "root" and group "root". This could be dangerous.
Capturing on 'upfgtp'
    1 07:50:44.430627594    60.60.0.1 → 60.60.0.2    TCP 59 60142 → 60000
 [PSH, ACK] Seq=1 Ack=1 Win=510 Len=7 TSval=255852167 TSecr=2282608875
    2 07:50:44.430671043    10.0.2.15 → 60.60.0.1    ICMP 87 Redirect
         (Redirect for host)
    3 07:50:44.430687636    60.60.0.1 → 60.60.0.2    TCP 59 [TCP Retransm
ission] 60142 → 60000 [PSH, ACK] Seq=1 Ack=1 Win=510 Len=7 TSval=25585216
7 TSecr=2282608875
    4 07:50:44.431796082    60.60.0.2 → 60.60.0.1    TCP 52 60000 → 60142
 [ACK] Seq=1 Ack=8 Win=506 Len=0 TSval=2282631900 TSecr=255852167
    5 07:50:44.431822964    10.0.2.15 → 60.60.0.2    ICMP 80 Redirect
         (Redirect for host)
    6 07:50:44.431835915    60.60.0.2 → 60.60.0.1    TCP 52 [TCP Dup ACK
4#1] 60000 → 60142 [ACK] Seq=1 Ack=8 Win=506 Len=0 TSval=2282631900 TSecr
=255852167
```

Fig. 12. The monitored packets in CN

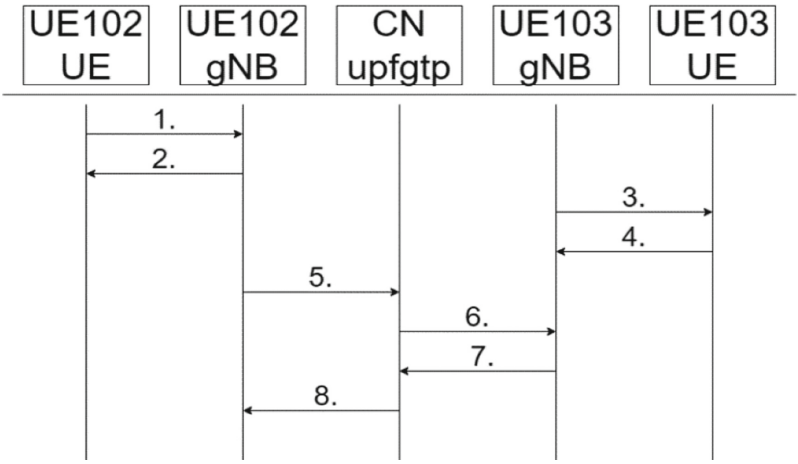

Fig. 13. Practical packet flow of IM message

5 Conclusion and Future Works

The 5G applications are on the fly based on the mobile network technology development. The paper presented an instant messaging application implemented by Python and realized in the free5GC framework. The experimental results showed that the packets of the IM application truly pass through the free5GC but the sequence of packets is not as expected. We presume that it may be attributed to the modification of free5GC's settings or the synchronization of core network and UEs. In the future works, we intend to fix the problem by synchronizing these components. Furthermore, we will try to propose VoIP and video conferencing applications.

References

1. Chen, C.-Y., Tseng, F.-H., Lai, C.-F., Chao, H.-C.: Network planning for mobile multi-hop relay networks: network planning for MMR networks. Wirel. Commun. Mob. Comput. **15**(7), 1142–1154 (2015). https://doi.org/10.1002/wcm.2396
2. Zhang, H., Wang, Y., Qiu, X., Li, W., Zhong, Q.: Network operation simulation platform for network virtualization environment. In: 17th Asia-Pacific Network Operations and Management Symposium (APNOMS), pp. 400–403. IEEE, Busan, South Korea (2015)
3. Tseng, C.-W., Huang, Y.-K., Tseng, F.-H., Yang, Y.-T., Liu, C.-C., Chou, L.-D.: Micro operator design pattern in 5G SDN/NFV network. Wirel. Commun. Mob. Comput. **2018**(3471610), 1–14 (2018)
4. Tseng, F.-H., Chang, K.-D., Liao, S.-C., Chao, H.-C., Leung, V.C.M.: sPing: a user-centred debugging mechanism for software defined networks. IET Networks **6**(2), 39–46 (2017)
5. Wollschlaeger, M., Sauter, T., Jasperneite, J.: The future of industrial communication: automation networks in the era of the internet of things and industry 4.0. IEEE Indust. Electron. Magaz. **11**(1), 17–27 (2017)
6. Sabuj, S.R., Ahmed, A., Cho, Y., Lee, K.-J., Lo, H.-S.: Cognitive UAV-aided URLLC and mMTC services: analyzing energy efficiency and latency. IEEE Access **9**, 5011–5027 (2021)

7. Dhir, A., Kaur, P., Rajala, R.: Continued use of mobile instant messaging apps: a new perspective on theories of consumption, flow, and planned behavior. Soc. Sci. Comput. Rev. **38**(2), 147–169 (2020)
8. Kaufmann, K., Peil, C.: The mobile instant messaging interview (MIMI): Using WhatsApp to enhance self-reporting and explore media usage in situ. Mobile Media Commun. **8**(2), 229–246 (2020)
9. free5GC. http://www.free5gc.org/. Accessed 21 Nov 2022
10. Open5GS. http://open5gs.org/. Accessed 21 Nov 2022
11. Open Mobile Evolved Core. http://opennetworking.org/omec/. Accessed 21 Nov 2022
12. Jang, H., Jeong, J., Kim, H., Park, J.: A survey on interfaces to network security functions in network virtualization. In: IEEE 29th International Conference on Advanced Information Networking and Applications Workshops, pp. 160–163. Gwangju, South Korea (2015)
13. Guo, Y., Zhang, Y.: Study on core network security enhancement strategies in 5G private networks. In: IEEE 21st International Conference on Communication Technology (ICCT), pp. 887–891. Tianjin, China (2021)
14. Lee, W., Na, T., Kim, J.: How to create a network slice? - a 5G core network perspective. In: 21st International Conference on Advanced Communication Technology (ICACT), pp. 616–619. PyeongChang, Korea (2019)
15. Kim, E., Choi, Y.: Traffic monitoring system for 5G core network. In: Eleventh International Conference on Ubiquitous and Future Networks (ICUFN), pp. 671–673. Zagreb, Croatia (2019)
16. Huang, L.: Instant messaging based on semantic P2P network. In: Fourth International Conference on Networking and Distributed Computing, pp. 33–35. Los Angeles, CA, USA (2013)
17. Fu, C., Tang, Y., Yuan, C., Xu, Y.: Cross-platform instant messaging system. In: 12th Web Information System and Application Conference (WISA), pp. 27–30. Jinan, China (2015)
18. Zhang, H., Wang, C., Lai, J.: Topic detection in instant messages. In: 13th International Conference on Machine Learning and Applications, pp. 219–224. Detroit, MI, USA (2014)
19. Su, K., Cheng, H., Wang, H., Lv, R.: Instant messaging application for the internet of things. In: IEEE International Conference Computer Science and Service System (CSSS), pp. 166–169. Nanjing, China (2012)
20. Open source 5G core network base on 3GPP R15. http://github.com/free5gc/free5gc. Accessed 21 Nov 2022
21. Tshark. http://tshark.dev/. Accessed 21 Nov 2022