Demonstration of Containerized Central Unit Live Migration in 5G Radio Access Network

Shunmugapriya Ramanathan[#], Abhishek Bhattacharyya[•], Koteswararao Kondepu[•],
Miguel Razo[#], Marco Tacca[#], Luca Valcarenghi[$], Andrea Fumagalli[#]
[#]Open Network Advanced Research lab, The University of Texas at Dallas
[•]Indian Institute of Technology Dharwad, Dharwad, India
[$]Scuola Superiore Sant'Anna, Pisa, Italy
e-mail:sxr173131@utdallas.edu, k.kondepu@iitdh.ac.in

***Abstract*—The 5G Radio Access Network (RAN) architecture provides a split option, whereby a gNodeB Central Unit (gNB-CU) is connected to one or more gNB-Distributed Units (gNB-DUs). The CU is in turn connected to the 5G Core Network (CN) and its functions can be virtualized through software containers. This demonstration showcases live migration of a containerized Central Unit (CU) component in a Cloud-native 5G network without loss of service. In terms of resiliency, virtual function live migration can circumvent the failure of the server hosting the gNB-virtualized CU (gNB-vCU) that would otherwise cause an interruption of user-plane (UP) traffic and disconnection of User Equipment (UE). The proposed gNB-vCU container live migration technique reduces the end-user service temporary downtime by 50% when compared to the traditional backup/restore option.**

***Index Terms*—5G, virtual RAN, docker container, container migration, CRIU.**

I. Introduction

While transitioning from 4G to 5G deployments, network providers must leverage both Software-Defined Networking (SDN) and Network Function Virtualization (NFV) concepts to implement cost-effective and scalable cloud-native 5G network solutions. With a cloud-native deployment a number of networking functions are implemented as software-based Virtual Network Functions (VNFs). Realizing that a monolithic implementation of these VNFs usually requires long time in both deployment and upgrade cycles [1], an alternative approach based on Containerized Network Functions (CNFs) is gaining momentum as it is believed to accelerate software implementation of network functions. The resulting CNFs can be easily managed through container platforms such as Kubernetes or Dockers. CNF is suitable to support the so-called New Radio (NR) access technology in the Next Generation Radio Access Network (NG-RAN) [2]. In NR, the next generation NodeB (gNB) functions are split into distinct — and possibly containerized — network entities [3], [4] such as: the Radio Unit (RU), the Distributed Unit (DU), and the Central Unit (CU).

Despite their potential benefits, software-based network functions are more vulnerable to failures than traditional middleboxes[1]. CNF platforms often require additional virtual layers between CNFs and the underlying hardware. Any misconfiguration of these intermediate layers may lead to CNFs failures. In addition, the reliability of dedicated hardware is inherently superior to that of commodity servers.Therefore, guaranteeing high availability of CNFs — which is essential to providing reliable network services — brings new challenges. Deploying backups is a well-established and robust method to improve the availability of CNFs. However, it comes with the extra cost of fully duplicated resources.

Realizing the importance of offering virtualized NR solutions with built-in capability for CNF live migration, this work describes an experimental setting that is designed to achieve this goal. Specifically, this demo focuses on container live migration of the gNB-vCU, i.e., migrating the gNB-vCU from one host to another to provide improved resiliency in mobile communication. Docker-based CNF migration is achieved through Checkpoint/Restore In Userspace (CRIU) [5]. The experimental setting is obtained by leveraging open software and standard solutions whenever possible and by implementing additional custom software packages when that is necessary to complete the required NFV/SDN architecture. All hardware components are commercially available. Our ultimate objective is to validate the feasibility and compare the performance of a few plausible NFV/SDN architectures, which provide docker container live migration of gNB-vCU with minimal service temporary downtime to the mobile user.

II. vRAN Live Migration Mechanisms

We first discuss the difference between *Stateless* and *Stateful* container live migration. We then describes our design process.

A. Background

State of a container refers to the changeable condition of the application during its run time that includes the results of the internal operations, interactions with other applications, the contents of memory or temporary storage or files operations. In the *Stateless* migration type, the state of the container is lost during migration, which results in restarting the container application from scratch at the destination server (node) and the deletion of the old container at the source node. In the *Stateful* migration type, both the volatile and persistent states of the container are migrated to the destination node. The

[1]A single gNB-CU is connected to multiple gNB-DUs, becoming a single point of failure for the User Plane (UP) of many User Equipment (UE) devices

Open Containers Initiative (i.e., runC) [6] is a popular container engine solution that achieves stateful migration through CRIU [5].

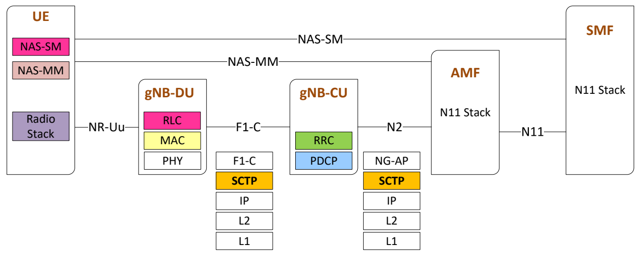


Fig. 1: 5G Control Plane Protocol Stack

B. 5G Control Plane Protocol Stack

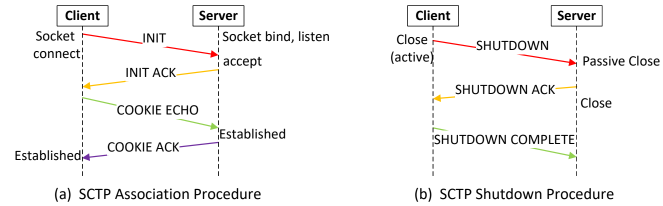
5G network supports two distinct planes, namely Control Plane (CP) and User Plane (UP). The radio resource control (RRC) protocol carries out the UE attach and other procedures that require signaling messages in the CP to be exchanged by the 5G Core Network, gNB, and UE. The attach procedure assists the UE to create and maintain the UP for the user-data transactions processed by protocols such as UDP and IP. As shown in Fig. 1, the gNB-CU CP makes use of Stream Control Transmission Protocol (SCTP) to exchange N2 control messages with the Access and Mobility Management Function (AMF) in the 5G Core Network. Similarly, the SCTP transport layer protocol is used to exchange F1-C control messages with the gNB-DU.

C. SCTP Handshake

SCTP is a transport-layer protocol that ensures reliable, in-sequence transport of data. SCTP association establishment requires the four-way handshake procedure in Fig. 2(a). During the gNB-vCU startup, the gNB-vCU acts as a client and sends the INIT control message to AMF (the server) to initiate the association. After verifying the authenticity of the state cookie using the secret key, the server allocates the resource for the client (the gNB-vCU). Similarly, during the gNB-vDU startup, the gNB-vDU acts as a client and establishes the association with the gNB-vCU (which acts as the server now). SCTP also supports graceful termination of an association through the submission of a SHUTDOWN request message as shown in Fig. 2(b). This SHUTDOWN request can be initiated from either side of the endpoints. The SCTP association and termination procedures are applied during the startup and termination of a gNB-vCU application, respectively.

D. Proposed Solution

The widely used container migration software CRIU does not support SCTP, despite a number of research efforts that have tried to overcome this limitation [7]. In our design, for achieving successful gNB-vCU docker container migration, we implemented a custom SCTP support in CRIU through Linux kernel changes as described next. During the container checkpoint at the source node, CRIU stores the SCTP socket endpoint details (IP address and port number) of the connected AMF and gNB-vDU. The customized Linux kernel at the source node does not send the SCTP-SHUTDOWN message to gNB-vDU and AMF. During the container restoration at the destination node, the gNB-vCU recovers the SCTP endpoint connection details from the checkpoint metadata. Consequently, the customized Linux kernel at the destination node will maintain the SCTP association with the gNB-vDU and AMF endpoints since the connection details are already available and no SHUTDOWN notification was sent earlier.



(a) SCTP Association Procedure (b) SCTP Shutdown Procedure

Fig. 2: SCTP handshake procedure

III. Demo System Architecture

Fig. 3 shows the demo setup. The OpenAirInterface (OAI) 5G mobile software stack is used to implement the 5G New Radio (NR) components. The OAI 5G-NR provides different split options as defined in [8]. Here, the $F1$ (i.e., Option 2) split interface is applied to connect gNB-vDU and gNB-vCU. The implemented 5G core network (CN) component comprises: Unified Data Repository (UDR), Unified Data Management (UDM), Authentication Server Function (AUSF), NF Repository Function (NRF), Access and Mobility Management Function (AMF), Session Management Function (SMF), and User plane Function (UPF) (i.e., SPGW-U). These functions are deployed as multiple docker containers using the OAI 5G CN source [9]. The 5G NR and CN components run on the CloudLab federated testbed [10].

gNB-vCU Live Migration: Two container migration scenarios are demonstrated: (i) intra-data center live migration and (ii) inter-data center live migration. As shown in Fig. 3, the connectivity between the 5G New Radio and Core Network components is implemented through an Open-Virtual Private Network (Open-VPN) tunnel. This option enables us to configure experiments that require resources from multiple testbeds. The demo is conducted as follows.

The OAI-based 5G NR and CN modules are installed in the federated testbed. The NR setup contains emulated nrUE, gNB-vDU, and gNB-vCU (as individual containers), whereas the 5G CN runs its components as separate container entities. A custom implemented GUI is used to trigger the gNB-vCU live migration, by launching a collection of shell scripts that runs in the backend.

UE service downtime and *migration time* are two key performance indicators that are monitored during our demonstration. The *UE service downtime* is defined as the time required to regain UE service connectivity from the moment the gNB-vCU CNF is disconnected from the network. The *migration time* is defined as the time that is required to migrate the gNB-vCU CNF from the source to the destination server.

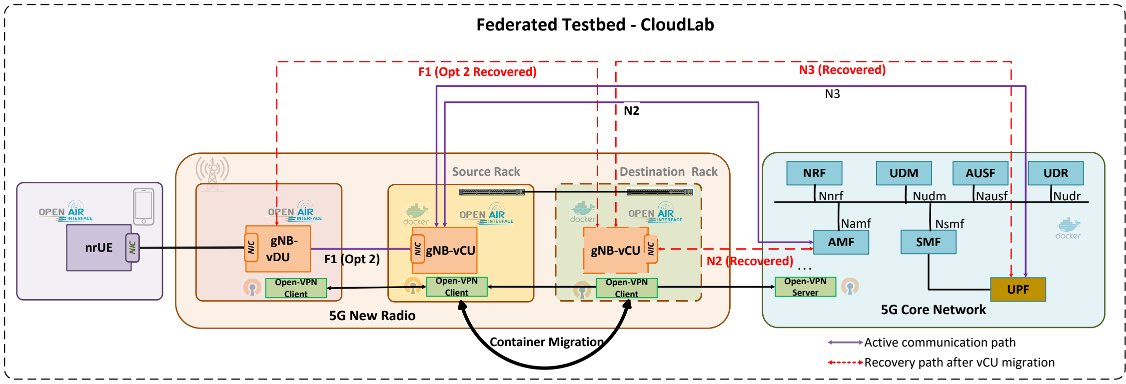


Fig. 3: Demo Setup

IV. Experimental Results

Ping messages from the UE to the Core Network interface are sent every $1s$ during the experiment. The *UE service downtime* is measured as the elapsed time between the last ping request sent to the 5G Core Network GTP interface (at the User Plane) from the UE before the gNB-vCU CNF failure and the first successful ping reply received after the gNB-vCU component is migrated.

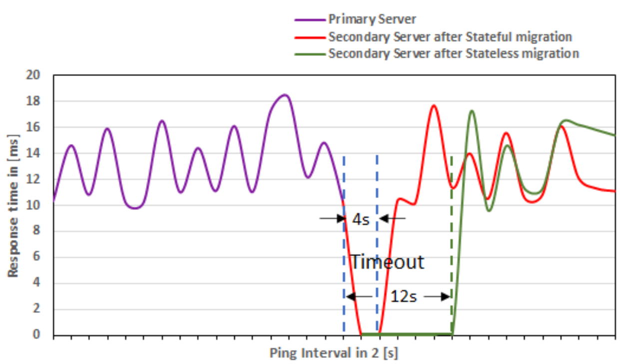


Fig. 4: UE Service Downtime

The average measured service downtime in the intra-data center scenario is shown in Fig. 4. The results show that the gNB-vCU CNF migration can be performed without service loss. As shown in Fig. 4, the proposed gNB-vCU *stateful* migration causes a *UE service downtime* of 4s that is one third when compared to *stateless* migration of 12s. This *service downtime* can be further optimized with more advanced container migration approaches and server configurations [4]. A total migration time is observed around 4s during the gNB-vCU stateful migration in the intra-data center model. Key performance indicators for the inter-data center migration time will be discussed while running the live demo at the conference. However, the intra-data center migration time will depends on the location of secondary server. The open-source software tool Zabbix [11] will be used to monitor these performance indicators along with the UE service functionality.

V. Summary

In this demo paper we report a successful live-migration of the OAI gNB-vCU module running as a container from a source to a destination server without incurring loss of UE service. The gNB-vCU *stateful* migration is achieved by means of both the CRIU module and a few custom modifications that we apply to the Linux kernel to overcome the current lack of CRIU support for SCTP, which is used by gNB-vCU to communicate with both gNB-DUs (downstream) and AMF (upstream). Compared to the *stateless* migration of gNB-vCU (which does not require our custom software changes in the Linux kernel) the *stateful* migration yields a threefold reduction of the UE service downtime when running the experiment on the CloudLab federated testbed.

Acknowledgment

This work is supported in part by funding from the ECSEL JU grant agreement No 876967 (BRAINE project), the JU receives support from the EU Horizon 2020 research and innovation programme and the Italian Ministry of Education, University, and Research (MUR). This work is also received funding from DST SERB Startup Research Grant (SRG-2021-001522), the SGNF project (“Reliability Evaluation of Virtualised 5G”), and NSF grant CNS-1956357.

References

[1] M. JayaKumar, “Why Use Containers and Cloud-Native Functions Anyway?” [Online]. Available: https://www.intel.com/content/www/us/en/communications/why-containers-and-cloud-native-functions-paper.html
[2] 3rd Generation Partnership Project, “Study on new radio access technology; radio access architecture and interfaces,” *Technical Specification Group Radio Access Network*, vol. 2.0.0, 2017.
[3] F. Giannone, K. Kondepu *et al.*, “Impact of virtualization technologies on virtualized ran midhaul latency budget: A quantitative experimental evaluation,” *IEEE Commun. Letters*, vol. 23, no. 4, pp. 604–607, 2019.
[4] S. Ramanathan, K. Kondepu, M. Razo, M. Tacca, L. Valcarenghi, and A. Fumagalli, “Live migration of virtual machine and container based mobile core network components: A comprehensive study,” *IEEE Access*, vol. 9, pp. 105 082–105 100, 2021.
[5] CRIU Community, “Checkpoint/Restoration In UserSpace (CRIU),” 2019. [Online]. Available: https://criu.org/.
[6] runc blog. [Online]. Available: https://www.docker.com/blog/runc/
[7] Jesutofunmi, 2019. [Online]. Available: http://rvs.unibe.ch/research/pub_files/Tofunmi_Thesis_.pdf
[8] ITU-T, “Transport network support of IMT-2020/5G,” ITU-T Technical Report, Feb. 2018.
[9] OpenAirInterface. [Online]. Available: https://gitlab.eurecom.fr/oai/cn5g
[10] D. Duplyakin *et al.*, “The design and operation of CloudLab,” in *Proceedings of the USENIX Annual Technical Conference (ATC)*, Jul. 2019, pp. 1–14.
[11] zabbix. [Online]. Available: https://www.zabbix.com/whats_new_6_0