Experimental Demonstration of 5G Virtual EPC Recovery in Federated Testbeds

K. Kondepu•, F. Giannone•, S. Vural#, B. Riemer+, P. Castoldi•, L. Valcarenghi•
•Scuola Superiore Sant'Anna, Pisa, Italy; Email:k.kondepu@sssup.it
#5G Innovation Centre, University of Surrey, Guildford, UK
+Software-based Networks (NGNI), Fraunhofer FOKUS, Germany

***Abstract*—In virtualised mobile networks schemes shall be devised to guarantee network resilience. For example, virtualised Evolved Packet Core (vEPC) resilience can be guaranteed by backup vEPCs placed in different network data centers. The demo implements a 5G vEPC resilience schemes based on vEPC hot backup. The demo will show the considered resilience scheme performance when hot backup vEPC is deployed close to or far from a working vEPC in federated testbeds.**

***Index Terms*—5G, virtual EPC, failure detection, resiliency, functional split.**

I. Introduction

The envisioned 5G network architecture, including the Next Generation Core (NGC, i.e., the new Evolved Packet Core — EPC — for 5G) and the Next Generation Radio Access Network (NG-RAN), will be heavily based on virtual network functions (VNFs) [1]. Network Function Virtualisation (NFV) enables an easy introduction of new network services by adding dynamic programmability to network devices (e.g., routers, switches, and applications servers) that, in turn, empowers fast, flexible, and dynamic deployment of new network and management services. Moreover, NFV also enables network slicing by providing multiple instances of the same network function. The exploitation of NFV is foreseen also in the NGC [2] and the New RAN technology [3].

In the technical specification (TS) 23.007 [4], 3GPP specified different failure detection and recovery mechanisms for EPC components. Moreover, scalable architectures for reliability management are being defined by ETSI NFV [5] and implemented in current open source orchestration frameworks such as Openstack [6]. For example, in [7] resilient schemes for recovering C-RAN failure are proposed based on the concept of access cloud network. In [8], a two-step resiliency scheme orchestrating lightpath transmission adaptation and eNB functional split reconfiguration is proposed. However, the performance of resilience schemes based on the aforementioned approaches once applied to 5G core network have not been fully evaluated so far.

The demo shows the capability of recovering vEPC failures by means of a vEPC in "hot backup". Both working vEPC and backup vEPC are deployed in multiple Network Function Virtual Infrastructure Points of Presence (NFVI-PoPs) made available by the federated testbeds belonging to the SoftFIRE project [9]. The demo is designed to evaluate the Service Recovery Time (SRT), that is the time required to regain user equipment (UE) connectivity, when the proposed resilient scheme is deployed in different NFVI-PoPs.

978-3-903176-15-7 © 2019 IFIP

II. The SoftFIRE Framework

The SoftFIRE Middleware Framework [10] provides an orchestrated federated virtualisation testbed consisting of component testbeds in multiple countries in Europe. The testbed provides virtualisation platforms controlled by OpenStack [11] and orchestrated by ETSI MANO [12] compliant Open Baton [13] orchestrator. SoftFIRE provides the testbed as an experimentation platform to third parties, which would like to test their 5G applications and virtualisation solutions in a real multi-site testbed. The platform has a Middleware solution based on TOSCA experiment definitions, parsed by an Experiment Manager component. This component invokes relevant sub-managers based on the virtualisation resources requested by an experimenter and provides requested resources (i.e. NFV monitoring, NFV deployment, security as a service, SDN, or reservation of physical devices such as UE or Long Term Evolution—LTE femto-cells). Open Baton then deploys the requested network service defined by the experimenter file, on requested component testbed(s).

III. Considered scenario and proposed scheme

The considered scenario and the proposed resilience scheme are depicted in Fig. 1 and Fig. 2 by referring to functional elements of the LTE-Advanced (LTE-A) architecture. The proposed resilience scheme considers a scenario where the vEPC fails (e.g., a virtual machine where the vEPC runs crashes). Fig. 1 shows the two considered vEPC resilience schemes based on vEPCs hot backup deployed in federated NFVI-PoPs. The one on the left features two co-located vEPCs (i.e., vEPCa and vEPCb deployed in Surrey 5GIC testbed) while the one on the right features a remote hot backup vEPC (i.e., vEPCr) deployed in a different compute resource available in another testbed (i.e., FOKUS). In the latter case two testbeds will be contemporarily utilized to implement the resilience scheme. In the Surrey 5GIC testbed two different VNFs (i.e., vOAISIM and vEPC) will be implemented by exploiting open source mobile platforms (i.e., OpenAirInterface—OAI). Here, vOAISIM VNF provides emulation of virtual user equipment (vUE) and evolved NodeB (eNB) while vEPC will be used to emulate the core network.

Fig. 2 shows the considered scenario and life-cycle event when vEPC VNF fails. Here, when VNFs are deployed,

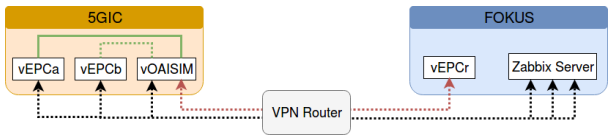

Fig. 1: RAN and Core network deployment in federated environment

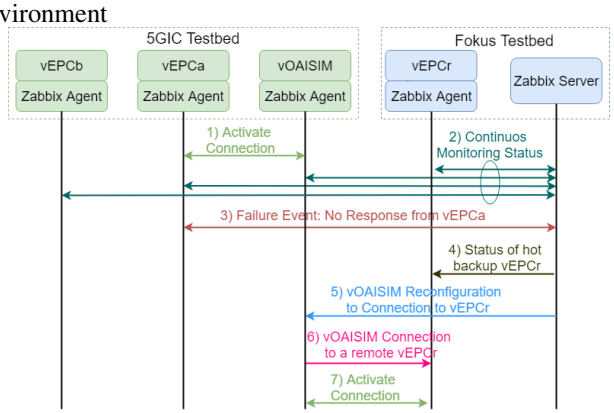

Fig. 2: Proposed scheme experimental evaluation setup

vOAISIM connects with vEPCa, and Zabbix server start monitors the VNFs that are associated corresponding Zabbix agent. Note that each vEPC VNF and vOAISIM VNF deployment contain also Zabbix agent. If the Zabbix server detects an anomaly activity in vEPCa (e.g., overload) or does not receive any status report from vEPCa (i.e., vEPCa crashed) for a predefined period of time (i.e., time to trigger the activity), the Zabbix server check the status of the hot backup vEPC to initiate a recovery procedure. The recovery procedure consist in reconfiguring vOAISIM to connect to the hot backup vEPCr. Upon reconfiguration vOAISIM is able to communicate hot backup vEPCr. Similarly, the experiment also demonstrate the recovery based on the local vEPCb deployed in 5GIC testbed.

IV. Initial Experimental Results

The initially considered performance evaluation parameter is the SRT, that is the time required to regain UE connectivity. SRT is measured as the time elapsing between the last ping reply sent by the vEPCa to the vUE before a hot backup remote connection initiation and the detection of the first successive ping reply after successful vUE reconnection with vEPCr (as shown in Fig. 3). Ping messages from the vUE to the EPCa are sent every 1ms.

Failure detection is implemented by configuring an action in the Zabbix monitoring server to detect an anomaly activity in vEPCa by monitoring output traffic from gtp0 interface (e.g., interface is not responsive) for a configurable period of time (i.e., 10ms). Once the anomaly is detected, the Zabbix server starts the recovery procedure (as shown in Fig. 2) to connect with remote or local hot backup vEPC. Here, the remote hot backup (vEPCr) connection is considered when the active vEPCa failure is detected.

Fig. 3 reports the initial vUE is attached to vEPCa both located in Surrey 5GIC testbed, and a Wireshark capture is performed at the vUE interface (172.16.0.2) of the ping messages exchanged by vUE and vEPCa (GTP interface address 172.16.0.1). When the vEPCa fails (e.g., interface is not reachable) the proposed resilience scheme is activated to connect vEPCr deployed in FOKUS testbed. The timestamp of the Wireshark is measured in seconds. As shown in the Fig. 3, once the recovery process is undergoing to connect from vEPCa to vEPCr, only ICMP request messages at the vUE interface are observed. The measured SRT is around 10s.

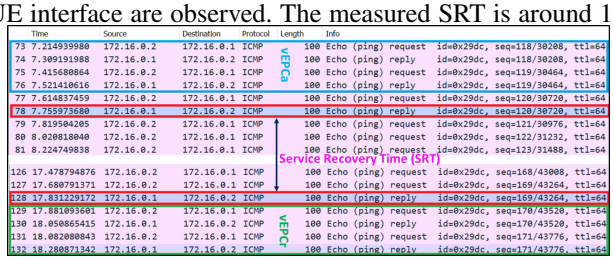

Fig. 3: Capture (at the vOAISIM VNF) of ICMP messages exchanged between the vEPCa and vEPCr.

During the demo the deployment of working and backup vEPC VNFs, and the virtualised RAN VNF on federated environment will be shown. The Zabbix tool will be used to show failure triggering and service recovery. The SoftFIRE testbeds will be accessed from a PC connected with Internet. Moreover, a pre-recorded video will be used as support material.

V. Conclusions

The proposed demonstration showed the performance of resilience schemes for virtualised mobile network functions. In particular, the demo focused on evaluating the time required to regain UE connectivity when a hot backup vEPC is deployed close to or away from a working vEPC. The demo exploited the remote access to federated testbeds.

Acknowledgment

This work has been partially funded by the EC H2020 “SoftFIRE” Project (Grant Agreement number 687860) and by the EC H2020 “5G-TRANSFORMER” project (Project ID 761536).

References

[1] 5G PPP, “View on 5G Architecture,” White Paper, https://5g-ppp.eu/white-papers (accessed 10/07/2017).
[2] V. G. Nguyen et al., “SDN/NFV-Based Mobile Packet Core Network Architectures: A Survey,” in IEEE Communications Surveys & Tutorials, vol. 19, no. 3, pp. 1567-1602, third quarter 2017.
[3] 3GPP TR 38.801, “Study on new radio access technology: Radio access architecture and interfaces (Release 14)”, V14.0.0 (2017-03).
[4] 3GPP TS 23.007, “Technical Specification Group Core Network and Terminals; Restoration procedures”, Dec 2017.
[5] ETSI GS NFV-REL 002, “Network Functions Virtualisation (NFV); Reliability; Report on Scalable Architectures for Reliability Management”, V1.1.1 (2015-09).
[6] F. F. Moghaddam, et al., “Self-Healing Redundancy for OpenStack Applications through Fault-Tolerant Multi-Agent Task Scheduling,” IEEE CloudCom 2016
[7] C. Colman-Meixner et al., “Resilient cloud network mapping with virtualized BBU placement for cloud-RAN”, in Proc. of ANTS (2016).
[8] K. Kondepu et al., “Orchestrating Lightpath Recovery and Flexible Functional Split to Preserve Virtualized RAN Connectivity”, J. Opt. Commun. Netw. 10, 843-851 (2018).
[9] EU SoftFIRE project, https://www.softfire.eu/
[10] SoftFIRE Middleware, http://docs.softfire.eu/softfire-middleware/
[11] OpenStack, https://www.openstack.org/
[12] ETSI GS NFV-MAN 001, “Network Functions Virtualisation (NFV); Management and Orchestration”, V1.1.1 (2014-12)
[13] Open Baton, https://openbaton.github.io/