EUROPEAN COOPERATION IN SCIENCE & TECHNOLOGY

SHORT TERM SCIENTIFIC MISSION (STSM) SCIENTIFIC REPORT

This report is submitted for approval by the STSM applicant to the STSM coordinator

Action number: CA 15127
STSM title: Measurements of 5G Networks in disaster based situations
STSM start and end date: 14/05/2017 to 27/05/2017
Grantee name: Dr. Eryk Schiller

PURPOSE OF THE STSM:

During the previous STSM [Sch17], the University of Bern[1] (UBERN) and EURECOM[2] classified the result of various critical situations affecting the 4G mobile network. In normal circumstances, the evolved Node B (eNB) cannot operate in the disconnected core scenario, and the radio network goes down. We demonstrated, however, that the network of disconnected operational Mobile Edge Computing (MEC)-enabled eNBs can remain (partially) functional using the Information Centric Networking (ICN)/Delay Tolerant Networking (DTN) paradigm.

In the previous STSM, we studied Network Function Virtualization (NFV)-based orchestration, i.e., provisioning times for necessary services, i.e., Base Band Units (BBUs), Mobility Management Entities (MMEs), Serving Packet Gateways (SPGWs), Home Subscription Services (HSSs) providing a functional an on-demand Evolved Universal Terrestrial Radio Access Network (E-UTRAN) on the MEC infrastructure. The STSM resulted in one book chapter published [Sch17], and a joint project proposal submitted to French ANR[3] and Swiss SNF[4] in the area of mobile networks. We studied the orchestration framework of MEC services (BBU, MME, SPGW, HSS, DTN) [Sch17], however, we did not perform any measurements of actual radio transmissions. In this STSM, we fill this gap by concentrating on performance of the Software Defined Networking (SDN)-based communication between the UE and DTN service running on the MEC infrastructure.

[Sch17] E. Schiller, E. Kalogeiton, T. Braun, A. Gomes, N. Nikaein. ICN/DTN for Public Safety in Mobile Networks. In: D. Camara, N. Nikaein (eds.) Wireless Public Safety Networks 3. Applications and Uses. London: ISTE Press - Elsevier, 2017.

DESCRIPTION OF WORK CARRIED OUT DURING THE STSMS

In previous projects carried out between EURECOM and UBERN, i.e., FP7 MCN [MCN16] and FP7 FLEX [FD541, FD542], we have developed an NFV/SDN-based MEC infrastructure allowing us to redirect traffic from UEs to the MEC applications running on the MEC Cloud infrastructure. As an example, in Figure 1, we present a MEC infrastructure running a web cache service providing web content to UEs directly from the network edge. Hence, the end-users could experience a good quality service with high throughput and low delay.

1 http://www.cds.unibe.ch
2 http://www.eurecom.fr
3 http://www.agence-nationale-recherche.fr
4 http://www.snf.ch

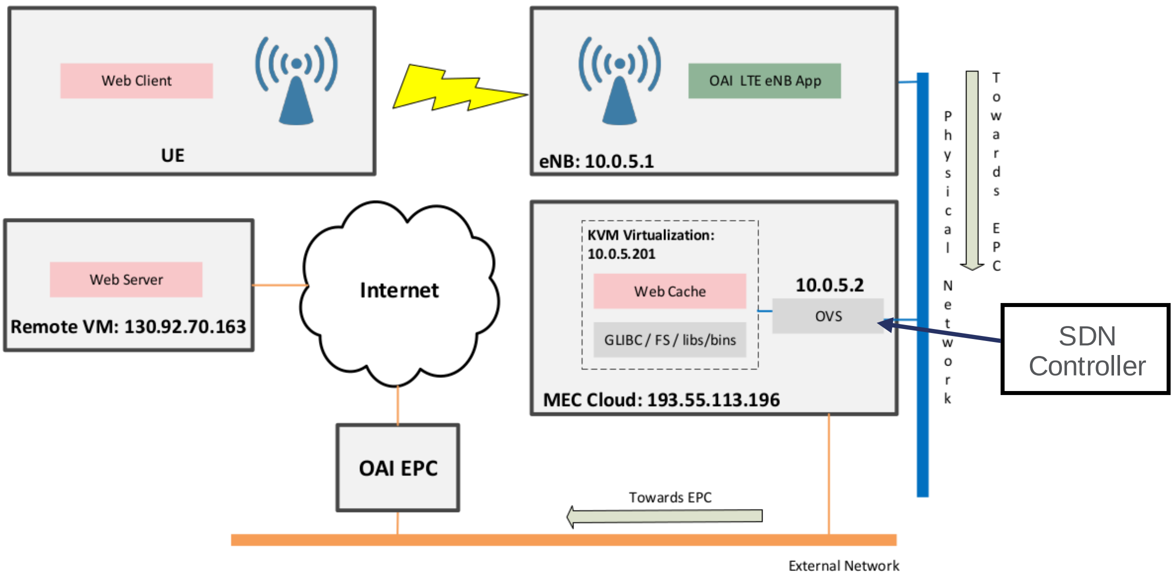

Figure 1: An example web caching squid-based application running on the MEC infrastructure.

In Listing 1, we present an example set of OVS rules that redirect all traffic belonging to a given UE towards a web-cache. To install the rules, we use the *ovs-ofctl* command line controller provided with the OVS distribution. Please notice that the OVS switch has to implement tunneling and matching for the GPRS Tunnelling Protocol (GTP-U) to allow for this functionality. The matching code can be downloaded from our respository https://github.com/ejschiller/FLEX/tree/master/ovs, while the GTP-U tunnelling patch can be downloaded from https://patchwork.ozlabs.org/patch/579431/

```
# MATCHING THE PACKET OF A GIVEN UE (TEID BASED) AND REDIRECTING IT TO THE
# LOCAL GTP TUNNEL END-POINT

ovs-ofctl add-flow OVS-SWITCH
"in_port=$ENB_PORT,ip,udp,tp_dst=2152,gtp_teid=$ENB_TEID,nw_dst=$SPGW_IP,ac
tion=mod_nw_dst=$LOCAL_IP,NORMAL"

# DECAPSULING THE GTP HEADER
# PROVIDING A UE PACKET TO THE CACHE

ovs-ofctl add-flow OVS-SWITCH
"tun_src=$ENB_IP,tun_id=$ENB_TEID,action=mod_dl_dst:$SERVICE_MAC,output:$SE
RVICE_PORT"

# ENCAPSULATING THE PACKET FROM CACHE WITH APPROPRIATE UE SPGW TEID
# SENDING THE ENCPASULATED PACKET TO THE ENB

ovs-ofctl add-flow OVS-SWITCH
"in_port=$SERVICE_PORT,dl_type=0x0800,vlan_tci=0,ip,nw_dst=$UE_IP,nw_src=$S
ERVICE_IP,action=set_tunnel:$SPGW_TEID,set_field:$ENB_IP>tun_dst,output:$GT
P_TUN_PORT"
```

Listing 1 An example action on the controller redirecting traffic from the UE towards MEC Applications.

where:
OVS-SWITCH is the OVS instance deployed on the MEC Cloud,
$ENB_PORT – the OpenFlow Port connecting the switch with the eNB (number),
$SERVICE_PORT – the OpenFlow Port connecting the switch with the (caching) service (number),
$ENB_TEID – the GTP tunnel endpoint identifier of the user on the eNB side,
$SPGW_TEID – the GTP tunnel endpoint identifier of the user on the SPGW side,
$ENB_IP – the IP address of the eNB,
$LOCAL_IP – the local address of the switch on the MEC Cloud,
$SPGW_IP – the IP address of the Serving Packet Gateway (SPGW),
$SERVICE_MAC – the MAC address of the service (i.e., cache),

$SERVICE_IP – the IP address of the service (i.e., cache),
$UE_IP – the IP address of the UE,
$GTP_TUN_PORT – the OpenFlow Port number of the local GTP tunneling end-point.

Most of the variables are deployment dependant (i.e., $ENB_PORT, $CACHE_PORT, $ENB_ADDR, $LOCAL_ADDR, $SPGW_ADDR, $CACHE_MAC, $CACHE_IP, $GTP_TUN_PORT), however, some of them are dynamic and established upon the UE attachment by the Mobility Management Entity (MME) (i.e., $ENB_TEID, $SPGW_TEID, $UE_IP).

During the STSM, we modified the caching environment, and developed the following Public Safety (PS) Solution (c.f., Figure 2).

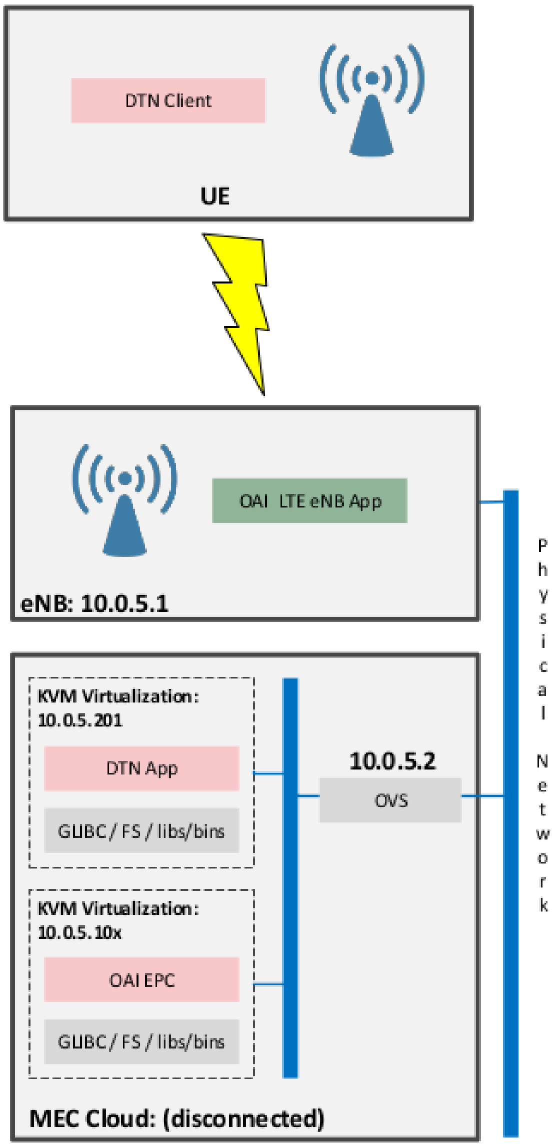

Figure 2: Public Safety Solution with SDN/NFV.

The main core of the provider is now disconnected. We therefore instantiate a complete small core on the MEC Cloud (i.e., MME, SPGW, HSS) as Virtual Network Functions (VNFs) through the JUJU[5] Virtual Network Function Manager (VNFM). The eNB connects to the newly instantiated core. We use OpenAirInterface [OAI15] from the OAI development branch as the EPC and eNB. The eNB is configured to use 5 MHz channels in Band7 of the LTE spectrum. We also instantiate a DTN PS service, and the OVS-based traffic redirect on the MEC cloud using our SDN-based solution previously used in caching in FP7 FLEX Project. The UE is also equipped with a DTN PS client. Hence, the UE and DTN PS can directly communicate when the UE is successfully attached to the eNB.

5 https://insights.ubuntu.com/2015/07/01/juju-for-telcos-and-service-providers-pt-1-2/

[MCN16] B. Sousa *et al*., "Toward a Fully Cloudified Mobile Network Infrastructure," in *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 547-563, Sept. 2016.
[FD541] E. Schiller, A. Gomes, T. Braun, N. Nikaein, D5.41 – MEC: Architecture of MEC Cashing. Deliverable of the FP7 FIRE LTE testbeds for open Experimentation (FLEX). Grant Agreement 612050, Aug. 2016.
[FD542] E. Schiller, D5.42 – MEC Caching Prototype. Deliverable of the FP7 FIRE LTE testbeds for open Experimentation (FLEX). Grant Agreement 612050, Jan. 2017.
[Pfa15] B. Pfaff, J. Pettit, T. Koponen, E. J. Jackson, A. Zhou, J. Rajahalme, K. Amidon, The Design and Implementation of Open vSwitch. In NSDI (pp. 117-130), 2015.
[OAI15] N. Nikaein, R. Knopp, L. Gauthier, E. Schiller, T. Braun, D. Pichon, C. Bonnet, F. Kaltenberger, and D. Nussbaum. 2015. Demo: Closer to Cloud-RAN: RAN as a Service. In Proceedings of the 21st Annual International Conference on Mobile Computing and Networking (MobiCom'15). ACM, New York, NY, USA, 193-195.

DESCRIPTION OF THE MAIN RESULTS OBTAINED

We verified that our PS solution of the SDN/NFV architecture works. We were able to successfully dtnping (use the dtn ping function) between the UE of (dynamically assigned) IP address *172.16.0.2*, eNB TEID *0x0000001*, SPGW TEID *0xca6fe0dd*, and DTN address *ue.dtn* towards the DTN service on the MEC cloud of address *10.0.5.203* and DTN address *enb.dtn*. The dtnping values that reflect the discovery time in a small DTN network of ue.dtn and enb.dtn are provided in Listing 2.

```
20 bytes from [dtn://enb.dtn/ping]: 'dtnping!' seqno=0, time=233 ms
20 bytes from [dtn://enb.dtn/ping]: 'dtnping!' seqno=1, time=202 ms
20 bytes from [dtn://enb.dtn/ping]: 'dtnping!' seqno=2, time=233 ms
20 bytes from [dtn://enb.dtn/ping]: 'dtnping!' seqno=3, time=203 ms
20 bytes from [dtn://enb.dtn/ping]: 'dtnping!' seqno=4, time=215 ms
20 bytes from [dtn://enb.dtn/ping]: 'dtnping!' seqno=5, time=210 ms
20 bytes from [dtn://enb.dtn/ping]: 'dtnping!' seqno=6, time=226 ms
20 bytes from [dtn://enb.dtn/ping]: 'dtnping!' seqno=7, time=231 ms
20 bytes from [dtn://enb.dtn/ping]: 'dtnping!' seqno=8, time=243 ms
20 bytes from [dtn://enb.dtn/ping]: 'dtnping!' seqno=9, time=369 ms
20 bytes from [dtn://enb.dtn/ping]: 'dtnping!' seqno=10, time=209 ms
20 bytes from [dtn://enb.dtn/ping]: 'dtnping!' seqno=11, time=186 ms
20 bytes from [dtn://enb.dtn/ping]: 'dtnping!' seqno=12, time=240 ms
20 bytes from [dtn://enb.dtn/ping]: 'dtnping!' seqno=13, time=236 ms
20 bytes from [dtn://enb.dtn/ping]: 'dtnping!' seqno=14, time=240 ms
20 bytes from [dtn://enb.dtn/ping]: 'dtnping!' seqno=15, time=220 ms
20 bytes from [dtn://enb.dtn/ping]: 'dtnping!' seqno=16, time=198 ms
20 bytes from [dtn://enb.dtn/ping]: 'dtnping!' seqno=17, time=223 ms
20 bytes from [dtn://enb.dtn/ping]: 'dtnping!' seqno=18, time=202 ms
```

Listing 2 The outcome of a DTN ping between the PS MEC Application and the UE.

It takes around 237 ± 38 ms to discover the connected *enb.dtn* from the UE. When the connectivity was successfully tested, we started sending files between the *ue.dtn* and *enb.dtn*. It is worth noting that the dtnping times are much larger than the classical ICMP ping times, as the DTN service has to first discover the destination (if connected). The example transmission of six 1 MB files (the time period between consecutive transmissions is 40 s) is provided in Figure 3. The throughout of 1.06 ± 0.08 MB/s was established in a single transmission (c.f., Figure 4).

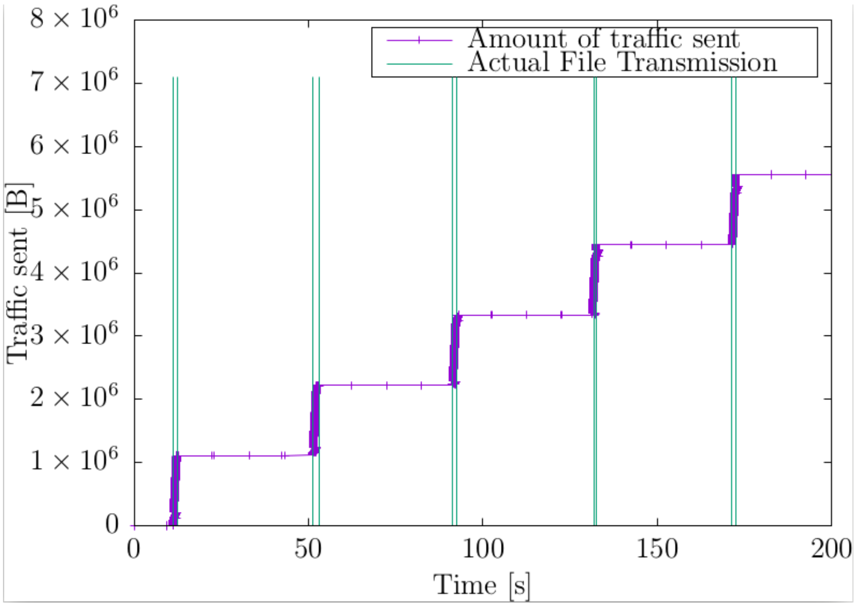

Figure 3: 6 consecutive transmissions of 1 MB files using the DTN service.

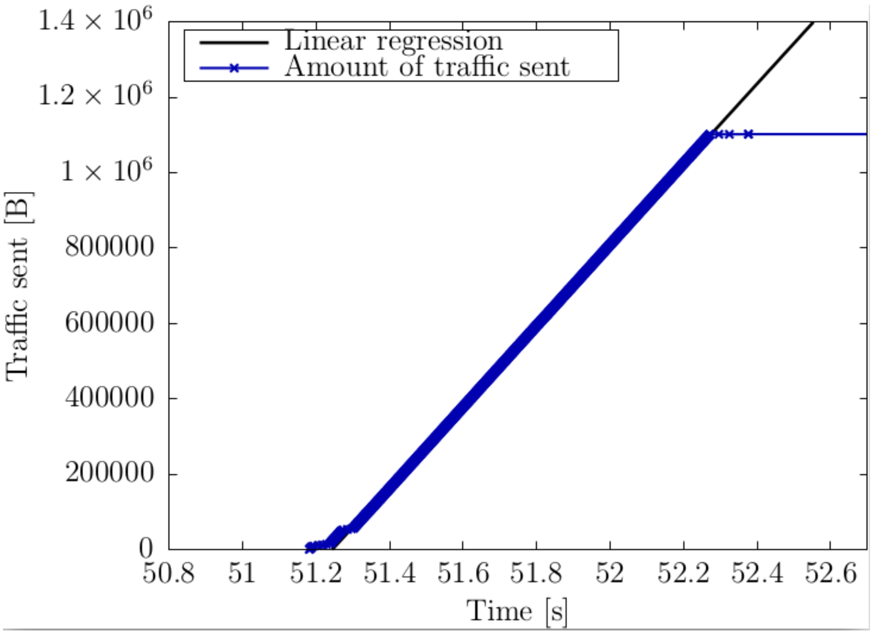

Figure 4: Zoom at an arbitrary transmission

Concluding, we have successfully established a small core on the MEC infrastructure that allowed us to exchange data between clients using the DTN service. The service might be deployed on MEC infrastructures to allow for a data exchange in disaster situations when the network core is unavailable. The DTN throughput between connected entities was established at around 8.64 Mbit/s using OAI in the development branch with 5 MHz channels and the Single Input Single Output (SISO) mode.

We used the following JUJU charms to instantiate services with JUJU VNFM:
eNB: https://jujucharms.com/u/navid-nikaein/oai-enb
MME: https://jujucharms.com/u/navid-nikaein/oai-mme
SPGW: https://jujucharms.com/u/navid-nikaein/oai-spgw
HSS: https://jujucharms.com/u/navid-nikaein/oai-hss
DTN: https://jujucharms.com/u/navid-nikaein/dtn2/

Hardware characteristics of devices:
eNB: Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz quad-core, 32 GB RAM, 1 GB HDD, USRP B210.
MEC Cloud: Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz quad-core, 16 GB RAM, 200 GB SDD.

FUTURE COLLABORATIONS (if applicable)

Currently, we are waiting for the results of the joint SNF-ANR call submitted in Oct 16 (1st Stage), and Apr 2017 (2nd Stage). The results are expected in early July 2017. In the case our proposal is successful, UBERN, INRIA[6], and EURECOM will run a research project in the domain of network slicing for mobile edge systems.

6 https://www.inria.fr/centre/sophia