Network Slicing for Mission Critical Communications

Davide Borsatti*, Chiara Grasselli◇, Luca Spinacci†, Marina Settembre†, Walter Cerroni*◇ and Franco Callegati#◇
*Department of Electrical, Electronic and Information Engineering - DEI - University of Bologna
◇ Centre for Industrial ICT Research - University of Bologna
† Leonardo S.p.A.
#Department of Computer Science and Engineering - DISI - University of Bologna.

***Abstract*—Mission Critical (MC) communications are key to effective Public Protection and Risk Reduction (PPRR) actions. The 3GPP standards include the definition of MC applications and services in an architectural framework compatible with current (LTE) and future (5G) mobile networks. In this paper we report an experimental activity where MC communication services are implemented in a fully virtualized environment, being deployed and tested in a multi-domain network slicing architecture compliant with the ETSI NFV MANO specifications. The level of automation in service deployment and the slice isolation features are demonstrated, in line with the 5G approach of separation between control and data plane, showing the benefits in terms of application performance and management flexibility.**

I. Introduction

The support to communication during missions related to public safety, in case of disaster or other special events, is critical to the success of the mission itself. For this reason Mission Critical (MC) communications and services were declared a key priority by the 3rd Generation Partnership Project (3GPP) [1], that placed significant effort in designing the standards for a platform capable to support such services over state-of-the-art mobile networks such as LTE and 5G.

A key concept behind the 5G definition is the capability to serve in an effective and efficient way vertical applications, among which MC communications. This should be enabled by the network slicing concept [2], [3]. One single infrastructure, mostly based on virtualized network functions (VNFs) hosted in data centers, supports different network architectures and set-ups, with different characteristics and devoted to different vertical applications. To date, four slice types have been standardised: enhanced mobile broadband (eMBB), massive IoT (MIoT), ultra reliable low latency communications (URLLC) and Vehicle to everything (V2X) [4].

The various network slices offer to the network users differentiated characteristics and, most of all, full isolation (users of a slice are completely unaware of other slices in the same portion of infrastructure) also at the performance level (two slices with different QoS characteristics should not influence each other in terms of performance).

In this manuscript we report the results of an experiment of network slicing for MC communications. A Network Slice (NS) is spread over several data centers, interconnected by a wide area network (WAN), and hosts network functions that implement the MC applications. In particular, in Section II we describe the network architecture and its main components, which also serves as a primer on the theoretical background related to NFV-MANO and MC communications. In Section III we describe how we devised the MC network slice. In Section IV we report the results obtained from the experimental test-bed. Finally, Section V draws the conclusion of the paper.

II. Network architecture and system components

The implementation scenario considered in this work is in line with the current trends and exploits:

- Cloud computing, allowing virtualization of computing resources in data centers equipped with general purpose hardware;
- Network Function Virtualization (NFV), that fosters flexible and cost-effective service orchestration through the deployment of virtualized network functions;
- Software Defined Networking (SDN), that decouples software-based network control and management planes from the hardware-based forwarding plane, turning traditional vendor locked-in infrastructures into communication platforms that are fully programmable.

The general network architecture considered here is depicted in Fig. 1. To resemble a general networking scenario both mobile and fixed access are considered. The network building blocks are all implemented as VNFs located in two data centers interconnected by a transport network, the Edge Data Center (E-DC) and the Core Data Center (C-DC). The E-DC emulates the access network, with its processing capabilities, close to the fronthaul of the mobile network, therefore more suitable to support functionalities with stringent latency requirements. The C-DC is in the backhaul of the mobile network and will be devoted to more data-intensive applications with less critical latency constraints.

The MC communication network is deployed as a NS that includes all the logical components of the mobile network, as well as the server for the MC communication support, compliant with the 3GPP specifications [1].

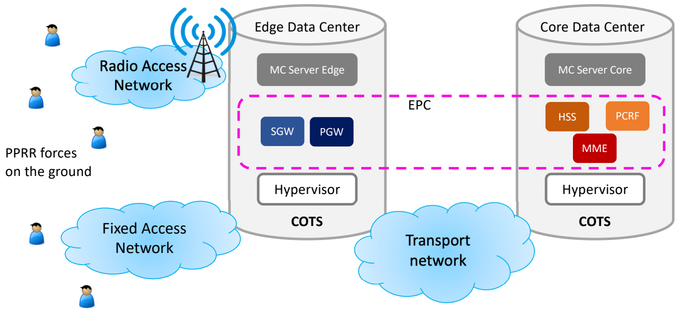


Fig. 1. General network slicing architecture for MC communications.

A. NFV-MANO and OSM

The ETSI NFV Management and Orchestration (MANO) standards focus on the scenario where VNFs are deployed over a set of data centers that may be either closely or remotely located. The data centers hosting the VNFs are managed by the infrastructure management system chosen by the owner/provider, while general networking services are managed by SDN controllers. MANO addresses these components as Virtual Infrastructure Managers (VIMs).

On top of the VIM the MANO architecture places the VNF Manager (VNFM) and NFV Orchestrator (NFVO). The VNFM is responsible for the lifecycle management of the VNFs, while the NFVO orchestrates the set of resources provided by the underlying infrastructures and through specific interfaces communicates with the VIMs and the VNFMs to manage the VNFs lifecycle. This includes guaranteeing proper connectivity, traffic steering, configuration of functionalities etc. In both cases the VNFM and NFVO actions will be implemented by talking with the VIMs.

VIMs may differ one another because they serve different purposes and manage different technologies, or because the owners of distinct infrastructures choose different tools to manage them. MANO takes this into account and just requires that the VIM may be contacted in some standard and abstracted way. When the infrastructure to be managed is a Wide Area Network (WAN), i.e. a transport network interconnecting different domains, the manager of the infrastructure is called WIM (WAN Infrastructure Manager). Generally speaking, the VIM and the WIM de-couple the service abstractions from the underlying technology-specific resources.

The ETSI-MANO NFV approach allows the full exploitation of the isolation capabilities of data centers. In our experiment the NFV-MANO orchestration was implemented using Open Source MANO (OSM) [5], the open-source software platform promoted by ETSI itself. OSM is now getting in full maturity as an orchestration platform and implements both the VNFM and the NFVO components. OpenStack is natively supported as a VIM allowing a rather seamless integration with the cloud platform. More details will be given in the following about its usage in our experiment.

B. The MC server

Leonardo MCX (Mission Critical Services) is part of the Leonardo CSP (Communications Service Platform) product family [6]. It extends the portfolio of standard solutions for Public Protection and Risk Reduction (PPRR) communications, ranging from Digital Mobile Radio (DMR) to Terrestrial Trunked Radio (TETRA) technologies, with next generation broadband capabilities. It is a complete Mission Critical solution compliant with 3GPP standard MCX. Includes features from MC Push-to-Talk (PTT), MC Video and MC Data, providing PPRR users with the next generation platform for critical communications over 4G/5G networks. The full solution for MCX is made of the following components:

- an Android Client designed for on-field operations, with a complete set of functionality, that can be installed in off-the-shelf smartphones as well as on ad-hoc terminals;
- a Web based dispatcher, providing control, monitoring and management of the operations of the teams;
- a Management interface for the management and monitoring of the platform KPIs;
- a Session Initiation Protocol (SIP) Core for IP Multimedia Subsystem (IMS)-less scenarios and that can inter-operate with external IMS.

The MCX server can be deployed in a distributed fashion, with a share of roles. In particular the media servers, i.e. the SIP servers that will manage and deliver the media streams, can be de-coupled from the registration server used for signalling. This is the feature that was exploited in our experiment, with the goal to keep the media servers as close as possible to the final users and guarantee optimal performance.

C. The mobile access network

The mobile access network is fully virtualized exploiting well known open-source software components. The focus is on the Evolved Packet Core (EPC), assuming that the Radio Access Network (RAN) will be deployed already on the ground either with dedicated resources or by sharing the resources of the public mobile network.

In our experiment the RAN was simulated. Both user equipment (UE) and eNodeB were simulated with the L2 network Functional Application Platform Interface (nFAPI) Simulator provisioned by OpenAirInterface [8]. This simulator does not require any specific hardware and simulates L2 and above stack layers, short-cutting the physical layer. Furthermore, it gives the possibility to simulate multiple UEs with a single instance. The EPC was implemented with the NextEPC platform [9]. NextEPC implements a full functional LTE EPC in a similar way to other platforms, such as for instance OpenAirInterface. We opted for NextEPC because of its flexible modular architecture that has been designed already to be deployed in a virtualized environment. The NextEPC software suite is composed of 5 modules (*nextepc-mmed*, *nextepc-sgwd*, *nextepc-pgwd*, *nextepc-hssd* and *nextepc-pcrfd*) that can be individually installed as packages in several Linux distributions and can be managed as daemons with the respective native system and service managers. Each module provides one or more dedicated configuration files that must be modified according to the actual set-up of the data plane and control plane interfaces compliant with the 3GPP standards. In addition, this software suite gives the possibility to install a Web User Interface that allows to add in the Home Subscriber Server (HSS) database the information related to users and service subscriptions, and ease their further management.

Although NextEPC does not yet provide the Control and User Plane Separation (CUPS), its modular architecture allows a deployment of the various components in different data centers. We exploited this feature to implement ad-hoc a partial CUPS, as will be described in the remainder of this manuscript.

D. *Data center management infrastructure deployment*

In our experiment, OpenStack was used as a cloud management platform [5]. OpenStack is the leading open-source software tool for this task and is now in its full maturity. OpenStack represents the VIM used for the resource management in each data center. The OpenStack installation was carried out via Kolla-Ansible, which allows to quickly get a production-ready container-based OpenStack environment. Each component (i.e., Nova, Neutron, Cinder, etc.) is deployed inside a separate Docker container, thus granting a separate working environment for each one of them. The installation strictly follows the official guide and only the most common components were used (only the traditional software update and upgrade was carried out on bare metal machines prior to the installation process).

III. A Network Slice for MC communications

A. *Actors and Roles*

Network slicing is a process that involves three main actors:

- Infrastructure Provider (IP): the owner of the infrastructure providing all the infrastructural management actions, in the specific example a network provider acting at a local or national scale operating a private network to support the MCX services;
- Network Slice Provider (NSP): the provider of the communication service implemented with the network slice, in this specific case the governmental agencies that provide the MCX support and/or third parties under contract to provide this kind of service;
- Network Slice Customer (NSC): the user of the communication service, in this specific case the PPRR forces that will use the MC network during operations (police, firefighters, hospital ER, etc.).

These actors must have rights according to their respective roles, with IP and NSP having specific management roles to keep the infrastructure up and running. Therefore the slice architecture must be defined in such a way that allows a seamless co-existence of these actors and provides all of them with the required functionalities.

An important characteristic of the NS under investigation is that it is not bound to a single data center, but is basically split into 4 logical sections:

1) Mobile and fixed access network;
2) Edge Data Center (E-DC) virtualizing the access part of the EPC and the edge MCX server;
3) Core Data Center (C-DC) virtualizing the core part of the EPC and the core MCX server;
4) Interconnection network between the DCs, that could be either a public network or a private geographical interconnection.

Moreover, the NS must be designed to satisfy the following main characteristics:

- interconnection with the outside of the DC using two logical networks, the former dedicated to inter-DC connectivity, the latter used to connect to the outer world;
- capability to establish tunnels and/or specific routing policies on the external networks;
- VNFs must be manageable objects as required by the NFV-MANO architecture;
- VNFs must be protected, meaning that their interfaces must not be directly exposed on the interconnection network to the outside of the data center;
- separate management must be guaranteed for the IP and for the NSP;
- specific management console for the NSP must be reachable from outside the data centers, to keep NS management fully transparent to the IP.

In the following we will explain the principles and the instruments that we used for the slice design and deployment.

B. *Network Slice Architecture and Characteristics*

According to [3] the NS specifications are described with the NEtwork Slice Type (NEST), a set of parameters with associated values that are defined using a generalized dictionary (Generic Slice Template or GST) but referring to a specific service or set of services. A possible NEST for the network slice here considered is presented in Table I.

The GST acts as a template for the NEST and the NEST provides QoS and/or functional specifications for the NS. None of them says how the NS should be implemented. The specific implementation of the NS is usually called the *Blueprint*, i.e. the collection of all the technical details that are necessary to implement that particular NS. As we will see in the following, the NS implemented in this work is rather complex and its deployment was split into several steps, to make configuration and debugging easier and more controllable.[1]

Every section of the slice is specified by means of several NFV-MANO descriptors, including one that describes how to put together the various components. Some of these descriptors are common to the various slice sections and can be re-used.

[1] It is also worth underlining that the slice architectures presented here are a graphical sketch. The actual implementation in OpenStack is even more complex since many VNFs are made of two VMs, one for production and one for management, with an additional network in between to connect them.

TABLE I
EXAMPLE OF SOME NEST PARAMETERS FOR THE MC COMMUNICATIONS NETWORK SLICE

ATTRIBUTE	**VALUE**
Coverage	Local (Outdoor)
Guaranteed Downlink Throughput per Network Slice	391600 (391.6Mbps, band 3, channel 20MHz(100RB), 256QAM, 4x4MIMO)
Mission Critical Support	1. mission critical
+ Mission-Critical Capability Support	1: Inter-user prioritization, 2: Pre-emption, 3: Local control
+ Mission-Critical Service Support	1: MCPTT, 2: MCData, 3: MCVideo

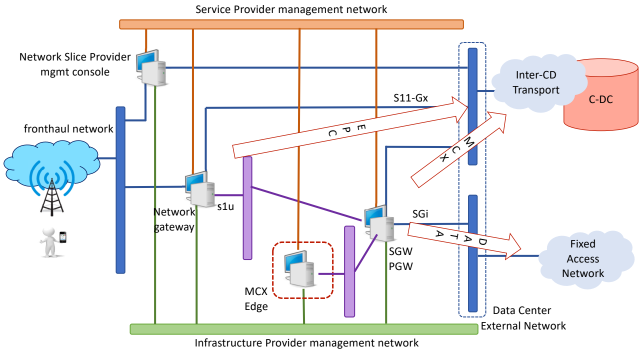


Fig. 2. Architecture of the access section of the slice (E-DC), with SGW and PGW still shared according to the LTE architecture.

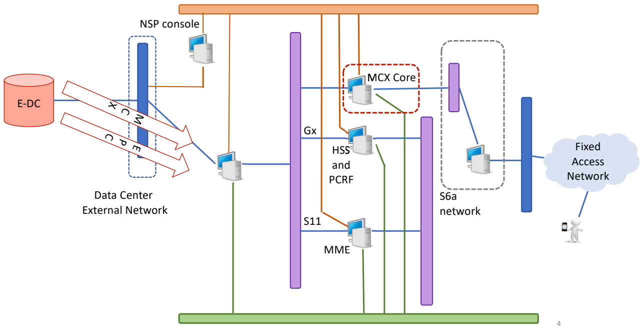


Fig. 3. Architecture of the core section (C-DC) of the slice with control plane components.

The full set of these descriptions and related configuration files represents the NS Blueprint.

Figures 2 and 3 show the deployment architectures for the two sections of the slice to be hosted in E-DC and C-DC. The section in the E-DC (Fig. 2) will host the two gateways of the EPC, namely the Serving Gateway (SGW) and the Packet Data Network Gateway (PGW), together with some other components we believe are needed to guarantee full slice functionalities. These additional components are:

- Network Slice Provider management console, connected to the various slice components for management purposes;
- Network gateway, providing the network functionalities required for correct traffic routing between the slice components and the external networks, thus also providing the required traffic isolation for security purposes.

In this scheme the SGW and the PGW will carry data traffic to the Internet, basically being devoted to the user data plane. Control plane traffic will be routed directly to the control plane components in the C-DC by the network gateway. In this way we achieve a basic CUPS, that can be extended gradually to a fully fledged 5G compliant architecture. In this schematic we assume the eNodeB is also hosted directly in the E-DC and connected to the gateway via an external network of the DC. This is not mandatory in general: in case one or more eNodeBs are implemented with dedicated hardware outside the data center, the interconnection will be exactly the same and therefore the slice blueprint would not need any variation. Together with the components of the mobile network the E-DC will also host the edge MCX server, that will be responsible mostly for the data traffic among end users.

The section in C-DC (Fig. 3) will host the control plane components, i.e. Mobility Management Entity (MME), HSS and Policy and Charging Rules Function (PCRF) as well as the MCX core server. The slice section also includes a NSP management console and network gateway. The control plane traffic will be routed to the slice components by the network gateway via an internal network, according to the proper addressing configured at the eNodeB. Similarly, the interconnection between the MME and the SGW will be guaranteed.

A general comment is related to the external network interconnecting the two DCs. In our architecture it is split in two logical sections, the former devoted to DC interconnection, the latter devoted to WAN connectivity. However, this splitting has the aim to show that these could be two different infrastructures, as well as just a single infrastructure with two logical roles, maybe mapped on different IP networks.

C. Network Slice Delivery and Lifecycle management

In [11] the various steps implementing a full NS lifecycle management are defined and described as in Fig. 4. All these steps have been implemented in the test-bed described in this work. The preparation phase includes the NS description and the environment preparation.

The NS description consists in creating the OSM descriptors that, according to ETSI MANO approach, provide all information regarding:

1) the VNF packages to be run in the slice;
2) the interconnections between them (Virtual Links in NFV-MANO terminology), described in the Network Service Descriptors (NSD) and Virtual Link Descriptors (VLD);

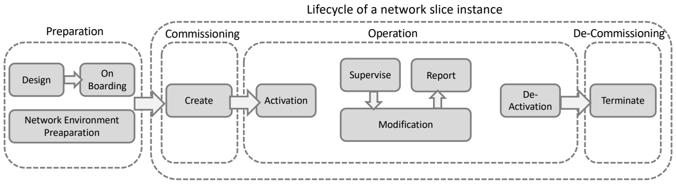


Fig. 4. The schematic representation of the various steps of the NS life-cycle management, as described in [11].

3) the Network Slice Template (NST) as a combination of Network Service Descriptors (NSDs);
4) the details of the VIMs where the NS has to be instantiated;
5) the VNF Forwarding Graph Descriptor (VNFFGD), specifying the traffic path from one VNF to another, which has to be implemented in the NS.

The preparation phase includes the setup of that part of the infrastructure which is not NS specific. In this particular case it refers to the networks in the cloud platform that must be shared between slices and must exist before the NS is started. Three such networks were set up by the administrator of OpenStack (acting as IP):

- the management network of the IP, that will be connected to the parts of the NS that the IP has to control in case of some emergency event, collaborating with or overriding the management actions from the NSP;
- the inter-DC interconnection network;
- the external networks that will be used to connect to the access networks, either mobile or fixed.

IV. Experimental Results

All the experiments were run in a private data center, with two separate OpenStack clusters for the E-DC and C-DC, respectively. Each one of them is composed by two physical servers, equipped as follows: 64 GB of RAM; 40 CPUs; 1.2 TB of disk; 1 Gbit/sec interfaces; Ubuntu 18 LTS as OS.

The overall scenario considered is shown in Fig. 5. From the SIP Uniform Resource Identifier (URI) point of view the domain is simply called `test` and two UEs were used registered as `user1@test` and `user2@test`. The paths of the signalling traffic flows are also shown in Fig. 5. Although the gateways (such as SGW and PGW) are not split and will carry both control and user plane traffic, according to the LTE architecture implemented in NextEPC, the Figure shows that the slice is ready for CUPS and enables splitting the various control plane components between the E-DC and the C-DC, leaving closer to the user the components that may help providing better performance.

Coming to the experiments, at first we tested the correct functional splitting of roles of the two MCX servers according to the planned split of workload. In the considered scenario the core MCX server is dedicated to handle signalling traffic, such as SIP registration and call set-up messages, while the edge MCX server acts as media server only. Figure 6 shows the flow of an MCVIDEO call from the point of view of the caller (`user1@test 10.250.123.101`) to the callee (`user2@test 10.250.123.102`). The call flow is produced with Wireshark out of the traffic traces. The core MCX is located at `10.250.2.249` while the edge MCX is located at `10.250.2.35`. The MCX servers are configured in order to force the communication to go through the media server coupled with the signalling server. The Figure shows that the split of roles is correctly realized in the slice. Indeed, the call forwarded by the MCX servers to the callee shows a clear separation of the signalling from data. The SIP traffic required to set-up and close the multimedia call between the two users is routed to the core MCX server. In fact, we see that SIP messages such as INVITE, TRYING, RINGING flow between the core MCX server (`10.250.2.249`) and the callee (`10.250.123.102`). Instead, the Real-time Transport Protocol (RTP) media traffic is exchanged between the edge MCX server and the users. In particular, with reference to the reported traffic trace, the RTP packets are forwarded from the edge MCX (`10.250.2.35`) to the callee.

Then to prove the effectiveness of the CUPS approach we exploited the performance measure feature of the MC mobile app. This is an Android app that can be installed in a commercial smartphone or in an Android emulator and provides all the MC service implementations as per the 3GPP standard, in particular MCDATA, MCVOICE and MCVIDEO as required by the NST. The performance feature of the app provides a series of evaluation tools for measuring network latency and capacity as shown in Fig. 7. To emulate a greater latency when connecting to the core infrastructure we forced a delay of $T = 200$ ms on the inter-DC connection. This delay was forced with Linux traffic control on the outgoing interface of the PGW. We asked the app to register on both the MCX core and on the MCX edge.

Obviously the MCX core is the only one which allows the registration of a SIP user since it is the only one running the management functions. When we ask the MC app to register on the MCX edge which is acting as media server only, the registration is not successful but still the app allows the execution of the performance test, even though in a limited way. As a consequence the two screenshots are different. For the scopes of this research the field of relevance that can be compared are: `2. CONNECT TCP` and `3. HTTP PING`.

These values depend on the round trip time (RTT) of the data connection. We can see that in both cases they are approximately 200ms larger in the connection to the MCX core than to the MCX edge. This is perfectly in line with the additional latency introduced in the path towards the C-DC, that is in this experiment 200ms.

Therefore we can conclude that, in case of a real call, the RTT of the media flows (voice and video) would be significantly lower when compared to the RTT of the signalling towards the MCX in the core. This is one of the advantages expected by the CUPS approach.

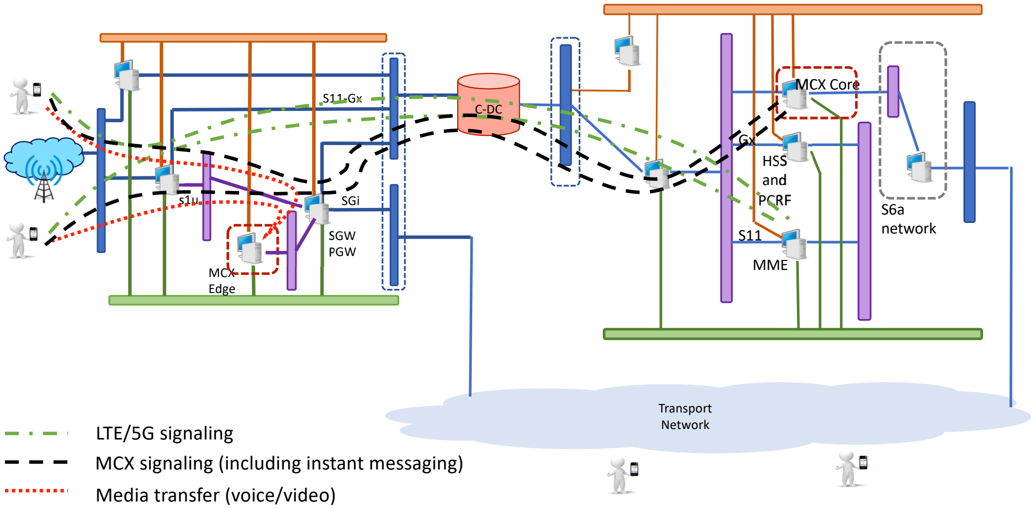
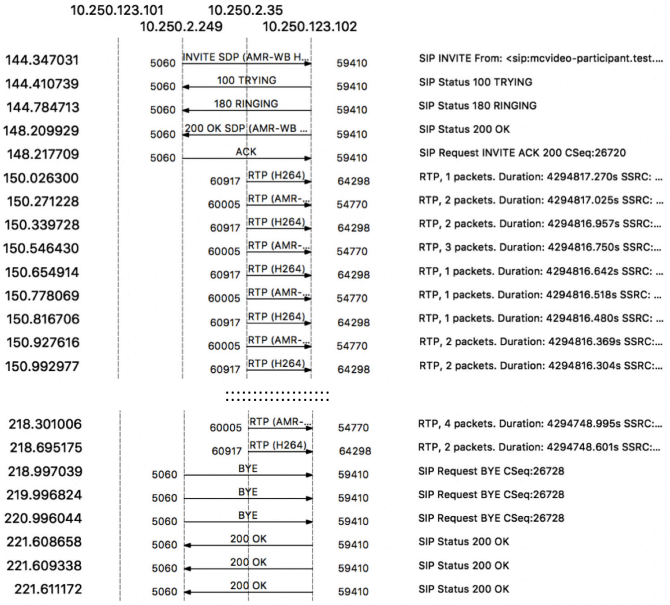


Fig. 5. Full slice blueprint with signalling and data traffic flows, for the test where both user1 and user2 are connected via the emulated eNodeB.

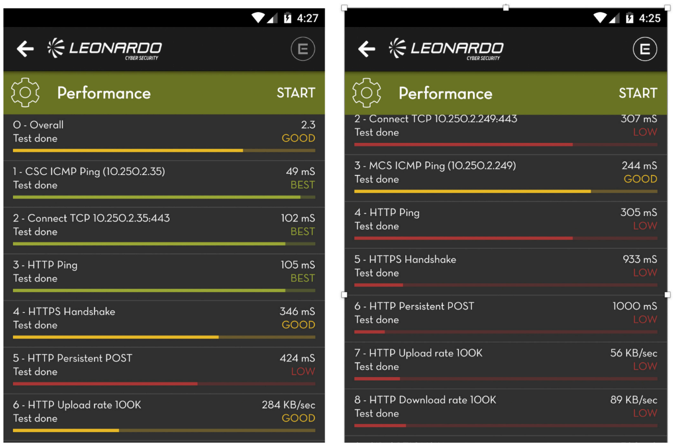


```
              10.250.123.101      10.250.2.35
                        10.250.2.249      10.250.123.102

144.347031   5060  INVITE SDP (AMR-WB H...  -->  59410   SIP INVITE From: <sip:mcvideo-participant.test....
144.410739   5060  <--  100 TRYING               59410   SIP Status 100 TRYING
144.784713   5060  <--  180 RINGING              59410   SIP Status 180 RINGING
148.209929   5060  <--  200 OK SDP (AMR-WB ...   59410   SIP Status 200 OK
148.217709   5060  ACK  -->                      59410   SIP Request INVITE ACK 200 CSeq:26720
150.026300        60917  RTP (H264)  -->         64298   RTP, 1 packets. Duration: 4294817.270s SSRC: ...
150.271228        60005  RTP (AMR-...  -->       54770   RTP, 2 packets. Duration: 4294817.025s SSRC: ...
150.339728        60917  RTP (H264)  -->         64298   RTP, 2 packets. Duration: 4294816.957s SSRC:...
150.546430        60005  RTP (AMR-...  -->       54770   RTP, 3 packets. Duration: 4294816.750s SSRC:...
150.654914        60917  RTP (H264)  -->         64298   RTP, 1 packets. Duration: 4294816.642s SSRC: ...
150.778069        60005  RTP (AMR-...  -->       54770   RTP, 1 packets. Duration: 4294816.518s SSRC: ...
150.816706        60917  RTP (H264)  -->         64298   RTP, 1 packets. Duration: 4294816.480s SSRC: ...
150.927616        60005  RTP (AMR-...  -->       54770   RTP, 2 packets. Duration: 4294816.369s SSRC:...
150.992977        60917  RTP (H264)  -->         64298   RTP, 2 packets. Duration: 4294816.304s SSRC:...

                    ::::::::::::::::::::

218.301006        60005  RTP (AMR-...  -->       54770   RTP, 4 packets. Duration: 4294748.995s SSRC:...
218.695175        60917  RTP (H264)  -->         64298   RTP, 2 packets. Duration: 4294748.601s SSRC:...
218.997039   5060  BYE  -->                      59410   SIP Request BYE CSeq:26728
219.996824   5060  BYE  -->                      59410   SIP Request BYE CSeq:26728
220.996044   5060  BYE  -->                      59410   SIP Request BYE CSeq:26728
221.608658   5060  <--  200 OK                   59410   SIP Status 200 OK
221.609338   5060  <--  200 OK                   59410   SIP Status 200 OK
221.611172   5060  <--  200 OK                   59410   SIP Status 200 OK
```

Fig. 6. SIP flows of an MCVIDEO call obtained by capturing the traffic on the callee (`user2@10.250.123.102`).

V. Conclusion

In this paper we reported the implementation of a network slicing test-bed for mission critical communications. The goal of the experiment was to test a fully virtualized infrastructure in line with the architectural principles of the 5G. A full network slice based on virtualized components was deployed according to the ETSI-MANO standard, with the MC communication functions split between core and edge, as well as between data and control plane. The test-bed validated the effectiveness of this approach and demonstrated the better performance of the MC communications when the media capabilities are kept closer to the edge, i.e. to the workforce.



4:27

Performance START

0 - Overall Test done	2.3 GOOD
1 - CSC ICMP Ping (10.250.2.35) Test done	49 mS BEST
2 - Connect TCP 10.250.2.35:443 Test done	102 mS BEST
3 - HTTP Ping Test done	105 mS BEST
4 - HTTPS Handshake Test done	346 mS GOOD
5 - HTTP Persistent POST Test done	424 mS LOW
6 - HTTP Upload rate 100K Test done	284 KB/sec GOOD

4:25

Performance START

2 - Connect TCP 10.250.2.249:443 Test done	307 mS LOW
3 - MCS ICMP Ping (10.250.2.249) Test done	244 mS GOOD
4 - HTTP Ping Test done	305 mS LOW
5 - HTTPS Handshake Test done	933 mS LOW
6 - HTTP Persistent POST Test done	1000 mS LOW
7 - HTTP Upload rate 100K Test done	56 KB/sec LOW
8 - HTTP Download rate 100K Test done	89 KB/sec LOW

Fig. 7. Screenshot of the MC application executing performance measurements towards the MCX in the edge and in the core.

Acknowledgment

This work was supported by Leonardo S.p.A. with contracts n. COLB-TEC-SSI-2018-003-A and COLB-TEC-2019-009.

References

[1] Mission Critical Services in 3GPP `https://www.3gpp.org/news-events/1875-mc_services`, visited on May 12, 2020.
[2] GSMA Association, “An Introduction to Network Slicing”, 2017.
[3] GSMA Association, “From Vertical Industry Requirements to Network Slice Characteristics”, August 2018.
[4] 3GPP TS 23.501, “System architecture for the 5G System (5GS)”, release 16.4.0, 03/2020
[5] Open Source MANO `https://osm.etsi.org/`, June 2020.
[6] LTE broadband solutions `https://www.leonardocompany.com/it/security-cyber/professional-communications/professional-broadband`, Leonardo s.p.a., June 2020.
[7] GSMA Association, “Network Slicing: Use case Requirements”, April 2018.
[8] `https://www.openairinterface.org`, March 2020.
[9] `https://nextepc.org`, March 2020.
[10] `https://www.openstack.org`, March 2020.
[11] 3GPP TS 28.530, “Management and orchestration; Concepts, use cases and requirements”, release 15.3, 12/2019.