Enabling Open Access to LTE network components; the NITOS testbed paradigm

Nikos Makris*†, Christos Zarafetas*†, Spyros Kechagias†,
Thanasis Korakis*†, Ivan Seskar§ and Leandros Tassiulas‡,
* Informatics & Telematics Institute, CERTH, Greece
† Dept. of ECE, University of Thessaly, Volos, Greece
‡ Dept. of ECE, Yale Institute for Network Science, USA
§ WINLAB, Rutgers University, USA
{nimakris, hrzarafe, spkehagi, korakis}@uth.gr, seskar@winlab.rutgers.edu, leandros.tassiulas@yale.edu

***Abstract*—The lessons already learned from the existing protocols operation are taken into deep consideration during the standardization activities of the potential technologies opted for the future 5th Generation mobile networks. Prior research on wireless technologies in general has clearly shown the need for open programmable experimental facilities which can be used for the implementation and evaluation of novel algorithms and ideas under real world settings, even directly comparable to existing technologies and methodologies. Nevertheless, provisioning of such testbed platforms mandates the respective tools which will enable access to the testbed resources and will expose the maximum possible flexibility in configuring them. In this work, we present our efforts in building such a facility, along with the tools and services that cope with such requirements. The facility upon which we build is the long-established NITOS wireless testbed, which is offering commercial as well as open source LTE components in a 24/7 basis.**

I. Introduction

The rapid penetration of smart devices in our everyday living has boosted the overall traffic exchanged over the Internet. Smart devices generate constantly growing amounts of traffic, which is in most times exchanged over a wireless access network. This growing demand for enhanced network capacity is taken into deep consideration as the community employs innovative techniques for coping with the existing spectrum crunch, in a step towards standardizing a fifth generation mobile networking protocol. Nevertheless, apart from the access networking technology, new services have to be employed in the backbone network in order to support massively generated data. Software Defined Networking (SDN) seems to be the enabler for smart services over existing networks [1].

As former research has shown ([2], [3]), the need for real world evaluation of new protocols and algorithms is very prominent in the case of wireless networks; simulators can only model the wireless channel, while several external effects present when running them under real world conditions might not be taken into consideration. This inaccuracy has driven the establishment of several facilities, which offer experimentation services and in most cases free of charge. These facilites usually reside in a lab environment, thus rendering it difficult to create the appropriate experimentation environment for emulating an Internet scale setup. Towards tackling this challenge, several efforts have been establishing the appropriate mechanisms which will enable efficient testbed interconnection, by standardizing the APIs for facilitating the federation. An excellent example of such cases are the OneLab [4] and the Fed4FIRE [5] efforts, that federate several European heterogeneous testbed islands under one common facility, accessed through a unified control tool. Yet, these facilities provide limited support for mobile broadband technologies, since these kind of resources are present and geographically limited to specific sites.

978-1-4799-7899-1/15/$31.00

A similar effort to these ones is the FLEX project [6], standing for FIRE LTE testbeds for open experimentation, which establishes experimental facilities with 4G technologies and offers them free of charge. FLEX incorporates resources that are distributed throughout Europe, in order to create a rich and diverse experimentation environment. Nevertheless, provisioning of such infrastructure mandates a unified control and management tool that will handle the different network components (i.e. base stations, switches, routers, etc.) in a common way, regardless of the actual equipment. In this paper we present our efforts towards creating such an experimental tool, based on a number of challenges; (i) the existence of a common API for configuring the different LTE network components (like base stations, EPC networks, UEs) regardless of the resources or the way that they are configured, (ii) being easily extensible for incorporating support of more components, (iii) supporting for user-defined routing of data, beyond the LTE network, (iv) providing a standardized API for interconnecting different LTE testbeds, (v) supporting user defined SDN control beyond the EPC network and (vi) providing completely isolated slices of the LTE infrastructure, so that concurrent experiments can be executed by different users. For the rest of the paper, we present our efforts in establishing tools that handle these requirements and apply it over two sites of the FLEX facility.

The rest of the paper is organized as follows. In section II we illustrate the components available in the existing facilities, that are/will be supported by our tool. In section III we provide some information on the design of our unified tool and the way that it copes with the afore-set challenges. In section IV we showcase some proof of concept experiments and indicative results on the experimentation potential enabled by our approach. Finally, in Section V we present some similar efforts and we conclude in section VI.

II. Testbed Components

The framework that we design is applied over the federated FLEX European testbed. FLEX testbed consists of three independent testbed facilities; NITOS Future Internet facility [7], provided by University of Thessaly in Greece, w-iLab.t wireless testbed [8], provided by iMinds in Belgium and OpenAirInterface testbed [9], offered by EURECOM in France.

These three facilities are able to provide free-of-charge access to experimental 4G networks, based on real LTE equipment. Their setups span from femtocell to macrocell deployments, based on a rich heterogeneity of LTE components; from commercial femtocells (ip.access LTE245F), commercial macro-scale base stations (Airspan Air4GS WL), commercial EPC networks (SiRRAN EPC) to open source femto- and macro-cell setups using the OpenAirInterface platform [10] and the respective open source EPC solution. The UEs adopted by all the FLEX sites are either commercial, like the Huawei E392 USB dongles and the LG Nexus 5 Android smartphones, or open source by setting the OpenAirInterface compatible platforms to operate as UEs.

This high heterogeneity of resources and their divergent configuration APIs can create a significant overhead for an experimenter who takes access over these resources. Therefore, a common framework which will reduce the steep learning curve for configuring the different resources is mandatory. As such, in the context of the FLEX project we develop a common framework for managing and controlling the testbed components through a common API.

As our base framework, we exploit the cOntrol and Management Framework (OMF), which is already considered the state-of-the-art management and experiment control framework for the experimental facilities worldwide. OMF is consisting of three entities; (i) the OMF Experiment Controller, used for parsing an experiment description file and creating the appropriate OMF messages for configuring the involved entities, (ii) the OMF Resource Controller (RC) used for parsing the OMF messages and translating them to the appropriate commands, based on the underlying resource, and (iii) the OMF Aggregate Manager (AM) entity, for performing administrating actions on the target testbed, such as loading the appropriate configuration images on the testbed's nodes, turning them on/off, etc.

For the rest of the paper and the design of our service, we focus on the OMF AM entity, while we extend the core framework appropriately in order to support the LTE resources. We choose to extend this entity, since it provides an REST based API for accessing any service and has built-in support for representing data through an XML-based API. This approach is welcomed by the GENI experimental testbeds in the US as well, since the services for controlling the WiMAX components are based on a similar design.

III. Control and Management of LTE entities

The service design for controlling and configuring the different LTE components has to cope with several different identified challenges. In the following subsections we analyze these challenges and the design of our service in order to overcome some inherent limitations posed by the current hardware deployment.

A. Challenges in provisioning an LTE testbed

1) Common API for configuring the different LTE network components: Regardless of the underlying hardware, a common API for configuring the resources has to be employed. Incompatibilities in the component configuration mandate the definition of a common Northbound API, that the experimenters interface with, while dedicated Southbound APIs are able to configure the different hardware components, similar to [11]. Such incompatibilities rely on the different access method on configuring the components provided by each vendor; for example, the ip.access femtocells are configured by establishing a secure shell connection on them and altering database entries regarding their operation, whereas the OpenAirInterface cells are configured via a configuration file, while the Airspan configuration can be changed via SNMP commands.

To this end, the service has to provide a common interface for setting similar parameters, such as the transmission power, and appropriately map it to the respective command, based on the underlying hardware. This design approach provides the opportunity for the experimenter to remain totally agnostic of the underlying components and the way that the configuration is implemented, thus reducing the learning curve for experimenting with the resources.

2) Easily extensible for incorporating support of more components: The design of such a service should be modular; since a common API will be employed for the Northbound interface, the Southbound should be easily extensible in order to easily support independently deployable resources. Therefore, the service has to employ a discovery phase where the LTE resources are identified, and the appropriate resource driver is activated (eg. the respective driver for each one of the EPC components).

3) Support for user-defined routing of data: As it is already happening in the majority of European testbeds, an interconnection over the GEANT network is available. Therefore, the user should be able to define the manner in which the traffic will be routed, similar to existing approaches in the GENI WiMAX sites. Tailored to the European testbed specifications, the user shall be able to define as to whether traffic stemming from the UEs will be relayed to the Internet or the GEANT network.

4) Providing a standardized API for interconnecting different LTE testbeds: A prerequisite for enabling large scale experimentation with distributed resources is establishing testbed federations. To this aim, since the user already should be able to define the target network to which the traffic will be relayed, a standardized API should be adopted that will ease the testbed interconnection. This poses some additional challenges, as the way that the different EPC networks can provide an interface to external to the LTE network entities differs from each PDN-GW implementation; for example the SiRRAN EPC network adopted in NITOS handles the ARP requests inbound to the PDN-GW by always replying with the same MAC address. Such limitations should be handled appropriately for the correct provisioning of the LTE testbed.

5) User defined SDN support beyond the EPC network: Towards enabling potential 5G related services, the user should be able to interface the appropriate API for handling the data behind the EPC network. To this aim, the service shall employ SDN techniques in order to enable user defined routing of data, apart from the selection of the target network to which traffic will be sent. Consequently, we shall employ techniques which are already available in similar approaches, by enabling the user to submit a software defined description for handling the data. From our point of view, the SDN enablers that can be utilized in such production networks are the OpenFlow protocol or the Click modular router.

6) Providing completely isolated slices of the LTE infrastructure: Since the infrastructure is offered free-of-charge to experimenters, concurrent access to the resources should be provided. To this end, a detailed scheme for slicing support has to be employed, that will enable transmissions with guaranteed bit-rates and individual access to the resources, even on the wired backbone beyond the EPC network. Hence, VLAN tagging shall be used for the wired backbone, and a similar technique for the LTE network.

B. LTErf Service Design

In order to cope with the aforementioned challenges, our service named as *LTErf* has been designed by adopting the architecture illustrated in Figure 1. The service is built as an OMF AM service, able to provision a REST based interface for accessing the different components. Since a common API has been adopted for the resources, we have identified the following sub-services that can be configured using the same API:

- Base stations: The wireless parameters, as well as the configuration of the base stations regarding their EPC interconnection should be the same among different vendors of hardware. Examples of such common parameters are the channel bandwidth, transmission power, etc.
- EPC networks: Similar to the base station approach, different EPC networks should provide similar functionality and thus provide the same API for configuring them. Examples of such configurations are the different network configurations (IP addresses and ports for the S1-MME, S11, S6, S1-AP, etc. interfaces), Access Point Names (APNs) that will be used, etc.
- Datapath configurations: Setting a datapath, meaning the way that traffic will be routed beyond the EPC network, through a common API, regardless of the datapath chosen (eg. Internet/GEANT). For the cases of the GEANT network, the experimenter can set a VLAN tag for the traffic that will be exchanged, thus creating an end-to-end isolated slice on the wired network.
- Monitoring functions: As the equipment is already providing an API for the collection of network performance measurements, the service appropriately handles them and visualizes them to the end user.

As seen in Figure 1, the service has been developed in a modular way. The different Northbound interfaces for the sub-services are mapped to resource specific drivers for controlling and configuring the diverse components. These drivers consist the Southbound interface, written in the Ruby language, able to handle the different methods of accessing the resources (e.g. SNMP/SSH access for the components). Upon service start-up, a configuration phase is employed where the available resources (specified in a configuration file given to the service) are discovered and identified. During this phase, these drivers are initialized and set-up. From now on, the user is interacting with the web interface of the service, by addressing each resource using an identifier, like for example node1/node2 for the different base stations involved. The service parses any requests and delivers them to the appropriate driver for setting the respective resource.

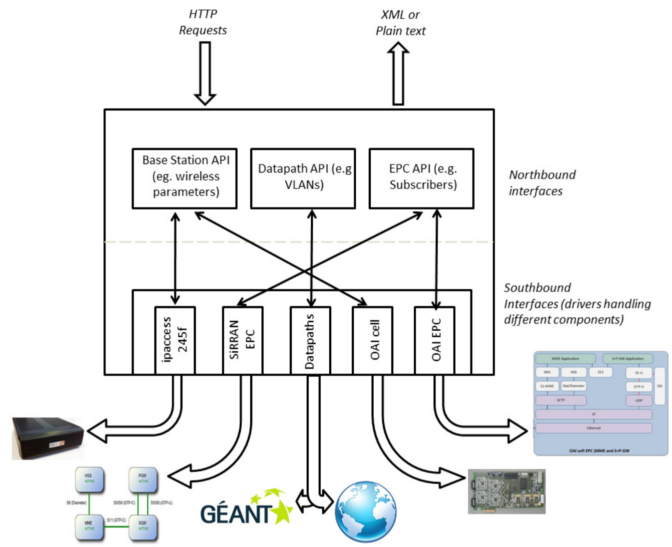

Fig. 1. LTErf architecture; different modules on the southbound interface are used to configure the different components

Regarding the datapath configuration, we have to employ SDN techniques in order to cope with the wired backbone challenges 3, 4, 5 and 6. Towards this objective, the service is using the well established Open-vSwitch framework ([12]) to create wired switches among the PDN-GW interface and the selected network for relaying traffic (Internet/GEANT). Moreover, in the case of the Internet connection, the service is able to initialize a NAT service in order to enable routing of data over the Internet. In the case of the GEANT network connection, the service employs more SDN techniques; by setting up a custom OpenFlow controller, launched as a *trema* instance ([13]), we manage to setup the appropriate rules for directly exposing the LTE resources to the GEANT network. Using such an approach, we manage to fulfil both challenges 4 and 5. The bridge interface can be set up to use almost any controller possible (apart the one generated by the service), thus enabling user based software defined routing of the traffic.

As the first EPC to be supported by the LTErf service is SiRRAN LTEnet, we coped with several factors in order to expose accordingly the LTE resources to the external to the EPC network. Since no ARP protocol is used on the LTE access network, and until data reaches the EPC, the EPC service is endowed with the process of handling the ARP messages for the data incoming to the EPC for the PDN-GW and towards a UE. As the address with which the EPC replies to any ARP request destined to a UE is always the same, we had to create a book-keeping mechanism for mapping the appropriate traffic flows to each UE. To this aim, the service is able to generate dynamically an OpenFlow controller, that is able to appropriately map each request to each client based on the APN they use, and establish accordingly the traffic flows.

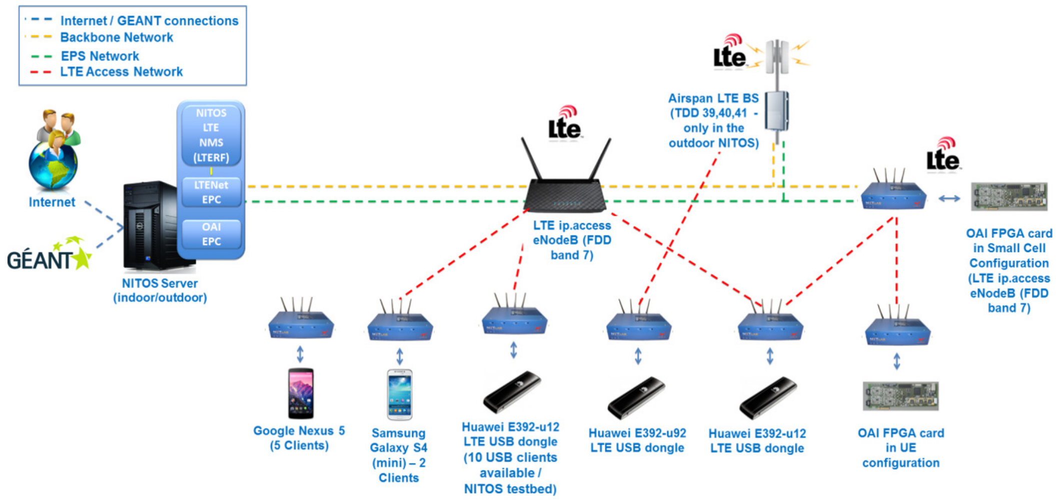

Fig. 2. The NITOS LTE Testbed Architecture; two different deployments in an indoor RF isolated and outdoor prone to RF interference setups with diverse LTE equipment.

Finally, as far as the slicing scheme is concerned, we manage to slice the network end-to-end in the following way. Regarding the LTE components, we use the protocol's ability to setup multiple APNs for slicing the network by providing bit-rate guarantees (via the EPC network) per each UE, thus enabling provisioning of different isolated slices on top of the same infrastructure. In the cases of supporting services evaluation over the testbed, this slicing scheme is sufficient given the variety of the resources existing in each testbed. Yet, in the cases of evaluating ideas and algorithms residing in the lower layers of the LTE protocol stack (e.g. MAC scheduling), complete access to the resources is mandatory. Regarding the rest of the backbone network components, the service has been enhanced to deal with several VLAN interfaces.

By employing the aforementioned slicing techniques, each experimenter can send traffic from the testbed nodes in the context of an APN. Based on the existing implementation of the SiRRAN EPC that we support, traffic exchanged within an APN is delivered to different PDN-GW interfaces at the EPC. Using this information, the traffic is able to bridge the APN-specific PDN-GW interface to an outgoing interface, tagged using a VLAN number set by the experimenter. Since the GEANT virtual circuits are delivered over VLAN interfaces, the service has to deal with QinQ VLANs in order to sufficiently provide such slicing. By adopting this setup, traffic can flow and federation among LTE testbeds can be realized at the MAC layer of the resources.

IV. Experimental Results

In this section we provide some proof of concept experimental results of the testbed operation and experimentation capabilities, applied in the NITOS testbed of the FLEX facility. The NITOS testbed is actually comprised of three different testbeds, offering diverse experimentation services; the NITOS outdoor testbed, which is prone to external RF interference, the NITOS indoor testbed, which is a completely RF isolated testbed, and finally the NITOS office testbed, deployed in an office environment. Their interconnection is based on a backbone connection provided by the Greek NREN, thus utilizing a part of the pan-European GEANT network.

For the demonstration of the capabilities of our service, we conduct two proof-of-concept experiments. One of them is involving base station configurations and collection of statistics, while the second one is utilizing the federation of the NITOS indoor and office testbeds. In order to showcase the demonstration capabilities, we choose to interconnect the NITOS indoor and office testbeds, as they offer diverse experimentation services. The NITOS indoor node is equipped with one ip.access femtocell and several UEs, while the NITOS office testbed is offering experimentation with one Airspan WiMAX base station and several WiMAX clients. For the realization of such a federation effort, we had to enhance the existing service for managing the WiMAX base station with the datapath support. Differences in the way that the two datapath services are configured lie on the different operation of the two protocols; LTE is mapping different VLAN interfaces to different APNs, whereas WiMAX uses the MAC and IP address of each WiMAX client to map it to the appropriate external VLAN.

A. *Performance measurement of the LTE DL channel*

The experimenter is able to login to the NITOS portal server and get access to the different services that are provided during the specific timeslot, within which he/she has an active reservation. Using the NITOS Scheduler [14] a NITOS user can define a specific timeslot that wants to use the resources and during that slot, the appropriate mechanisms are set in the testbed to allow access to the *LTErf* service.

The components that we use for this experiment are only the LTE access network with the commercial femtocells, two UEs and SiRRAN's EPC network. Using a predefined set of commands, we perform the following actions:

1) We establish a secure shell connection to the NITOS portal server. From this point on, we can have direct access to the NITOS nodes hosting the UEs.
2) By accessing the NITOS nodes, we are able to configure the LTE UEs over a serial port, by sending the appropriate AT commands to the dongle.
3) Within the AT commands, we pass as an argument the APN that we want to connect with.

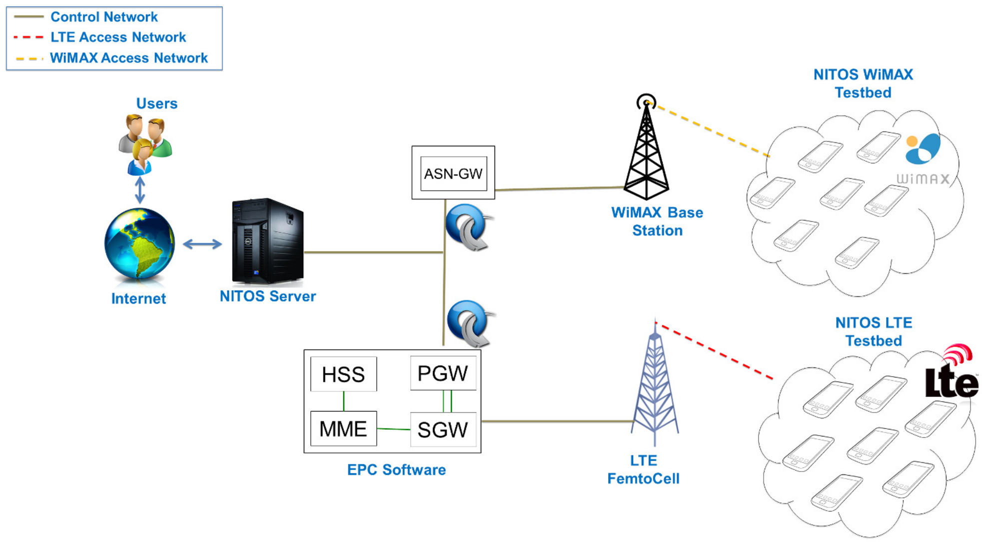

Fig. 3. Experimental topology for federating the WiMAX and LTE testbed islands; our techniques enable for user-defined OpenFlow support beyond the EPC network or an ASN-GW.

4) Once the connection is established, we are able to directly communicate with the PDN-GW component of the EPC network.

After setting up the resources, we are able to use the LTE network and communicate with all the LTE components under the same APN. In order to setup the proper forwarding mechanisms, we need to use the *LTErf* in order to appropriately configure the datapath. To this aim, we enable the *Internet* datapath, which establishes a NAT process beyond the PDN-GW entity and routes the traffic stemming from the LTE network to the Internet. For the needs of this experiment, we will measure the DL performance of the LTE channel, under different Modulation and Coding Schemes (MCS) settings. The femtocells and UEs are configured to use 2x2 MIMO, while the DL channel is set to use the LTE band 7 center frequency of 2655 Mhz (EARFCN equal to 3100) with a channel bandwidth of 10Mhz and a transmission power equal to 15dBm.

In order to configure the different MCS profiles in the femtocells, we use the LTErf service to disable the CQi reporting of the UEs to the eNB, and appropriately set the different MCS profiles to be used by the eNB. Upon each change we need to restart the eNBs in order for the changes to take effect. Once the eNBs are operating, we reconnect each client and start a traffic generator application measuring the DL UDP throughput of the LTE channel. As our traffic generator we use the *iperf* tool, running for 30 seconds for each measured profile. We are measuring only UDP traffic, as we want to measure the maximum throughput that each MCS profile can achieve. Finally, the experiments are conducted under ideal conditions, without any external interference. The mean values of RSSI and RSRP that we logged during our experiments where at least -53 dBm and -76 dBm respectively. Table I showcases our measured results over the deployed infrastructure.

B. User-defined Datapath Configuration

For our second set of experiments, we will utilize the API that the *LTErf* service establishes to facilitate federations. To this aim, we adopt the architecture illustrated in Figure 3.

TABLE I. MEASURED DL THROUGHPUT OF THE LTE CHANNEL

MCS Profile	DL UDP Throughput	MCS Profile	DL UDP Throughput
28	69.1 Mbps	13	22.1 Mbps
27	61.5 Mbps	12	19.2 Mbps
26	59.2 Mbps	11	17.0 Mbps
25	55.2 Mbps	10	15.4 Mbps
24	53.1 Mbps	19	15.5 Mbps
23	48.9 Mbps	8	13.5 Mbps
22	44.6 Mbps	7	12.0 Mbps
21	41.0 Mbps	6	9.99 Mbps
20	38.5 Mbps	5	8.48 Mbps
19	35.6 Mbps	4	6.97 Mbps
18	31.5 Mbps	3	5.50 Mbps
17	29.4 Mbps	2	4.24 Mbps
16	29.4 Mbps	1	3.44 Mbps
15	27.4 Mbps	0	2.63 Mbps
14	25.0 Mbps	-	-

By utilizing the GEANT connectivity of the NITOS LTE and NITOS WiMAX testbeds, we are able to setup a dedicated end-to-end communication channel, from an LTE UE to a WiMAX interface.

We setup the two testbeds to interconnect using a dedicated VLAN number, which in the LTE network is mapped to a specific PDN-GW (based on the APN the UE uses) while in the WiMAX network the VLAN is mapped to a specific set of MAC and IP addresses that belong to the interface mounted on a NITOS node. As we send the appropriate HTTP request to the *LTErf* service, we instruct the service to create a QinQ VLAN to the existing GEANT interface and bridge it by using Open-vSwitch with the PDN-GW interface that belongs to a specified APN. As we already explained, the service is also generating an OpenFlow controller based on the Trema framework [13], and launches it in order to appropriately expose the traffic to the GEANT network. Using a similar approach, we perform the same actions to the WiMAX gateway server as well.

For the needs of this experiment, we enable the highest available MCS profiles for the UL and DL channels of the LTE network, involving the same setup as in the first experiment.

TABLE II. PING DELAYS FOR THE DIFFERENT NETWORK COMPONENTS INVOLVED IN THE EXPERIMENT

Measured Component	Delay (ms)
LTE access delay	45ms
WiMAX access delay	75ms
LTE to WiMAX servers (wired network with a wireless hop)	2.5ms
LTE UE to WiMAX client	125ms

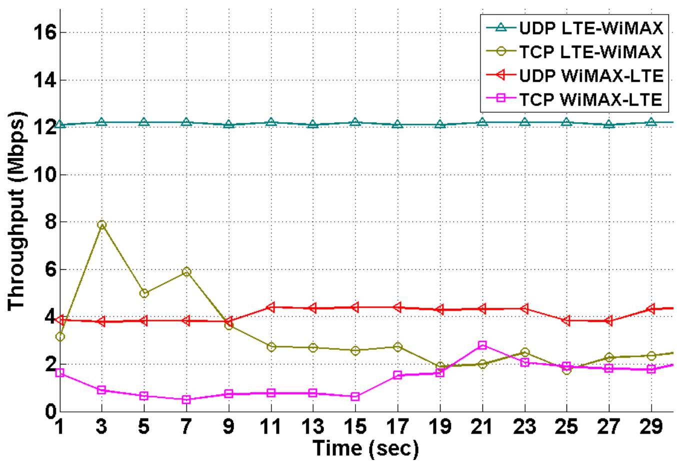

Fig. 4. Throughput experiment results with the federated infrastructure.

For the WiMAX network, we enable Adaptive MCS for the static NITOS node, and set the WiMAX base station to use the output power of 20dBm. The CINR and RSSI observed at the WiMAX client is measured approximately 25.8 dB and -69 dBm respectively. Table II illustrates the measured delays over the network. Similarly, Figure 4 demonstrates the achieved throughput performace for the UL/DL and DL/UL pairs of the LTE/WiMAX networks.

V. Related Work

Similar techniques as the ones we propose have been used in the past for the efficient testbed provisioning. An architecture for slicing schemes for WiMAX testbeds has been proposed in [15] where testbed access is provided to different users by deploying different VMs of the WiMAX management services. In the same context, virtual WiFi has been introduced in [16], where Virtual Access Points are deployed for serving different users or subsequently testbed slices. Our approach is targetting at a simpler approach, by utilizing the existing protocol operation and serving totally end-to-end isolated slices of the infrastructure, mainly targetting at services/applications evaluation over the facilities.

LTE-specific slicing support has been proposed in [17], where the authors slice the LTE RAN by allocating and isolating radio resources via a slice manager. Similar to this, authors in [18] slice the radio network using dynamic scheduling techniques. Yet, they are not directly applicable to any existing real system, and do not take into consideration the network end-to-end.

VI. Conclusion

In this work, we presented our efforts in building the *LTErf* service which provides the appropriate mechanisms for handling LTE testbed components. The service has been developed in a modular way, so that new resources can be easily supported. A highlight of the service is the support for OpenFlow based user routing beyond the LTE network. Although this approach seems to be handling the LTE network as a black-box, we provide the appropriate hooks for user-defined setting of the EPC and Base Station network components.

Since the service is able to provide isolated experimental slices with guaranteed bit-rates per client, we foresee the definition of NFV functions and their implementation over the deployed infrastructure. By implementing such an approach, combined with isolated end-to-end slices as it has been illustrated, we believe that real testbed infrastructure can serve as a vehicle for the better exploitation of such facilities from Mobile Virtual Network Operators, for the evaluation of their services prior to introducing them to the market.

Acknowledgment

The research leading to these results has received funding from the European Unions Seventh Framework Programme under grant agreement no 612050 (FLEX Project).

References

[1] L.E. Li, Z.M. Mao, and J. Rexford. Toward software-defined cellular networks. In *Software Defined Networking (EWSDN), 2012 European Workshop on*.

[2] David Kotz, Calvin Newport, Robert S. Gray, Jason Liu, Yougu Yuan, and Chip Elliott. Experimental evaluation of wireless simulation assumptions. In *Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '04.

[3] Krzysztof Pawlikowski. Do not trust all simulation studies of telecommunications networks. In *International Conference on Information Networking*.

[4] Serge Fdida, Timur Friedman, and Thierry Parmentelat. Onelab: An open federated facility for experimentally driven future internet research.

[5] Fed4FIRE project: Federation for FIRE testbeds, http://www.fed4fire.eu.

[6] FLEX project: FIRE LTE testbeds for Open Expeirmentation, http://www.flex-project.eu.

[7] Network Implementation Testbed Laboratory, http://nitlab.inf.uth.gr,.

[8] w-iLab.t wireless facilities, http://www.flex-project.eu.

[9] The OpenAirInterface testbed,, http://www.openairinterface.org.

[10] Navid Nikaein, Mahesh K. Marina, Saravana Manickam, Alex Dawson, Raymond Knopp, and Christian Bonnet. Openairinterface: A flexible platform for 5g research. *SIGCOMM Comput. Commun. Rev.*

[11] R. Riggio, K.M. Gomez, T. Rasheed, J. Schulz-Zander, S. Kuklinski, and M.K. Marina. Programming software-defined wireless networks. In *Network and Service Management (CNSM), 2014 10th International Conference on*.

[12] Open-vswitch, an open virtual switch, http://openvswitch.org.

[13] "Trema: Full-Stack OpenFlow Framework in Ruby and C", http://trema.github.io/trema/.

[14] NITOS Scheduler tool for managing testbed resources, https://github.com/NitLab/NITOS-Scheduler.

[15] Gautam Bhanage, Ivan Seskar, Rajesh Mahindra, and Dipankar Raychaudhuri. Virtual basestation: Architecture for an open shared wimax framework. In *Proceedings of the Second ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures*, VISA '10.

[16] G. Bhanage, D. Vete, I. Seskar, and D. Raychaudhuri. Splitap: Leveraging wireless network virtualization for flexible sharing of wlans. In *Global Telecommunications Conference (GLOBECOM 2010)*.

[17] Aditya Gudipati, Li Erran Li, and Sachin Katti. Radiovisor: A slicing plane for radio access networks. In *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*.

[18] M.I. Kamel, Long Bao Le, and A. Girard. Lte wireless network virtualization: Dynamic slicing via flexible scheduling. In *Vehicular Technology Conference (VTC Fall), 2014 IEEE 80th*.