

4

A Platform for 4G/5G Wireless Networking Research, Targeting the Experimentally-Driven Research Approach – FLEX –

**Nikos Makris¹, Thanasis Korakis¹, Vasilis Maglogiannis²,
Dries Naudts², Navid Nikaein³, George Lyberopoulos⁴,
Elina Theodoropoulou⁴, Ivan Seskar⁵, Cesar A. Garcia Perez⁶,
Pedro Merino Gomez⁶, Milorad Tosic⁷, Nenad Milosevic⁷
and Spiros Spirou⁸**

¹University of Thessaly

²Ghent University – iMinds

³EURECOM

⁴COSMOTE

⁵WINLAB, Rutgers University

⁶Universidad de Malaga

⁷University of Nis

⁸INTRACOM SA Telecom Solutions

4.1 Introduction

The proliferation of smart mobile devices and data hungry mobile applications are driving the demand for faster mobile networks. Long Term Evolution (LTE), the 4th Generation of mobile network technology standardized by the 3rd Generation Partnership Project (3GPP) [1], aims at satisfying this demand by offering faster connection speeds at both the downlink and the uplink, increased network capacity and better coverage. The rapid penetration of LTE in different countries creates a vast field for innovation in terms of mobile broadband services. At the same time, research for the next generation mobile networks has already begun with the examination and evaluation of candidate technologies and architectures. Given the practical requirement

for backward compatibility between successive technologies, it is rational to assume that these technologies, often referred to as Beyond 4th and towards the 5th Generation (B4G and 5G), will naturally evolve from the extension of LTE with new advanced features.

Evaluation of the performance of innovative broadband services over LTE and of candidate post-LTE technologies requires rigorous testing and validation. While network simulation software has evolved significantly over the years, it cannot still capture the complex real world environment, and field tests are still considered essential at the late stages of development. To that end, the existence of network testbed facilities plays a significant role in understanding the complexities associated with real use and therefore in building better solutions.

In Europe, since its establishment in 2008, the Future Internet Research and Experimentation (FIRE) initiative [2] has contributed in bridging the gap between visionary research and large-scale experimentation on new networking and service architectures and paradigms. Through the successful organization of several waves of research projects, an extensive and multidisciplinary open network testbed facility has been developed. Despite the diversity in the FIRE facilities in terms of available infrastructure and access technologies, a lack of truly open and operational LTE testbeds had been identified (and cellular testbeds in general). By “open” we mean that the facilities are available to external experimenters and that the latter can configure the testbed to some extent, according to their needs. By “operational” we refer to flexibility in accessing the core, gateways, access points and user equipment of the testbeds, and the capability to run full end-to-end services.

This lack was certainly not due to reduced interest from the community. On the contrary, there is a steadily increasing demand from the research community, including the industry, to have access to LTE and beyond experimentation facilities in different countries. However, the constraints typically posed by operators and large vendors, typically due to commercial considerations, restrict the configuration capabilities to an extent, which usually discourages testbed operators from deploying such infrastructure.

FLEX (FIRE LTE testbeds for Open Experimentation) [3] aims to remove these constraints through the development of a truly open and operational LTE experimental facility. Based on a combination of **truly configurable commercial equipment, truly configurable core network software, fully open source components, and on top of those, sophisticated emulation and mobility functionalities**, this facility allows researchers from academia

and industry to test services and applications over real LTE and beyond infrastructure, or experiment with alternative algorithms and architectures of the core and access networks.

4.2 Problem Statement

Several EU funded projects have paved the way for the federation of isolated testbed islands across Europe. Excellent examples of them are the OpenLab [4] and the Fed4FIRE [5] projects, which have addressed both the control and experimental plane federation of heterogeneous FIRE resources. With the control plane we mean the way that the resources are discovered, represented and reserved inside federations, whereas with the experimental the option to include resources from heterogeneous testbeds, decoupled from their geographical location, and bundle them in one single large scale experiment. Yet, the focus on these federations lies only on the support of generic nodes, meaning just an abstract representation of any testbed resource, with a limited number of parameters being defined by the experimenters.

FLEX is addressing this lack of experimentation services for LTE and beyond resources, by integrating all the LTE hardware extensions to the state-of-the-art control and management services of the testbeds. Three core FIRE testbeds have been extended with LTE support initially, and two more have been added to the consortium after the completion of an infrastructure upgrade Open Call process. All of the FLEX testbeds, have been federated over the GÉANT network [6], thus enabling dedicated guaranteed end-to-end connections from one testbed to another able to bear the traffic, and the setup of novel experiments for decentralized architectures.

Moreover, FLEX is offering two setups; 1) a commercial equipment based testbed, for the development of novel services and 2) an open-source setup for the development and evaluation of new protocols, leveraging the LTE protocol stack. The commercial equipment is fully programmable, provided by the partners of the project, and through the definition of high level APIs, experimenters can take access over them. As for the open source solution, the project is using the open source solution of OpenAirInterface (OAI) [7], that allows the execution of a full stack LTE eNodeB or User Equipment (UE) over commodity hardware with a compatible RF front-end.

The testbeds that are available within FLEX are publicly available 24/7, remotely accessible and provided free-of-charge. The five experimental facilities, along with their capabilities, are detailed in the following subsection.

4.2.1 FLEX Testbeds

The five experimental facilities, that are comprising the FLEX testbed, are resources rich in heterogeneous equipment, each of them allowing the configuration of several parameters along with the LTE configurations, and enabling the experimentation at a very large scale. Following, we list the capabilities of the five different FLEX islands (see Figure 4.1).

4.2.1.1 NITOS testbed

NITOS testbed [8], is a heterogeneous testbed, located in the premises of University of Thessaly (UTH), in Greece. The testbed facilitates access to open source and highly configurable equipment, allowing for innovations through the experimental evaluation of protocols and ideas in a real world environment. The experimental ecosystem is consisting of several wireless and wired networking components, coupled with powerful nodes and a cloud computing infrastructure. The key equipment components in NITOS are the following: 1) Over 120 nodes equipped with IEEE 802.11 a/b/g/n/ac compatible equipment, and using open source drivers. The nodes are compatible also with the IEEE 802.11s protocol for the creation of wireless mesh networks, 2) Commercial off-the-shelf (COTS) LTE testbed,

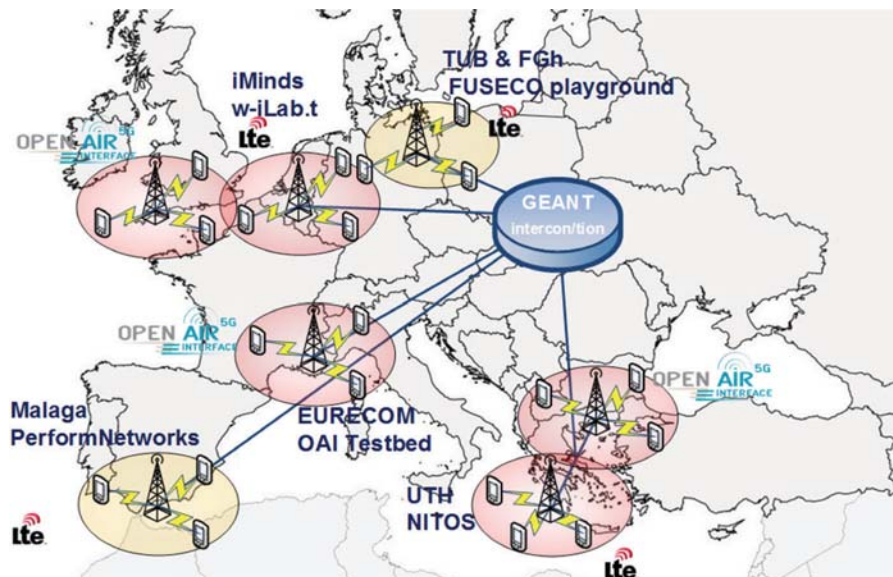


Figure 4.1 The FLEX testbed federation in Europe.

consisting of a highly programmable LTE macrocell (Airspan Air4GS), two femtocells (ip.access LTE 245F), an experimenter configurable EPC network (SiRRAN LTE net) and multiple User Equipment (UE), such as USB dongles and Android Smartphones, 3) Open Source LTE equipment, running over commodity Software Defined Radio (SDR) equipment, by the adoption of the OpenAirInterface [7] platform. OpenAirInterface can be set to operate as either a femtocell or UE, whereas its accompanying open source network is provided (OpenAirCN), 4) COTS WiMAX testbed, based on a highly programmable WiMAX base station in standalone mode, along with several open source WiMAX clients (USB dongles and Smartphones), 5) A Software Defined Radio (SDR) testbed, consisting of 10 USRPs N210, 8 USRPs B210, 2 USRPs X310 and 4 ExMIMO2 FPGA boards. MAC and PHY algorithms are able to be executed over the SDR platforms, with very high accuracy, 6) The nodes are interconnected with each other via 5 OpenFlow hardware switches, sliced using the FlowVisor framework, and allowing multiple experimenters control the traffic generated from their experiments using any OpenFlow controller, 7) a Cloud Computing testbed, consisting of 96 Cores, 286 GB RAM and 10 TBs of hardware storage. For the provisioning of the cloud, OpenStack is used.

The equipment is distributed across three different testbed locations, and can be combined with each other for creating a very rich experimentation environment. The nodes are running any of the major UNIX based distributions.

4.2.1.2 w-iLab.t testbed

The w-iLab.t [9] is an experimental, generic, heterogeneous wireless testbed and provides a permanent testbed for development and testing of wireless applications. w-iLab.t hosts different types of wireless nodes: sensor nodes, Wi-Fi based nodes, sensing platforms, and cognitive radio platforms. Each of the devices can be fully configured by the experimenters. The wireless nodes are connected over a wired interface for management purposes. This interface can also be used as a wired interface. Hence, heterogeneous wireless/wired experiments are possible. Furthermore, iMinds hosts the Virtual Wall, which consists of 2 testbeds:

- Virtual Wall 1 containing 206 nodes
- Virtual Wall 2 containing 159 nodes

The Virtual Wall offers network impairment (delay, packet loss, bandwidth limitation) on links between nodes and is implemented with software impairment. Additionally, some of the nodes are connected to an OpenFlow switch

to be able to do OpenFlow experiments in a combination of servers, software OpenFlow switches and real OpenFlow switches. Moreover, the following equipment has been installed in order to enable LTE experimentation in the testbed: 1) 2 ip.access LTE femtocells, 2) SiRRAN LTE net EPC solution with 9 licenses, 3) 22 LTE UEs as USB dongles, 4) 2 Emulated Mobility Frameworks consisting of 4 (big) and 3 (mini) shielded boxes respectively. The boxes are interconnected with each other via COAX cables. The attenuation of the RF components that are placed in the boxes is controlled by programmable attenuators, 5) 2 additional ip.access femtocells accompanied by 2 LTE dongles that are part of the (big) Emulated Mobility Framework, 6) 2 ExMIMO2 FPGA boards and 3 USRPs B210 equipped with RF front-ends compatible with OpenAirInterface. The testbed is also using 20 programmable moving robots, that can be used for real mobility experiments [10]. The users are able to draw interactively a trajectory that each robot will follow during their experiment. Each of the robots is equipped with a Nexus 6P smartphone to enable LTE experimentation. The control of the LTE experimentation can be done using Signal and Spectrum Analyzers or a USRP N210 equipped with an LTE compatible RF front-end.

4.2.1.3 OpenAirInterface testbed

Facilities at EURECOM that are available to the project include an 8-node testbed, equipped with the OAI compatible RF front-ends, UEs and VMs acting as core networks. The OAI testbed [11] nodes include: 1) 4 machines that can be used for running OAI as eNodeB, equipped with the appropriate SDR platforms (2 of them using USRPs B210 and 2 of them ExMIMO2), 2) Dedicated services are executed on top of them, for the orchestration of the experiments, such as OpenStack [12] and JuJu [13], 3) 4 nodes that are equipped with COTS UEs, that can be used for running the OpenAirCN platform (OAI EPC), 4) 2 more UEs as Android Smartphones.

4.2.1.4 PerformNetworks testbed

PerformNetworks [14], formerly PerformLTE, provides multiple scenarios to enable experimentation with different levels of realism [15]. The testbed has been extended in the project with interoperability tools that have been used to perform interoperability testing with equipment available in other FLEX testbeds. Currently, the federated part of the testbed is composed by: 1) T2010 conformance testing units by Keysight Technologies, that can be used to provide LTE end to end connectivity to commercial UEs in any standardized FDD or TDD band. These units have been extended during project to support

communication with standard core networks. 2) LTE release 8 small cells (Pixies) by Athena Wireless working on band 7. 3) Polaris Core Network Emulator (EPC), providing multiples instances in SGW, PGW, MME, HSS and PCRF (more details in [16]). This EPC has been successfully integrated with macro and pico-cells from Alcatel Lucent and with small cells from Athena Wireless and Sirran Technologies, 4) Several LTE UEs, working on different bands, successfully integrated with the T2010 units and the small cells, 5) ExpressMIMO2 and USRP SDR cards, 6) SIM cards from a Spanish LTE operator to be used on commercial deployments.

4.2.1.5 FUSECO playground

FUSECO Playground [17] allows FLEX experimenters to execute even larger scale experimentation with more LTE resources, in handover with 2G, 3G, Wi-Fi, and in collaboration with cloud services. FUSECO integration with the existing FLEX infrastructure adds values by supporting 5G research activities with NFV, SDN, etc. The hardware resources that FUSECO playground is offering to FLEX are summarized in the following: 1) ip.access LTE 245F eNodeB, supporting LTE FDD bands 7 and 13, 2) OpenEPC 3GPP Evolved Packet Core, 3) Virtualized LTE Network Functions (e.g. PDN-GW, SGW, MME) over SDNs, 4) 3 LTE dongles UEs and 3 Android Smartphones, 5) ip.access Nano3G E16 (model 239A) UMTS IMT 2100 (supporting LTE FDD bands 1, 2/5 and 4), 6) 3 Wi-Fi APs Cisco Aironet 3602e (supporting 802.11 a/b/g/n/ac), 7) Radio Signal Attenuation System with a frequency range from 700 MHz to 3 GHz, allowing the configuration of attenuation of 1–127 dB in 1dB steps, 8) OpenIMS Core (IMS Call Session Control Functions (CSCFs) and a lightweight Home Yes (ssh & OMF/FRCP Subscriber Server (HSS), which together form the core elements of all IMS/NGN access) architectures as specified today within 3GPP, 3GPP2, ETSI TISPAN and the Packet Cable initiative. The four components are all based upon Open Source software (e.g. the SIP Express Router (SER)).

4.3 Background and State-of-the-Art on Control and Management of Testbeds

In this section we provide some information on the state-of-the-art tools for testbed management and control, as well as federation setup, that existed prior to FLEX, along with some insights on how these have been extended in order to serve the goals paved by the project. These tools include control tools for the management of the testbeds and federations, experimental plane tools,

for conducting experiments over the testbed, as well as monitoring methodologies, for collecting measurements over the distributed testbed resources.

4.3.1 Slice-based Federation Architecture (SFA)

Slice-based Federation Architecture (SFA) [18] is used in order to facilitate testbed federations, via providing a standardized interface. It provides a minimal interface, which enables testbeds of different technologies and/or belonging to different administrative domains to federate without losing control of their resources.

SFA provides a secure, distributed and scalable narrow waist of functionality for federating heterogeneous testbeds. However, there are barriers to entry to using SFA: a testbed owner would normally need to implement the certificate-based authentication and authorization mechanisms used by SFA, as well as coders and parsers for files that describe the resources on their testbed.

Some examples of well-known tools that take advantage of the SFA architecture are jFed [19], mySlice [20], OMNI [21], used to graphically represent an experiment including resources from multiple sites.

4.3.2 cOntrol and Management Framework (OMF)

The management of several heterogeneous resources is a significant issue for a testbed operator. The testbeds, which are participating in FLEX have adopted the cOntrol and Management Framework (OMF) [22] for the administration and experiment orchestration with the underlying resources. OMF was initially developed in ORBIT by Winlab and currently its development is being led by NICTA along with the contributions of other institutions like Winlab and UTH. FLEX has adopted the “cOntrol and Management Framework (OMF)” for providing experimentation services on top of the FLEX testbeds. The framework allows for the transparent configuration of the underlying resources, via the submission of a simple experiment description in a high level language. The experimenter is able to submit this kind of description to the testbed, and the different OMF components communicate with each other and set up the experiment topology.

Currently, two different releases of the OMF framework are supported: OMF5.4 and OMF6. OMF version 6 has introduced radical changes in the architecture and philosophy of the framework. The main concept of the new architecture is that everything is being treated as a resource and for every

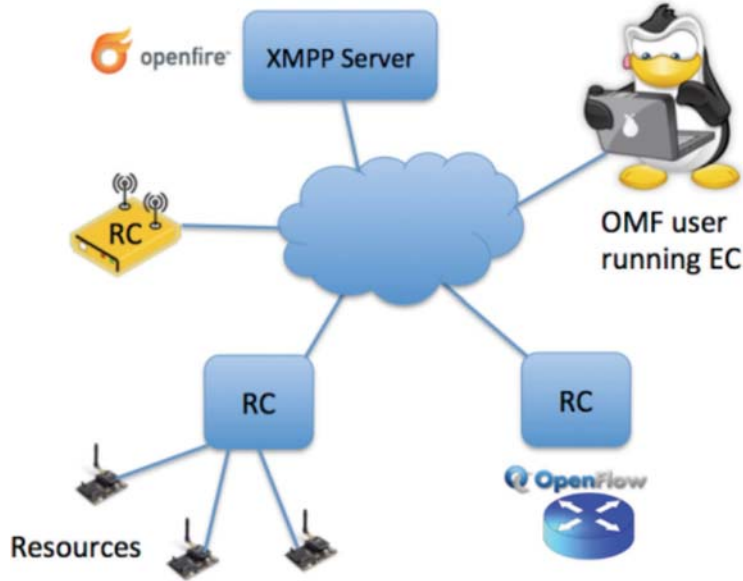


Figure 4.2 The OMF6 architecture.

resource there is a dedicated resource controller (RC) responsible for controlling it. OMF 6 moves towards to an architecture, which incorporates loosely connected entities, that communicate with a “publish-subscribe” mechanism by exchanging messages that have been standardized (Figure 4.2).

In overall, OMF 6 aims to define the communication protocol between all the entities rather than their specific implementation.

The messages of this communication protocol that are being exchanged are defined in the federated resource control protocol (FRCP [23]). This novel protocol defines the syntax of the messages, but not the semantics that are subject to the different implementations concerning the various kinds of resources (see Figure 4.2).

On the other hand, version 5.4 of the OMF framework is the most mature of the frameworks released under the 5th release. It supports interoperability with legacy OMF components. Although the exchange of messages is not standardized like in the 6th version, the testbed administrator is able to define a sequence of messages along the components and handle them appropriately. The different building blocks of OMF are the following, as shown in Figure 4.3:

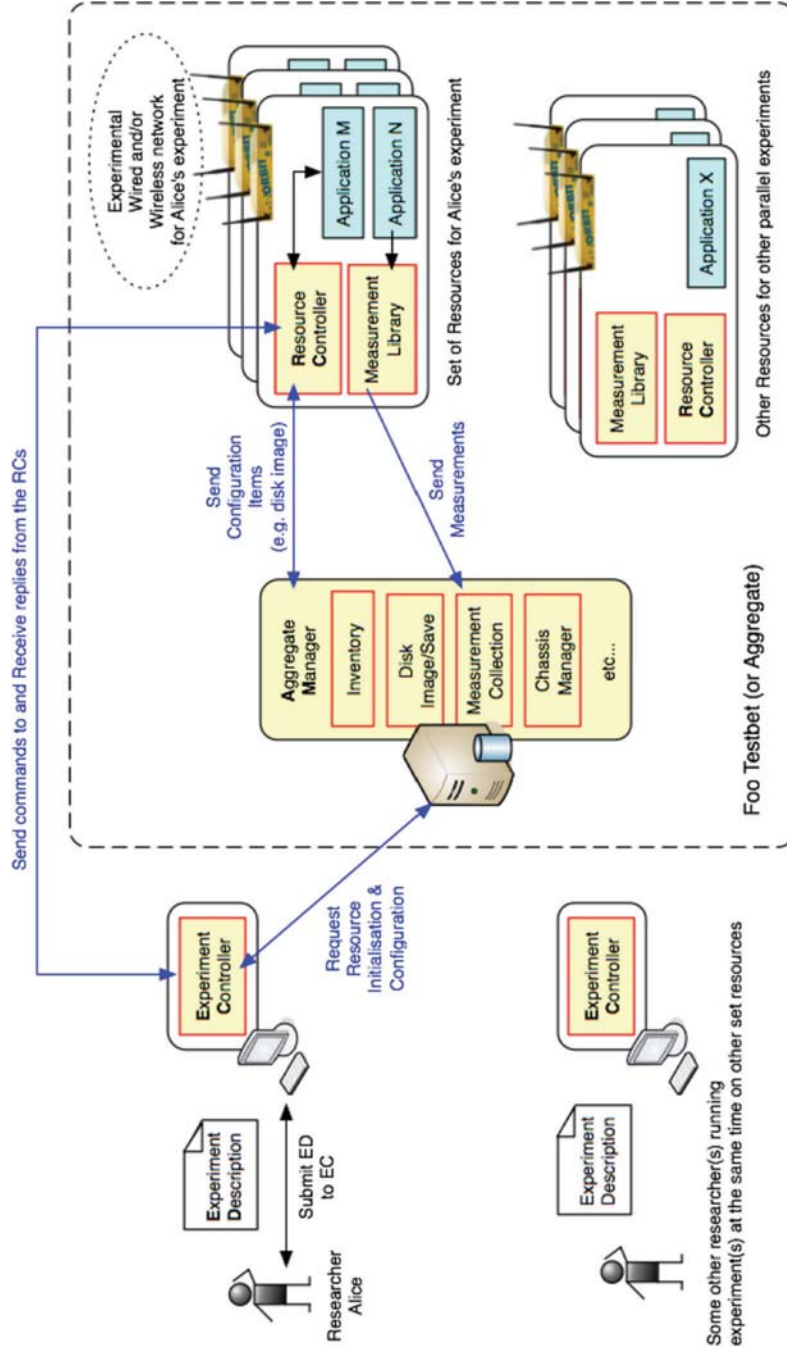


Figure 4.3 The OMF-5.4 architecture.

1. The OMF Experiment Controller (EC): The EC is in charge of receiving the experiment description in a high level language named OMF Experiment Description Language (OEDL) and generating the appropriate OMF messages sent to the Resource Controller.
2. The OMF Resource Controller (RC): The RC is in charge of parsing the OMF messages created by the EC and translating them in the appropriate commands for configuring the resources, installing/starting specific applications etc. The RC is generating OMF messages for monitoring the experiment process.
3. The OMF Aggregate Manager (AM): The AM is providing administration services for the testbed, like for example loading/saving an image on a node, turning a node on/off, etc.

4.3.3 OML

OMF Measurement Library (OML) [24] is acting complementary to the OMF framework and can be used for collecting distributed measurements from new or existing applications (Figure 4.4). Although initially it was developed to support the OMF framework, currently it can be used as a stand-alone library. OML is now a generic software framework for measurement collection.

OML is quite flexible and can be used to collect data from any source, such as statistics about network traffic flows, CPU and memory usage, input from sensors such as temperature sensors, or GPS location measurement devices. It is a generic framework that can be adapted to many different uses.

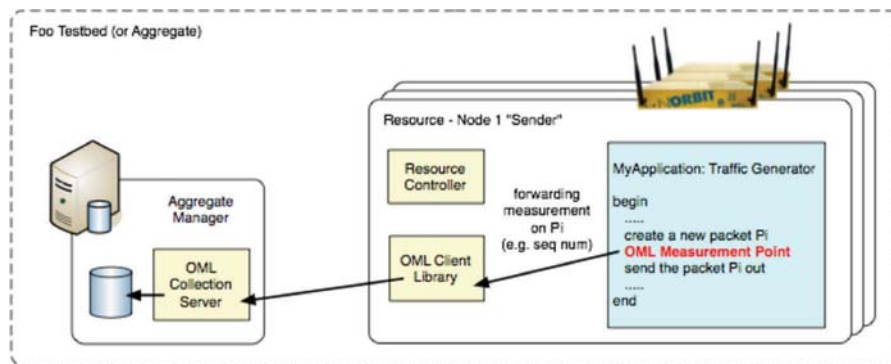


Figure 4.4 OML measurement library architecture.

Networking researchers who use testbed networks to run experiments would be particularly interested in OML as a way to collect data from their experiments.

OML consists of two main components:

- **OML client library:** the OML client library provides a C API for applications to collect measurements that they produce. The library includes a dynamically configurable filtering mechanism that can perform some processing on each measurement stream before it is forwarded to the OML Server. The C library, as well as the native implementations for Python (OML4Py) and Ruby (OML4R) are maintained.
- **OML Server:** the OML server component is responsible for collecting and storing measurements inside a database. Currently, SQLite3 and PostgreSQL are supported as database back-ends.

4.4 Approach

In order to enable the experimentation potential of the distributed FLEX platform, the resources offered by the consortium needed to be fully aligned with the testbed tools and frameworks. To this aim, FLEX has built extensions based on the aforementioned frameworks, as well as new platforms completely from scratch, in order to facilitate the experimenter access and usage of the LTE resources. The extensions and tools that FLEX has built are summarized in the following principles:

1. **Extensions for handling the LTE resources and SFA based federation:** These include the definition of new Resource Specifications (RSpecs) for the LTE network components that are present in each facility. Moreover, the integration of these RSpecs and handling of the equipment by higher layer tools, such as jFed, NITOS broker [25] and Emulab [26] are included.
2. **Tools for facilitating experimentation with the FLEX resources:** These tools include the development of a completely new service, able to handle parameters from the base stations and core networks, and provide a standardized API to experimenters. This service is built from scratch during FLEX and named LTERf. Moreover, the tools in this section include the definition of new OMF controllers for handling the LTE equipment.
3. **Monitoring applications of the LTE network status:** Monitoring applications have been developed by COSMOTE, the largest mobile

operator in Greece, along with UTH. The applications are aiming at both the depiction of network related information (e.g. Cell-Id, RSSI/RSRP, LAC/TAC) and the identification of possible network issues (e.g. poor/no coverage, unsuccessful handover). The tools are designed so as to fulfill the commercial requirements both in terms of presentation and functionalities. The tools developed are utilized in the context of FLEX project during the project time course by the project partners as well as by COSMOTE's engineering staff, mainly.

4. **A toolkit for enabling handover experimentation over FLEX:** As handover experimentation is of major importance for next generation and 5G technologies, FLEX members have developed a rich toolkit for enabling user-friendly experimentation and definition of handover experiments. The handover experiments that are currently supported include S1- and X2-based for LTE, as well as an SDN based handover scheme for cross-technology based handovers (e.g. LTE to Wi-Fi/WiMAX/Ethernet).
5. **Mobility emulation and real-mobility framework:** FLEX is providing sites offering real mobility, through either predefined trajectory control (iMinds) or fully uncontrolled mobility (UTH) inside the coverage area of a macrocell setup. Using the information collected through these real-world setups, including the signal fading for the different wireless channels, etc., FLEX is able to provision an emulation mobility platform using the programmable attenuation platforms for the LTE network. Through this framework, mobility patterns are used as predefined patterns, which can be programmed in the emulators by the experimenters.
6. **Functional federation of the testbeds:** This principle includes the operational engagement of the extensions for the control and experimental plane tools, as well as the physical interconnection of the testbeds over the GÉANT network in Europe. Using the extensions for the federation, resources from different testbeds inside FLEX are able to be bundled in one single experiment description, including scenarios of cross-platform interoperability (e.g. OAI femtocells and commercial macrocells from NITOS in Greece, controlled by an EPC network setup in Eurecom testbed in France).

The following section is describing in detail the extensions that FLEX has built in order to provision truly open LTE and beyond resources to the research community.

4.5 Technical Work

4.5.1 Control Plane Tools

The control plane tools that FLEX has focused are the ones that existed in the FIRE community before FLEX. The extensions to these tools are summarized in the following list:

- Extensions to the NITOS Scheduler – Portal platform
- Extensions to jFed
- Extensions to the NITOS Brokering tool

4.5.1.1 NITOS Scheduler

The NITOS Scheduler [27] is a framework developed by UTH, dedicated to the control and provisioning of testbed resources. It is developed in the spirit of serving as many users as possible without any complicated procedures. Its functionality relies on the OMF architecture. NITOS resources, namely nodes and wireless channels, are associated with the corresponding slice during the reserved time slots, in order to enable the user of the slice to execute an experiment. UTH has enabled Wi-Fi spectrum slicing support in NITOS, meaning that various users may use the testbed at the same time, without interfering with each other, since each one of them is using different spectrum blocks. The service can be adopted with very minor changes from any NITOS like testbed. It is worth to mention that already the Eurecom FLEX site is operating by adopting the NITOS Scheduler platform. It consists of a web frontend and a database backend for selecting and applying the appropriate firewall rules (for accessing the resources) and the spectrum restrictions (for not colliding with other experiments). In order to incorporate the FLEX resources, NITOS Scheduler has been extended in order to be able to parse the RSpecs regarding the LTE resources. Moreover, the web-frontend has been extended allowing the advanced filtering of the testbed resources, based on their type and frequency of operation.

4.5.1.2 jFed

jFed [19] is a framework that allows a user to design an experiment using resources of any of the Fed4FIRE's resource pool. It makes it possible to learn the SFA architecture and related APIs, and also to easily develop java based client tools for testbed federation. jFed is built around a low level library that implements the client side of all the supported APIs. A high level library manages and keeps track of the lifecycle of an experiment. On top of these two libraries various components have been built with different useful functionalities. The most important are:

- jFed Experimenter GUI (Graphical User Interface) and CLI (Command Line Interface) that allow experimenters to provision and manage their experiments.
- jFed Probe GUI and CLI that assist testbed developers to test their API implementations.
- jFed Automated tester GUI and CLI that perform extensive automated tests of the different testbed APIs.

The jFed framework that is used in FLEX has been extended to support LTE experimentation. Hence an experimenter can design his/her experiment and use the available LTE equipment. The equipment includes resources that are filtered through their defined RSpecs, regarding either base stations, EPCs or UEs. Moreover, the experimenter can alter the parameters that are used for setting up their experiment (e.g. transmission power, IP address of MME and PGW, etc.).

4.5.1.3 NITOS brokering

Fed4FIRE [5] project has been working towards federating experimental facilities using one unified framework. The Broker entity, which is designed by the Fed4FIRE project and implemented by the two partners who are also participating in the FLEX project (UTH and NICTA) is offering the means for resource discovery, reservation and provisioning of federated infrastructure to the testbed users. Broker's responsibilities contain the advertisement of testbed's resources to the interested users, but also the reservation and provision of them. It is a way to easily federate OMF testbeds under the scope of SFA [18]. However, it is not limited serving the SFA specification with the XML-RPC interface. Broker should be seen as the main way for experimenters to interact with an experimental facility. It offers additional interfaces beyond XML-RPC, like RESTful and XMPP which leverages the new OMF Messaging System. The main functions of the Broker are communication (through the Broker's available interfaces), Authentication/Authorization, Scheduling and AM Liaison.

The brokering service adopted by NITOS-like testbeds has been developed over the OMF6 framework and support the following configurations towards allowing the efficient provisioning of the project's testbeds:

- Discovery of the available LTE equipment in each testbed (base stations, EPCs and UEs).
- Configuration of this equipment tailored to each experimenter's needs (e.g. using a NITOS base station with a 3rd party EPC network using only the Internet connection).

- Intercommunication among the the testbeds for the resource reservation.
- Setting up the proper user accounts for accessing the LTE components.
- Configuring the appropriate access rules on each testbed for isolating concurrent experiments among different users.

The broker entity is interfacing the scheduler of each testbed and based on the resources creates the appropriate RSpecs for advertising the testbed components. It is also featuring multiple APIs for interfacing the SFA API that it provides. The supported APIs are three; 1) an SFA client based, using for example applications like SFI [28], 2) a REST based and 3) an FRCP [23] based.

4.5.2 Experimental Plane Tools

The extensions that are described in this section regard the following:

- The definition of the LTERf [29] service, for handling all the FLEX component parameters and easing the testbed federation, by allocating end-to-end isolated paths.
- The extensions to the core OMF framework for supporting experimentation with the LTE resources.

4.5.2.1 The FLEX LTERf service

One of the main challenges in provisioning an Open LTE testbed is the provided API for the configuration and setup of the involved LTE components. The LTE components we refer to are the base stations, EPC network, monitoring and datapath functions. In the following sections we refer to the “LTERf” [29] service that has been developed through the FLEX project, aiming for providing open and configurable APIs to the experimenters that take advantage of the FLEX testbeds.

The service is built on top of the OMF AM entity and provides a REST based interface for interacting with it. It is configured to reply with either an XML format or plain text, depending on the query and the representation that is requested by the end users. The APIs that are provided to the users are abstractly divided to four classes:

- *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/GÉANT). For the cases of the GÉANT 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.

The service has a modular architecture as shown in Figure 4.5. The different northbound interfaces for the subservices are mapped to resource specific drivers for controlling and configuring the diverse components. These drivers

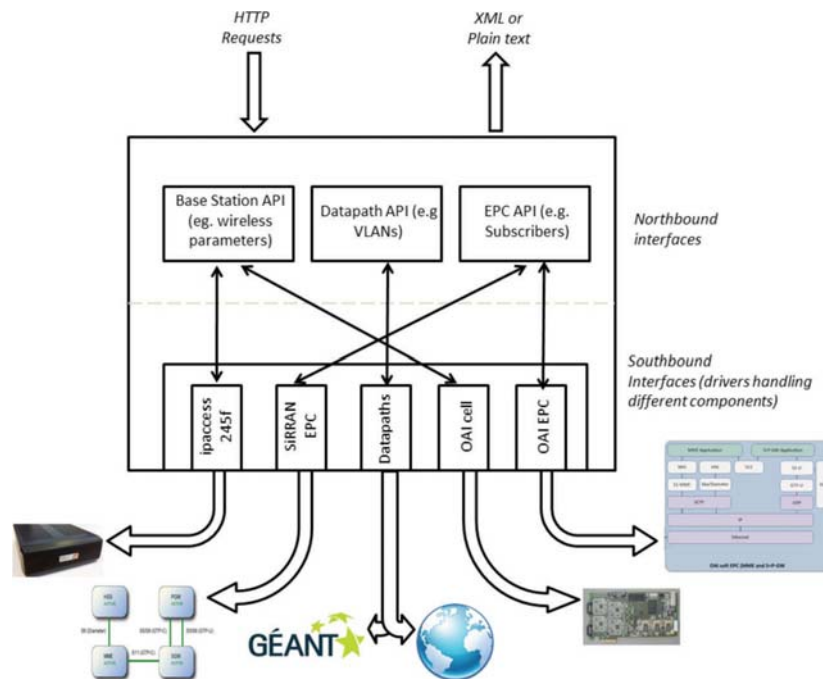


Figure 4.5 The LTERf service architecture; single northbound interfaces are mapped to several southbound depending on the type of the equipment.

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 startup, a configuration phase is employed where the available resources (specified in a configuration file) are given to the service.

Different modules on the southbound interface are used to configure the different components 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.

The existing cellular solutions that are currently supported by the LTERf service are the following: 1) ip.access femtocells, 2) OpenAirInterface cells, 3) Airspan Air4GS LTE macrocells, 4) OpenBTS components, for configuring 2G/3G circuit-switched networks along with the 4G and beyond ones, 5) the Keysight T2010 conformance testing” units, 6) The SiRRAN EPC instances, 7) OpenEPC instances and 8) OpenAirInterface EPCs.

4.5.2.2 OMF extensions

As OMF has been widely deployed worldwide, FLEX has extended the available OEDL language for specifying experimental resources in order to include LTE resources as well. The LTE resources that are currently supported by incorporating them in an OMF experiment are:

1. LTE USB dongles, for connecting testbed nodes to the provisioned LTE networks,
2. LTE Android enabled Smartphones, connected to the FLEX networks and controlled over the Android Debug Bridge (ADB),
3. UE instances of the OAI platform.

These resources are currently supported by the FLEX platforms, by means of the respective OMF Experiment Description Language (OEDL) extensions, extended EC's for controlling the LTE equipment and brand new RCs (for both OMF versions).

The syntax is supporting configuring the LTE dongle to operate as a modem/USB mass storage device, restarting it, turning on/off the radio, setting an APN that will be activated for setting up the required PDP context, attaching and connecting to the network and using a defined IP address.

The OMF ECs (both for OM6 and OMF5.4) have been extended in order to support the updated experiment syntax and the generation of the OMF

messages that are sent to the respective OMF RCs. For the case of the OAI UE, the same API is used as in the case of the LTE dongles, yet the vast configurability of the platform is allowing for the further extension of it in order to support more configuration parameters.

The RCs are responsible for receiving and decoding the OMF messages (FRCP for the case of OMF6) and translating them to the appropriate commands. For the case of the LTE dongles, the diversity of the available dongles inside the FLEX federation is posing several barriers that have to be overcome by the RCs. To this aim, the RCs are using the standardized protocol of AT commands [30] for interacting with the LTE dongles. The RCs for the smartphone components have been developed in the same spirit the respective ones for the LTE dongles.

Regarding the smartphone control, two RCs have been developed; an OMF5.4 RC for controlling the smartphone over the Android Debug Bridge (ADB) and an OMF6 RC for controlling it over the Wi-Fi interface. For the case of the ADB, the smartphones are connected in the NITOS testbed to the lightweight Raspberry-Pi based nodes that UTH has developed, or to standard NITOS nodes, via the USB connection.

4.5.3 Monitoring Applications

COSMOTE and UTH have developed over the FLEX platform three mobility/performance-related tools (Figure 4.6). The tools are decomposed to:

- (a) Client applications running on Android devices, in “on-demand” mode, “on-event” mode or “periodically”.
- (b) Server-side infrastructure utilized to collect, store and process the related mobility/performance measurements.
- (c) A graphical environment (WebGUIs) with advanced filtering and presentation capabilities, through which the measurements will be depicted.

4.5.3.1 FLEX QoE tool

The purpose of this tool is to present 2G/3G/HSPA/HSPA+/4G network related information (including BSs locations/capabilities/name, cell reselections/locations info, handover locations/info, etc.) in real time, over Google Maps. It is also able to measure and depict QoE related measurements in real time, such as signal strength (RSSI, RSRP, RSSNR, RSRQ, etc.), latency, maximum download bitrate, maximum upload bitrate and upload the QoE related measurements to a dedicated server for storage, post processing. The collected measurements are depicted via a user friendly web interface.

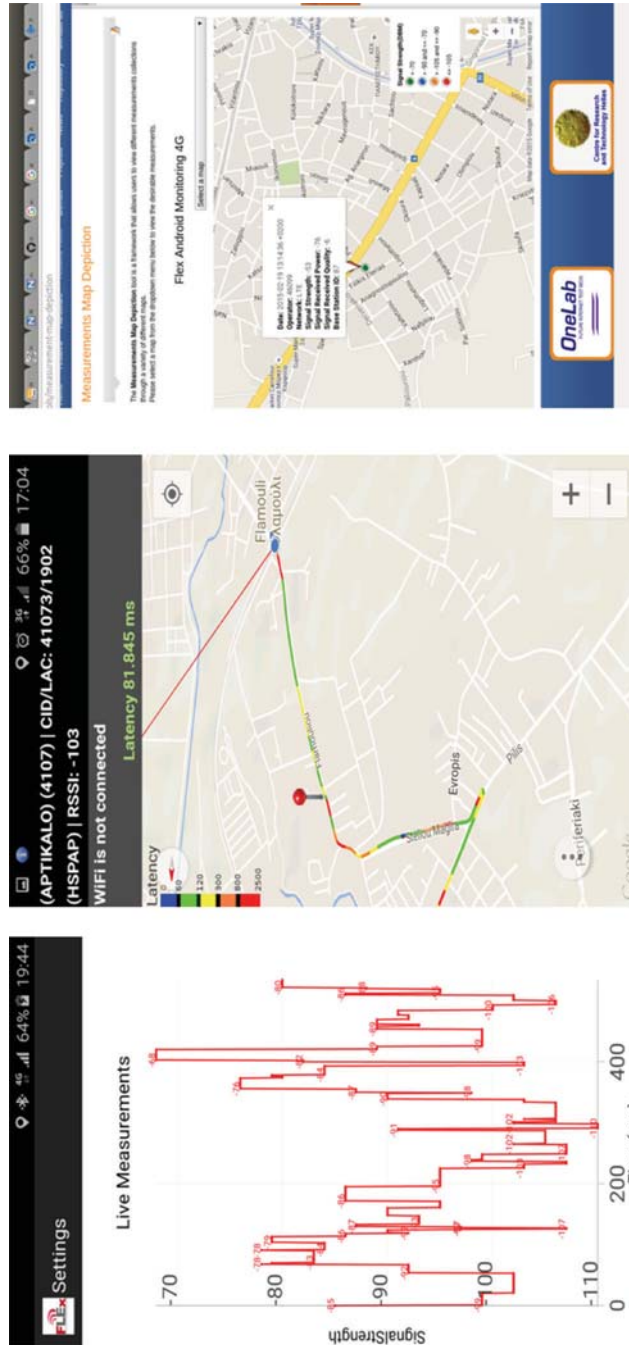


Figure 4.6 Mobile operator android tools for monitoring in FLEX.

4.5.3.2 FLEX_problems

The aim of the FLEX_problems tool, is to notify the MNO, in real-time, on network issues/problems (e.g., areas exhibiting huge number of cell reselections, poor coverage, no coverage, high number of handover failures). The client application runs on Android devices and could start either at power on, or manually. The application could be: (1) utilized by MNO staff (mobile UI is required in this case) and/or (2) offered by a MNO as a commercial application (running in background – no mobile UI required). In either case a graphical environment (WebGUI) shall be made available to the MNO so as to be informed on those network events. More specifically, the basic features of the FLEX_problems client App are the following: 1) Presents at terminal screen 2G/3G/4G network-related info (BS name, BS-id, RAT, cell-id, LAT/TAC, RSSI/RSRP, RSRQ, etc.), 2) “Listens” to the environment (2G/3G/4G) continuously and the terminal status (offhook, busy), 3) In case of an event (cell change on idle, handover, low-RSSI) it uploads, in real-time, to a dedicated server, the relevant measurements. 4) If the network is not available (handover failure, no coverage), it queues the “measurements” and uploads them (automatically) upon “network recovery”, 5) Presents at terminal screen info, in real-time, regarding the number of cell reselections, handovers, poor coverage location identified, along with the number of queued messages (if any).

4.5.3.3 FLEX_netchanges

The aim of this application is to (automatically) measure the network performance in terms of signal strength (RSSI, RSRP, RSSNR, RSRQ, CQI), latency, maximum download bitrate, maximum upload bitrate) periodically (e.g., every X minutes). The application could be: (1) deployed by an MNO, on its own terminals distributed at specific locations – terminal operation could be remotely controlled and/or (2) offered by the MNO as a commercial app (running in background – no mobile UI required in this case). This application can serve as “real-time” network probes, in order the MNO to be notified on network performance e.g. in cases of Self-Organized Networks, network changes, etc.

4.5.4 Handover Toolkit

The handover toolkit available across the FLEX testbeds is an open framework that allows the configuration of the handover parameters for facilitating this type of experimentation. The following setups are supported:

- S1-based handover, using the commercial FLEX equipment.
- X2-based handovers, using the OpenAirInterface equipment.
- Cross-technology handover frameworks, using SDN and any types of LTE equipment.

4.5.4.1 S1-based handovers

In accordance with the FLEX project requirements to support experimentation of handover scenarios, SiRRAN and ip.access have extended the capabilities of their equipment (femtocells and EPC) to include S1 based handovers, between eNodeBs, connected to a single MME. Although S1 handover is normally utilised to facilitate transfer between eNodeBs that are connected to different MMEs, the NITOS and w-ilab.t testbed installations of the SiRRAN EPC use only a single MME component, so the functionality was designed in the EPC with this in mind. Initial development and testing was performed in SiRRAN's labs, using ip.access LTE245 and E40 radios.

4.5.4.2 X2-based handovers

Regarding the setup of the X2-handovers using the OpenAirInterface platform [31], within FLEX the extensions to support this type of handover procedure has been developed. X2 handover has several advantages compared to the conventional S1/MME handover used by other FLEX testbeds. The main key-features are described below:

1. The whole procedure is performed directly by the eNBs (without EPC). There is a direct tunnel formed between source and target eNBs for downlink data forwarding in handover execution time.
2. MME is involved only when the handover procedure is completed in order to setup the new network path.
3. The UE release context at the source eNB side is triggered directly by target eNB.

Thus, X2 handover minimizes the latency of the EPS network. A handover experiment in OAI can be performed using a different set of parameters that are managed via configuration/command-line (User CLI) inputs. User CLI provides certain commands for runtime control and monitoring of the OAI X2 handover. The parameters that can be adjusted are time to trigger, hysteresis parameter for this event, the frequency specific offset of the frequency of the neighbour cell, the cell specific offset of the neighbour cell, the frequency specific offset of the serving frequency, the cell specific offset of the serving cell, the offset parameter for this event, coefficient RSRP/RSRQ,

parameter for exponential moving average (EMA) filter for smoothing any abrupt measurements variations. The developments take place over the OAI networking stack, thus enabling for the further extension and development of new policies for handover (e.g. [32]).

4.5.4.3 Cross-technology Inter-RAT SDN based handovers

Regarding the cross-technology inter-Radio Access Technology (RAT) handover framework, it is based on the OpenFlow technology [33], able to perform seamless handovers among different technologies (e.g. Wi-Fi to LTE, LTE to Ethernet, Wi-Fi to Bluetooth, LTE to WiMAX, etc.). The architecture adopted for the realization of the framework in NITOS is depicted in the Figure 4.7.

The framework is called **OpenFlow Handoff Control (OHC)** [34] and is consisting of two different entities; the mobile clients and the destination servers. During a handoff, network address changes take place at the mobile host, which break the established connections if no proper management is applied. These changes are induced by the different gateway used by each RAN, or by the NAT process that is always present before the traffic is routed to the Internet. With the OHC scheme the changes are handled at two points; on the client that performs the handoff and just before the traffic reaching the destination server. By using the OpenFlow technology, we are able to establish custom flows on a network switch, by mangling the exchanged traffic accordingly so as the connections are not dropped.

The key for applying our scheme relies on creating virtual OpenFlow enabled switches. To this aim, on the mobile node we employ the architecture illustrated in Figure 4.7; we place all the available networking interfaces in a single switch. By relying on the Open vSwitch framework [35] for the creation of our switches, the switches residing on the mobile node are OpenFlow enabled. The Operating System on the mobile node communicates only with the bridge device as a network interface and uses it as the default interface for any outgoing/incoming traffic from the mobile node. The controller that is establishing the flows on this virtual switch is in charge of selecting the appropriate southbound interface (e.g. Wi-Fi, LTE) for sending out the traffic.

The respective changes for adopting our framework have to take place before the traffic is delivered to the destination application. As we described, in the case that the bridge on the mobile node has an IP address of the 10.0.0.0/24 subnet while the Wi-Fi interface bears an IP address of the 192.168.0.0/24 subnet, the flow on the switch will change the source IP and MAC address

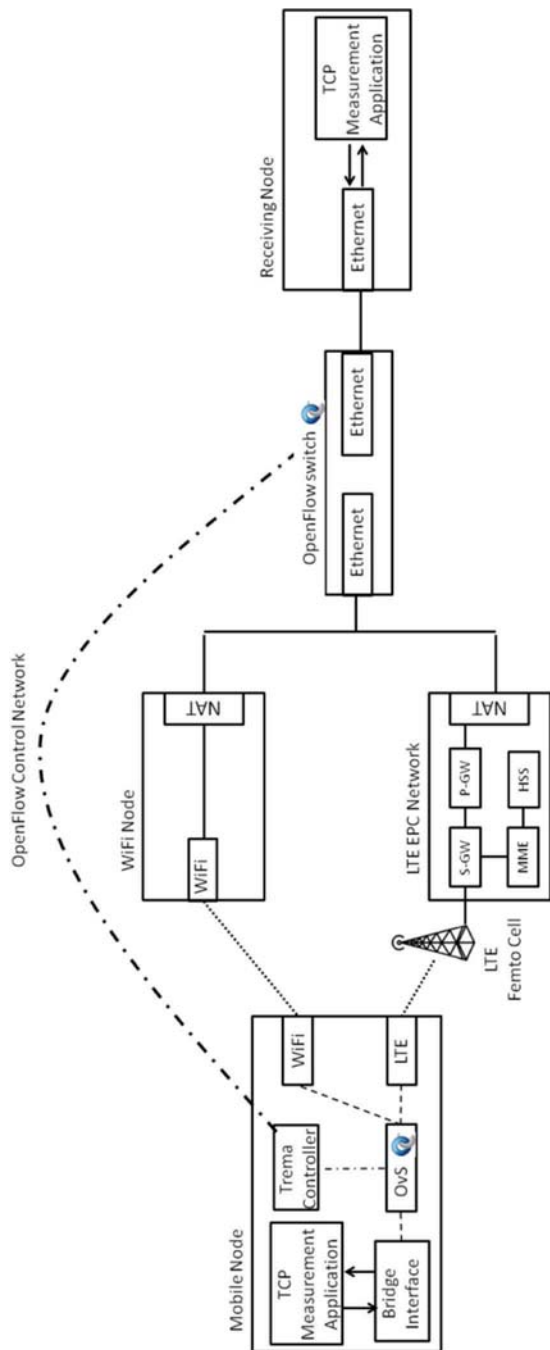


Figure 4.7 The Inter-RAT cross-technology handover framework.

of each outgoing packet to match the address of the Wi-Fi interface, and the respective destination MAC address to match the one of the target Wi-Fi Access Point. For the incoming packets, the opposite procedure has to take place.

The testbed application of our framework is the following. On our mobile node we use Open vSwitch (OvS) for our bridging solution, and enable its control from an OpenFlow controller residing on the same machine. We employed the Trema framework [36] as our solution for implementing our OpenFlow controller. Finally, we unified both the operation of our aforementioned algorithms (server side and mobile node side) in one controller instance, which is able to control multiple datapaths (mobile node and NITOS OpenFlow switch).

A comparison of the FLEX inter-RAT framework for LTE to Wi-Fi handovers against other state-of-the-art solutions for cross-technology handovers or higher-layer solutions is shown in Figure 4.8. As it is illustrated, both achieved throughput and delay through this scheme are better, compared to other technologies, and as if the interfaces were acting as standalone connections to their network.

4.5.5 Mobility Emulation Platforms

Data captured from the real network setup are used in order to feed the mobility emulation platforms. The data that is used for generating the patterns is collected from monitoring applications, residing at the FLEX testbed nodes, and after their anonymization (removing all the user sensitive information, such as the phone's IMEI, the card's IMSI, etc.) are fed to the emulation

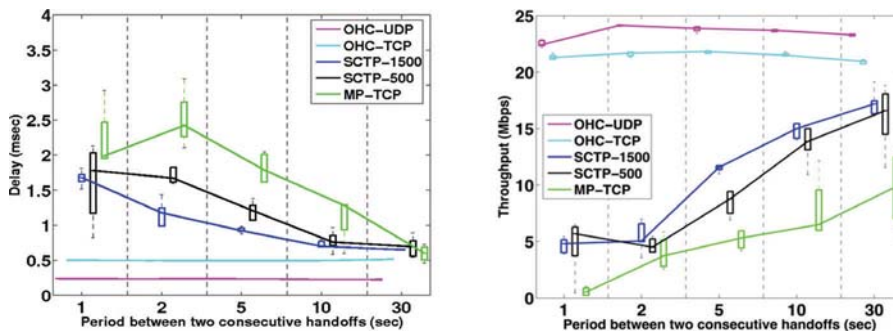


Figure 4.8 OHC comparison against other technologies for seamless handovers.

platform. The selection of a tool like the Qosmotec platform (by iMinds) is crucial, as it provides the experimenters with the potential to fully replicate a real world mobility experiment with the emulation platform.

Path loss models can be used to calculate the reduction in power density of the signal between two radio devices. The results of path loss model calculations can be used to feed the emulation mobility platforms (attenuators, LTE cells and UEs) and emulate signal attenuation. The simplest path loss model is the free-space path loss (FSPL) model that presents the loss in signal strength on a line-of-sight path without any obstacles [37]. The calculations are straightforward but do not model real conditions. For cellular networks, the Walfisch-Ikegami (COST231 project) [38] and Erceg model [39] are frequently used. The ITU-R P.1238 model [40] is developed for indoor conditions. Most of the models are used for lower frequencies (<2 GHz), but by adding a certain correction factor, they can still be used for higher frequencies.

4.5.6 Functional Federation

In order to enable the functional federation across the FLEX islands, dedicated end-to-end slices have to be reserved from one testbed to another, utilizing the GÉANT network. The tools that enable such access are the the LTERf service and jFed. LTERf has been developed in a manner that allows user defined datapath control. However, the incorporation of LTE resources in the testbed network creates several issues that are not present when dealing with other resources than the LTE ones. 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 the UE. As the address with which the EPC replies to any ARP request destined to the 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. Similar to this, the service is supporting the VLAN creation through an HTTP command, and adding it to the datapath so that the experimenter can create end-to-end isolated slices of the infrastructure, incorporating different components from different testbeds with guaranteed bit rates. Since the GÉANT connections are delivered as a VLAN interface at the testbeds, the service enables the creation of dedicated QinQ VLANs inside them, per each user request.

The jFed provisioning of end-to-end slices is based on VLANs which are provisioned and then stitched together at points where they meet. The workflow in the jFed tool is as follows:

- The experimenter draws in an experimenter tool a link between two nodes on different testbeds (which is translated in an RSpec).
- When the tool starts provisioning, it first calls the Stitching Computation Service (SCS) which calculates a route between the two testbeds based on the layer 2 paths it knows. The SCS augments the RSpec with this information.
- The tool then knows also intermediate hops in the path (e.g., GÉANT, Internet2) and can call them to set up the path.
- In the end, all the parts of the links and nodes become ready, and the experiment is ready.

For this fully automatic stitching, the VLAN numbers are dynamically chosen based on free VLAN overviews, tries and retries.

4.6 Results and/or Achievements

The experimentation potential that the FLEX platform is fulfilling is mirrored in the different number of use cases and scenarios that can be executed over the testbed. Indicatively, we present some experiments that have been successfully executed over the FLEX testbed, along with some experimental results. We focus on the following scenarios:

1. Spectrum coordination schemes for LTE in unlicensed bands, using semantics.
2. The development of an offloading framework using the commercial equipment.

4.6.1 Semantic Based Coordination for LTE in Unlicensed Bands

One of the types of different experiments that can be executed over FLEX testbeds deal with Dynamic Spectrum Access (DSA) for heterogeneous technologies, along with their spectrum coordination algorithms. To this aim, several works have been executed demonstrating the coordination of spectrum for different technologies, using either the commercial LTE equipment [41] or the OAI setup [42].

In this subsection, we focus on the LTE and Wi-Fi coexistence in an unlicensed band environment. Wi-Fi and LTE are different RATs designed for

specific purposes at different frequencies. In the cases when they are required to coexist in the same frequency (e.g. LTE in Unlicensed bands) time and space, increased interference is caused to each other along with an overall system degradation because of a lack of inter-technology compatibility.

For LTE-U (LTE in Unlicensed bands) operation, several challenges have to be tackled for the efficient coexistence of LTE and Wi-Fi technologies. The key differences among the two technologies lie in the medium access method; Wi-Fi uses CSMA/CA, a “listen before talk” method in order to access the medium. In case of an unsuccessful transmission, the Wi-Fi device executes an exponential backoff algorithm before accessing the medium again. Contrary to that, and since LTE is designed for use under a licensed band environment, LTE is using OFDMA (Orthogonal Frequency Division Multiple Access). The coexistence of the two different technologies within the same band, can seriously affect the performance of Wi-Fi. Therefore, efficient spectrum management and power control should be employed for accommodating both of these technologies within the same band. In this use case, we focus on the spectrum coordination solution called CoordSS [42], which is using semantics for the coordination between Wi-Fi and LTE.

Figure 4.9 presents a conceptual overview of the CoordSS networking architecture. Three verticals and three horizontals can be identified in the architecture. The following verticals represent different views on top of the same set of foundational concepts:

- Network Environment – represents the “real” world. This includes hardware devices as well as physical phenomena (such as frequencies) along with their properties.

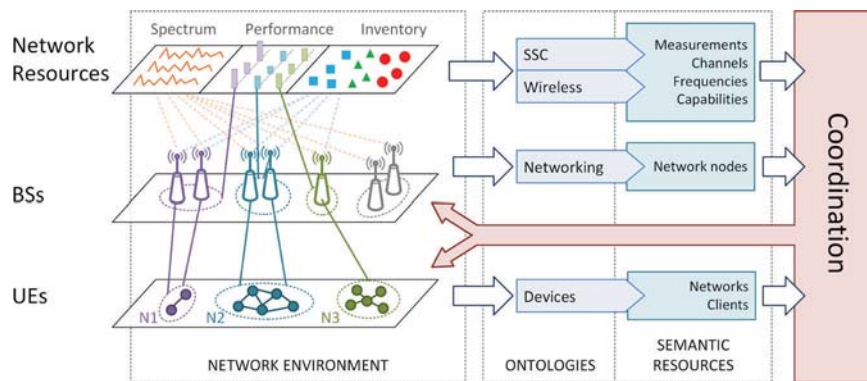


Figure 4.9 CoordSS Network model for semantic based coordination.

- Ontologies – are used to formalize domain specific knowledge that is independent of the context. They contain semantic definitions related to the meaning and purpose of the network environment. Ontologies are created by the domain experts and can be viewed, understood and managed by the humans as well as by the machines.
- Semantic resources – are the results of a semantic annotation of the network environment by mapping between the environment and ontologies. More precisely, if there is a physical resource that can be understood using the given set of ontologies it becomes the semantic resource.

Horizontals represent the main concepts in our network model. In the coordination algorithm, they play the roles of sources and/or destinations.

- Network resources – constitute the state and capabilities of the environment where BSs and UEs are working. They are the primary source of data for reasoning during the coordination. On the networking environment level, we are using spectrum sensing devices (such as Wiser [43]), connection bandwidth monitoring applications (such as *iperf*) and the inventory repository (Note that FLEX testbeds regularly provide such a service). The ontologies level consists of the Spectrum Sensing Capability (SSC) ontology (for describing spectrum sensing) and the Wireless ontology (for describing frequencies, channels and radio bands). And at last, semantic resources level contains data for FFT analysis of frequencies, connection speed, device parameters and their changes over time.
- BSs – nodes that provides access points for UE. They are a backbone for network communication. The OAI [7] ontology is used to describe such devices. The coordination protocol uses a semantic representation of BSs to decide which parameters can be changed to improve networking. Such parameters include their power signals, position (if applicable) and communication channel.
- UEs – client nodes that form networks so they can send and receive data among them. We can have multiple networks, and one UE can belong to any number of networks (but we view it as a separate UE for each network). Therefore, each device is identified by a network name to which it wishes to belong to. Semantic resources for UEs contain client demands for communication.

Coordination is centralized on one machine that is running the CoordSS Coordination server (CCS). The CCS is responsible for running the coordination algorithm, providing client/server communication with the network resources,

mapping network resources to semantic resources, maintaining a semantic store that holds ontologies and semantic resources and executing SPARQL queries. The coordination algorithm is invoked in case the network environment changes, namely when a new BS or UE is introduced or when network resources fluctuate (e.g. changes are observed regarding the performance or spectrum). Clients send their spectrum, performance and node description to the server. This data is in a native format. CCS maps such data to semantic resources and stores them in the semantic store. The semantic store is used for storing and retrieving triplets, basic building blocks of ontologies and semantic resources. SPARQL queries are the standard way for retrieving semantic data, and are used by the coordination algorithm for all reasoning as well.

The main objective of the CoordSS coordination algorithm is to assign radio channels to the networks that are under its control. Any network that participates in our algorithm must have all of its nodes (UEs and BSs) registered to the CCS. Registered nodes send data to the CCS and also receive control messages from it. In our case, only channel allocation control commands are sent, but more elaborated control is also possible. When the algorithm decides to assign a channel to a network, commands are sent to all the nodes belonging to that network to switch to the new channel configuration.

There are two possible scenarios that we consider:

1. (S1) The network is part of the network environment and all of its nodes are aware of the CCS. This network does not have a channel assigned to it, but the coordination algorithm is responsible to provide one.
2. (S2) An uncoordinated network appears in the network environment (LTE or Wi-Fi network). This network uses its own algorithm for channel assignment. This network can interfere with existing coordinated networks. Our algorithm detects such a situation and resolves any interference by re-assign channels of the coordinated networks.

For the experimental evaluation of the proposed algorithm, we employ the NITOS testbed of the FLEX federation. The rich environment that NITOS is offering is utilized in order to configure the suitable environment for the experimental evaluation in real world settings of the CoordSS framework. To this aim, we employ the following testbed components:

- A pair of USRP B210 models, that will serve as the RF front-end of the deployed LTE network.
- Several Wi-Fi enabled nodes, that will be used as the contending traffic in the unlicensed under study bands.

- The OpenAirInterface (OAI) platform, that provides the execution of a 3GPP EUTRAN over commodity hardware, with the appropriate RF front-end. The OAI platform has been extended in order to allow its operation in the unlicensed bands.

The experiment topology is shown in Figure 4.10. The following methodology was used during the experiment. At first, only Wi-Fi stations were involved. Each Wi-Fi network would randomly choose a channel, and the resulting throughput was measured. This procedure was repeated 100 times and the average throughput was calculated. After that, wireless node 1 randomly chose a channel, wireless node 2 received a channel from CoordSS server, and the throughput was measured. The results are shown in Table 4.1. The second part of the experiment, besides the coordinated Wi-Fi networks, involved the LTE eNB, with and without coordination. A similar procedure, was applied. The results are shown in Table 4.2.

The results show the importance of the coordinated spectrum usage. Due to a relatively low number of the involved nodes, the average throughput is not very much improved by the coordination. However, the coordinated network has more stable throughput than the uncoordinated one, i.e. the difference between the lowest and the highest throughput is rather large in uncoordinated network. We should also have in mind that the output power of the USRP B210 is relatively low (10 dBm). Therefore, the influence of the dedicated LTE eNB on Wi-Fi would be much higher.

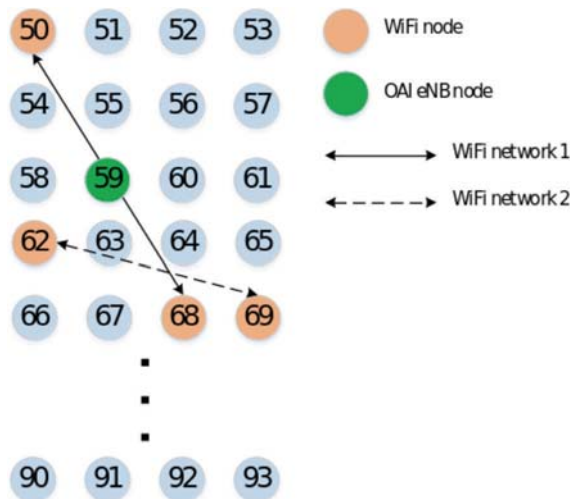


Figure 4.10 CoordSS experimental setup.

Table 4.1 Coordinated and uncoordinated shared spectrum access with Wi-Fi stations

	Wi-Fi Throughput (Mb/s)		
	Min	Average	Max
Uncoordinated	11.5	19.6	22.8
Coordinated	22.8	22.8	22.8

Table 4.2 Shared spectrum access with coordinated Wi-Fi networks and (un)coordinated LTE eNodeBs

	Wi-Fi Throughput (Mb/s)		
	Min	Average	Max
Uncoordinated	10.6	16.7	22.8
Coordinated	22.8	22.8	22.8

4.6.2 FLOW LTE to Wi-Fi Offloading Experiments

As the explosion of Internet and mobile data traffic has placed significant pressure on cellular networks, data offloading to complementary networks (e.g. Wi-Fi) seems to be the most viable solution. For the operators, in contrast to network planning strategies for upgrading, expanding and building up new infrastructure, which means extra capital and operational costs (CAPEX and OPEX), offloading can offer a sufficient and low cost solution for cellular load decongestion. Mobile Data Offloading is also significantly important for the mobile users, who can further benefit from short-range links so as to achieve better performance and experience better quality of communication by shifting to complementary networks. FLOW architecture aspires to address the challenges that offloading brings and create an open and applicable framework for implementing advanced offloading techniques in heterogeneous networks (LTE & Wi-Fi).

The FLOW experiment is realizing LTE to Wi-Fi offloading techniques over the FLEX testbeds (Figure 4.11). The components that have been developed during FLOW have been described in detail in [44]. Nevertheless, we provide a brief description of the components needed for the execution of the offloading framework:

1. **Wi-Fi Access Gateway (WAG):** WAG is serving the role of the the actual gateway of the Wi-Fi mesh network that is used for offloading the LTE clients. Although the implementation of such a device would seem straightforward, in the FLOW framework we differentiate the traffic that is exchanged from the offloaded clients in order to meet some minimum requirements paved by the SLA that they have with the network provider. To this aim, and as we have described, we employ the Linux traffic queues for traffic shaping services.

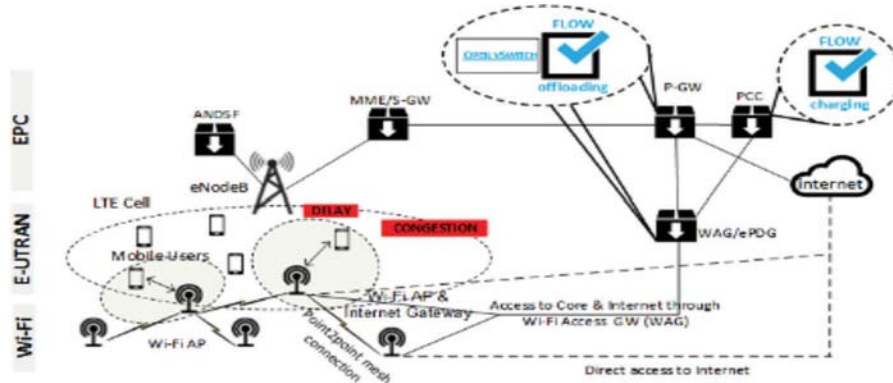


Figure 4.11 FLOW offloading framework.

2. **PDN Gateway (PGW):** The LTE PGW interface is in-charge of terminating the SGi interface towards the PDN. In the case of multiple PDNs, more than one PGW will be available for the UE of the network, depending on the Access Point Names (APNs) used for the network. With FLOW we extend the functionality of the PGW in order to enable the operation of our offloading scheme. We implement an Open-vSwitch [35] bridge that enables the dynamic bridging of two different entities, Wi-Fi mesh network and the LTE network, and attaches the FLOW framework to take care of the low level network functions that have to be employed for the proper operation and routing of packets to the Internet.
3. **FLOW offloading framework:** The FLOW offloading framework has been designed in order to coordinate the interaction among the WAG and PGW elements. By employing a Software Defined Networking manner, we bridge the heterogeneous RANs and through a controller service we are able to select the respective RAN from the network provider's perspective. The policies that we implement for the offloading process are based on the load that each femtocell can provide and some predefined SLAs that each client has contracted with the provider. Moreover, based on the QCI parameters per UE in LTE, we allocate each of the offloaded clients to the respective traffic queue, upon which we schedule the transmissions of the respective data to the WAG and then the rest of the Wi-Fi mesh network.
4. **PCC (Policy Control & Charging):** The PCC unit is in charge of applying the proper control of the policies and charging of the clients per subscriber basis, and based on the QoS class that they belong. As FLEX

components do not include a PCRF component, we have implemented it over the FLOW network to allow monitoring of each client. We are able to both monitor the data that a UE exchanges over the LTE or the Wi-Fi network, and is relying and interacting with the aforementioned schemes (FLOW, PGW, WAG).

For the setup of the FLOW offloading experiment, we employ the IEEE 802.11s extensions that are available in the NITOS testbed images. They are used for forming a multi-hop Wi-Fi mesh network for offloading the LTE clients. As in the NITOS indoor testbed all the nodes are able to “see” each other, we isolate the access nodes by adding specific next hop neighbours in order for the traffic that we send to use at least 2-hop paths before reaching the WAG gateway. The WAG component is also located in the NITOS testbed and is connected via a 1 Gbps connection to the EPC server that we use.

Regarding the WAG configuration, we use a tap-based tunnel for the communication of the EPC and the WAG components (Figure 4.12). We choose this type of connection as the PDN-GW is also represented a tap interface. On the node that is playing the WAG role, we use Open-vSwitch on the node in order to bridge the two interfaces (tap and Wi-Fi mesh). Based on a predefined set of IP addresses that we use for the Wi-Fi clients, sharing the same IP range with the LTE ones, we allocate them to a different traffic queue inside Open-vSwitch. Using external applications, such as the “tc” [45] traffic control service, we are able to throttle appropriately the traffic that is delivered to each client, based on the delivery IP address of each client. For the application of the different QoS profiles that each UE is using, we utilize the functionality that SiRRAN’s LTE net is offering, allowing us to setup different subscriber groups with multiple subscribers. Based on this configuration and groups, the EPC is able to throttle the traffic either on the DL or on the UL

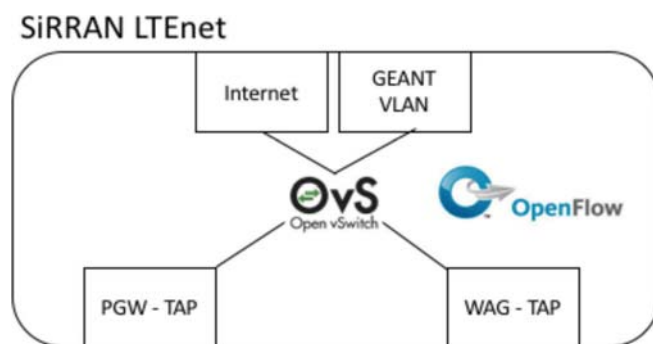


Figure 4.12 FLOW PGW extensions for FLEX.

that they exchange over the EPC network. This already supported functionality alleviates the employment of similar traffic throttling solutions for the LTE network, contrary to what happens for the Wi-Fi mesh network.

As within LTeNet the traffic that is delivered to the PGW interface is represented as an Ethernet tap interface, we used an altered version of the default “GÉANT” datapath that is available in NITOS as our starting point. This “GÉANT” datapath [29] is enabling the bridging of the PGW interface (that is reflecting an APN inside the network) and the GÉANT VLAN termination point in NITOS. The architecture that we have employed is depicted in Figure 4.13.

The cornerstone of the FLOW offloading management framework relies on the operation of the controller managing and establishing flows on the Open-vSwitch bridge on the LTeNet installation. For our initial tests and the experimental evaluation of the offloading frameworks, we developed a framework based on some predefined SLAs for all the involved clients.

The FLOW controller is in charge of the following actions:

1. Based on the first packet that it receives from the LTE client, checks whether the client’s SLA can be met from the current capacity and bearer allocation at the LTE network.
2. Decides whether to offload the client or not.
3. In case that the client will not be offloaded, the controller establishes the proper flows that allow the communication of a client from the PGW interface to the Internet or the GÉANT network.
4. If the client will get offloaded the following actions are triggered:
 - a. The controller triggers an assisting FLOW application running at the EPC which communicates the offloading message via the testbed control network to the UE. Another similar application that is installed on the testbed node, handles the message and instructs the wireless network interface to connect to the Wi-Fi mesh network. From now on, the offloaded UE will use the Wi-Fi network as the default gateway for sending traffic.
 - b. The controller is communicating a similar message to the WAG component. The WAG, based on the SLA for network capacity, allocates the node on the proper HTB queue of the system, thus scheduling appropriately and shaping the DL traffic that the client will receive over the Wi-Fi network. Finally, the controller establishes the appropriate flows on the Open-vSwitch bridge of the EPC network to use the WAG-tap interface as the default interface for the specific UE.

5. Continues monitoring the environment conditions, through similar messages received from the Wi-Fi mesh. In case that a client has left the LTE network, and the SLA of an offloaded client can be met from the LTE network, it reinstructs the client to connect to the LTE network, following a similar procedure like the one described in step 2.
6. Monitors the traffic load that each client has sent over the WAG/PGW interface in order to apply the pricing and charging functions.

The overall architecture that we adopted for an initial setup at the NITOS testbed is depicted in Figure 4.13. The setup at this point has been mapped over the NITOS testbed.

For the evaluation of the FLOW experiment, we performed offloading based on some pre-defined SLAs for the LTE network. The SLAs that we used for each LTE node are summarized in Table 4.3.

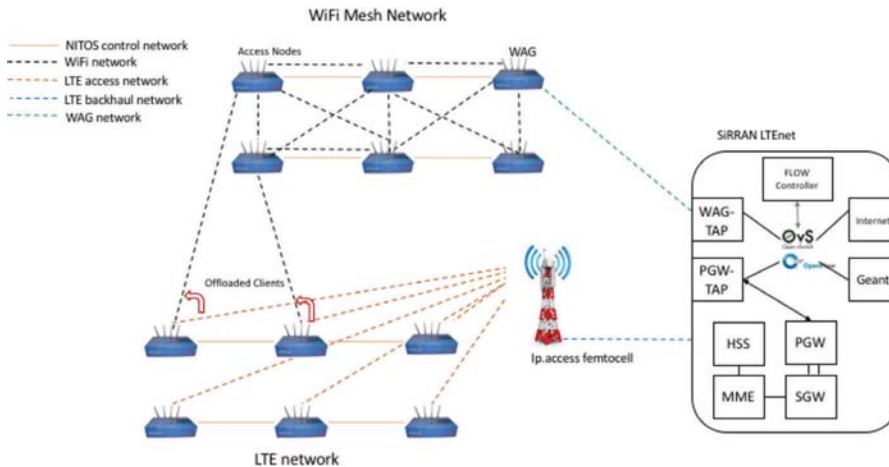


Figure 4.13 FLOW experiment setup.

Table 4.3 SLA setup for the FLOW offloading experiment

NITOS LTE Node	SLA for DL Bandwidth
Node054	15 Mbps
Node058	20 Mbps
Node074	10 Mbps
Node076	30 Mbps
Node077	7.5 Mbps
Node083	5 Mbps

The total capacity of the LTE network that the NITOS testbed is serving per femtocell is approx. 70 Mbps for the DL channel. Similarly, the total throughput (meaning the measured throughput from a client application) that the Wi-Fi mesh is achieving when using 2 hops is approx. 18 Mbps. Based on these given facts, and on the qualitative results that we expect to get from the theoretical framework that we have applied, always the client that is has the highest demand on DL bandwidth will be offloaded to the Wi-Fi network. If his/her demand cannot be met by the Wi-Fi network, the second highest in demand client will be selected to be offloaded, or else the third, etc.

Below we present some first experimental results and how the clients have been reallocated to use the Wi-Fi network, for the given SLAs. As we can see, the experimental analysis (Figure 4.14) matches the theoretical framework expectations. It is worth to mention, that for the validity of our results we used Wi-Fi bands in the 5 GHz band, so that there is no external noise or no overlapping with the rest of the 802.11 frequencies.

This experiment is depicting an example run from the FLOW offloading over the NITOS testbed. The clients are admitted to the LTE network every 10 seconds, and the FLOW framework is handling these requests for offloading them to the Wi-Fi network. For this experiment run, node054 and node058 are using the LTE channel for the first 20 seconds.

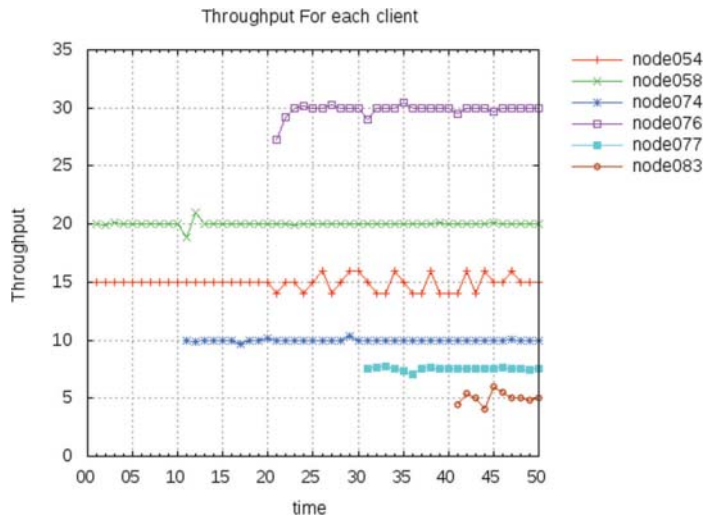


Figure 4.14 Throughput per each (offloaded) client.

When node074 is admitted to the network, the framework checks whether the node can be served by the Wi-Fi mesh network. However, the request for 30 Mbps DL traffic cannot be met by the Wi-Fi network and therefore the framework selects the second highest demanding client, which is node054. Similarly, when the rest of the clients are admitted to the LTE network, their total demand does not exceed the total LTE channel capacity. When the last client (node083) is admitted to the LTE network, the requested capacity will exceed the one that can be provided by the LTE channel. Therefore, the framework selects the most demanding client that is already served by the LTE network to offload to the Wi-Fi mesh. Nevertheless, the SLAs must be met at the Wi-Fi mesh network as well. Therefore, the choice that will make the best utilization of the network is the node083 itself, as it will be able to both get the remaining capacity of the mesh network and meet its demand for bandwidth.

Discussion

The potential of the FLEX federation of 4G and beyond testbeds has been demonstrated through the execution of some example experiments over the infrastructure. Yet, these are only a small portion of the experimentation capabilities of the platform, as several more have been proposed and are currently under execution. These include aspects regarding contemporary 4G network deployments, for either providing network measurements under a completely controlled environment, or developing new products designed for 4G and beyond applications, as well as aspects that will be addressed by the upcoming LTE releases and ultimately the 5G protocols, like for example narrow-band LTE development, Device-to-Device communications, NFV/VNF applications for the EPC, software defined backhauling for cellular networks, and even the development of software-defined base stations.

The platforms that are built through FLEX include high configurable equipment that is used for both development and evaluation of technologies for contemporary mobile networks, as well as for setting the cornerstone for the development of the first 5G pilots over the testbeds, using the open source software. Examples of such cases are also the experimental evaluation of functional splits for LTE over FLEX, the development of duplex schemes for wireless communications and others.

The high programmability of the platform and the vast potential that it has provides the community with the unprecedented chance to experimentally evaluate aspects for 5G networks using the existing infrastructure.

Moreover, the measurements that are provided by the testbeds, are given through open access to the community, thus enabling the implementation of algorithms regarding Big Data analysis, data mining techniques, etc.

Conclusions

FLEX is providing the infrastructure and platforms for the experimentally driven evaluation of scenarios including mobile broadband and potentially 5G networks. FLEX is filling a crucial gap in the existing infrastructures for the development of the Future Internet platforms, as it is the first pilot project that enhances FIRE's resource pool with cellular technologies.

In this chapter, we have presented briefly the FLEX platforms, and described the tools that have been developed in order to enable meaningful experiments to be executed over FLEX. These include tools for conducting federated experiments across the FLEX testbeds, always in line with the existing Fed4FIRE efforts in Europe, as well as for the user-friendly experimentation with the underlying equipment. Finally, some indicative use cases that take advantage of the infrastructure and platforms have been presented, as a means to demonstrate the potential of the platform. These include some crucial issues that are considered by the research community, such as the Wi-Fi and LTE coexistence in an unlicensed environment, as well as the Wi-Fi to LTE offloading process.

References

- [1] "3GPP global initiative for mobile broadband standards", [online] Available: <http://3gpp.org>
- [2] "Future internet research & experimentation", [online] Available: <http://ict-fire.eu>
- [3] "FIRE LTE Testbeds for Open Experimentation", [online] Available: <http://flex-project.eu>
- [4] Fdida, Serge, Timur Friedman, and Thierry Parmentelat. "OneLab: An open federated facility for experimentally driven future internet research." In *New Network Architectures*, pp. 141–152. Springer Berlin Heidelberg, 2010.
- [5] "Federation for FIRE testbeds – Fed4Fire", [online] Available: <http://fed4fire.eu>
- [6] "GEANT Pan-European network", [online] Available: <http://geant.org>

- [7] Nikaein, Navid, Mahesh K. Marina, Saravana Manickam, Alex Dawson, Raymond Knopp, and Christian Bonnet. “OpenAirInterface: A flexible platform for 5G research.” *ACM SIGCOMM Computer Communication Review* 44, no. 5 (2014): 33–38.
- [8] “NITOS – Network Implementation Testbed using Open Source platforms”, [online] Available: <http://nitlab.inf.uth.gr>
- [9] Bouckaert, Stefan, Wim Vandenberghe, Bart Jooris, Ingrid Moerman, and Piet Demeester. “The w-iLab. t testbed.” In *International Conference on Testbeds and Research Infrastructures*, pp. 145–154. Springer: Berlin, Heidelberg, 2010.
- [10] Maglogiannis, Vasilis, Dries Naudts, Ingrid Moerman, Nikos Makris, and Thanasis Korakis. “Demo: Real LTE Experimentation in a Controlled Environment.” In *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 413–414. ACM, 2015.
- [11] “OpenAirInterface Testbed”, [online] Available: <https://oailab.eurecom.fr/oai-testbed>
- [12] Sefraoui, Omar, Mohammed Aissaoui, and Mohsine Eleuldj. “OpenStack: toward an open-source solution for cloud computing.” *International Journal of Computer Applications* 55, no. 3 (2012).
- [13] “JuJu Service Modeling”, [online] Available: <http://www.ubuntu.com/cloud/juju>
- [14] “PerformNetworks testbed”, [online] Available: <http://morse.uma.es/performnetworks>
- [15] Merino-Gómez, Pedro. “PerformLTE: A Testbed for LTE Testing in the Future Internet.” In *Wired/Wireless Internet Communications: 13th International Conference, WWIC 2015, Malaga, Spain, May 25–27, 2015, Revised Selected Papers*, vol. 9071, p. 46. Springer, 2015.
- [16] “Polaris Networks, EPC Emulators Test Tools”, [online] Available: <http://www.polarisnetworks.net/epc-emulators.html>
- [17] “FUSECO Playground”, [online] Available: https://www.fokus.fraunhofer.de/go/en/fokus_testbeds/fuseco_playground
- [18] Peterson, L., R. Ricci, A. Falk, and J. Chase. “Slice-based federation architecture (SFA).” *Working draft, version 2* (2010).
- [19] “jFed tool.” [Online]. Available: <http://jfed.iminds.be/>
- [20] Augé, Jordan, Loïc Baron, Timur Friedman, and Serge Fdida. “Supporting the experiment lifecycle with MySlice.” In *Invited talk@ GENI Engineering Conference, GEC15-Oct*, pp. 23–25. 2012.

- [21] Mitchell, Tom. “GENI aggregate manager API.” In *Geni Engineering Conference 2010 (8th GEC)*, pp. 1998–2001. 2010.
- [22] Rakotoarivelo, Thierry, Maximilian Ott, Guillaume Jourjon, and Ivan Seskar. “OMF: a control and management framework for networking testbeds.” *ACM SIGOPS Operating Systems Review* 43, no. 4 (2010): 54–59.
- [23] Vandenberghe, Wim, Brecht Vermeulen, Piet Demeester, Alexander Willner, Symeon Papavassiliou, Anastasius Gavras, Michael Sioutis et al. “Architecture for the heterogeneous federation of future internet experimentation facilities.” In *Future Network and Mobile Summit (FutureNetworkSummit), 2013*, pp. 1–11. IEEE, 2013.
- [24] Singh, Manpreet, Maximilian Ott, Ivan Seskar, and Pandurang Kamat. “ORBIT Measurements framework and library (OML): motivations, implementation and features.” In *First International Conference on Testbeds and Research Infrastructures for the DEvelopment of NeTworks and COMmunities*, pp. 146–152. IEEE, 2005.
- [25] Stavropoulos, Donatos, Aris Dadoukis, Thierry Rakotoarivelo, Max Ott, Thanasis Korakis, and Leandros Tassiulas. “Design, architecture and implementation of a resource discovery, reservation and provisioning framework for testbeds.” In *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2015 13th International Symposium on*, pp. 48–53. IEEE, 2015.
- [26] Hibler, Mike, Robert Ricci, Leigh Stoller, Jonathon Duerig, Shashi Guruprasad, Tim Stack, Kirk Webb, and Jay Lepreau. “Large-scale Virtualization in the Emulab Network Testbed.” In *USENIX Annual Technical Conference*, pp. 113–128. 2008.
- [27] “NITOS Scheduler for NITOS-like testbed”, [Online]. Available: <https://github.com/NitLab/NITOS-Scheduler>
- [28] “SFI command line SFA tool”, [Online], Available: <http://www.fed4fire.eu/sfi/>
- [29] Makris, Nikos, Christos Zarafetas, Spyros Kechagias, Thanasis Korakis, Ivan Seskar, and Leandros Tassiulas. “Enabling open access to LTE network components; the NITOS testbed paradigm.” In *Network Softwarization (NetSoft), 2015 1st IEEE Conference on*, pp. 1–6. IEEE, 2015.
- [30] “ETSI TS 127 007 v 10.3.0, Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; AT command set for User Equipment (UE) (3GPP TS 27.007 version 10.3.0 Release 10)”.

- [31] Alexandris, Konstantinos, Navid Nikaein, Raymond Knopp, and Christian Bonnet. “Analyzing X2 Handover in LTE/LTE-A.” In *WINMEE 2016, Wireless Networks: Measurements and Experimentation*, May 9, 2016, Arizona State University, Tempe, Arizona, USA.
- [32] Alexandris, Konstantinos, Nikolaos Sapountzis, Navid Nikaein, and Thrasyvoulos Spyropoulos. “Load-aware Handover Decision Algorithm in Next-generation HetNets.” In *WCNC 2016, IEEE Wireless Communications and Networking Conference*, 3–6 April 2016, Doha, Qatar. 2016.
- [33] McKeown, Nick, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. “OpenFlow: enabling innovation in campus networks.” *ACM SIGCOMM Computer Communication Review* 38, no. 2 (2008): 69–74.
- [34] Makris, Nikos, Kostas Choumas, Christos Zarafetas, Thanasis Korakis, and Leandros Tassioulas. “Forging Client Mobility with OpenFlow: an experimental study.” In *WCNC 2016, IEEE Wireless Communications and Networking Conference*, 3–6 April 2016, Doha, Qatar. 2016.
- [35] “Open vSwitch, An Open Virtual Switch”, [Online]. Available: <http://openvswitch.org/>
- [36] Thomas Dietz. “Trema Tutorial.” (2012).
- [37] Rappaport, Theodore S. *Wireless communications: principles and practice*. Vol. 2. New Jersey: Prentice Hall PTR, 1996.
- [38] Cichon, Dieter J., and Thomas Kurner. “Propagation prediction models.” *COST 231 final report* (1999): 134.
- [39] Erceg, Vinko, K. V. S. Hari, M. S. Smith, Daniel S. Baum, K. P. Sheikh, C. Tappenden, J. M. Costa et al. “Channel models for fixed wireless applications.” (2001).
- [40] Shellhammer, Steve. “Overview of ITU-R P. 1238–1, Propagation Data and Prediction Methods for Planning of Indoor Radiocommunication Systems and Radio LAN in the Frequency Band 900 MHz to 100 GHz.” *doc.: IEEE802* (2000): 15–00.
- [41] Passas, Virgilios, Nikos Makris, Stratos Keranidis, Thanasis Korakis, and Leandros Tassioulas. “Towards the efficient performance of LTE-A systems: Implementing a cell planning framework based on cognitive sensing.” In *1st International Workshop on Cognitive Cellular Systems (CCS)*, pp. 1–5. IEEE, 2014.
- [42] Milorad Tomic, Valentina Nejkovic, Filip Jelenkovic, Nenad Milosevic, Zorica Nikolic, Nikos Makris, and Thanasis Korakis. “Semantic Coordination Protocol for LTE and Wi-Fi Coexistence”. In *European*

- Conference on Networks and Communications 2016: Testbeds and Experimental Research*, 26 June 2016, Athens. 2016.
- [43] Milošević, Nenad, Zorica Nikolić, Filip Jelenković, Valentina Nejkovič, Milorad Tošić, and Ivan Šeškar. “Spectrum sensing for the unlicensed band cognitive radio.” In *Telecommunications Forum Telfor (TELFOR), 2015 23rd*, pp. 250–252. IEEE, 2015.
- [44] “FLEX D5.13 FLOW: Experiment Description and Requirements”, [Online]. Available: <http://flex-project.eu>
- [45] Hierarchy Token Bucket (HTB), [Online]. Available: <http://linux.die.net/man/8/tc-htb>

