Hybrid Simulated-Emulated Platform for Heterogeneous Access Networks Performance Investigations

Igor Bisio, *Member, IEEE*, Alessandro Delfino *Student Member, IEEE*
Stefano Delucchi *Student Member, IEEE*, Fabio Lavagetto,
Mario Marchese, *Senior Member, IEEE*, Giancarlo Portomauro, Sandro Zappatore

***Abstract*—A communication network composed of heterogeneous access technologies can assure some benefits to mobile users. To exploit these benefits it is crucial to implement control techniques and algorithms to assure a proper level of Quality of Service (QoS). Performance studies are necessary to validate the proposed solutions before implementing them in real networks. Network simulators and emulators are useful tools.**
The main contribution of this paper is the description of a tool developed by the authors and called Hybrid Simulated and Emulated Platform (HySEP). HySEP is used to analyse the performance of different wireless networks such as Long Term Evolution (LTE) and Wi-Fi, simulated by using Network Simulator 3 (ns-3), connected to a transport network which implements the Differentiated Service (DiffServ) protocol, emulated through a group of virtual PCs. HySEP enables the creation of a simulated mobile node which implements heterogeneous network interfaces and can execute vertical handover while it is communicating with a real node. HySEP validation tests represent a further contribution of this paper.

***Index Terms*—Hybrid Real-Simulated Network, Performance Investigation, ns-3, Quality of Service, Heterogeneous Access Networks.**

I. Introduction

THANKS to the availability of different Radio Access Technologies (RATs), such as Wi-Fi and Long Term Evolution (LTE), telecommunication networks are able to support a new plethora of services and applications anytime and wherever the user is located. Typical heterogeneous networks are composed of different segments implementing different protocols and Quality of Service (QoS) solutions. In particular, to exploit the advantages of such heterogeneity, mobile nodes have to be equipped with multiple network interfaces. In this context, a fundamental issue is vertical handover that is the action of changing the Radio Access Network (RAN) used by a mobile node. Vertical handover is based on a decisional process in charge of selecting which network a mobile node has to use. Other important topics are flow identification, scheduling, policing, call admission control, routing, and resource allocation [1].
Considering the aforementioned scenario it is crucial to deeply test and validate new protocols and control algorithms through network simulators and/or emulators before applying them in real networks. Simulation tools are usually cheaper and allow tuning parameters simply. Emulation ones are able to handle real traffic flows. The ideal solution is to have a tool where both approaches are integrated so exploiting their advantages. The main topics of this paper are: the description of the tool Hybrid Simulated-Emulated Platform (HySEP), developed by the authors, and the discussion of the tests aimed at validating its structure and functionalities. HySEP can simulate different access networks (Wi-Fi and LTE are implemented in this paper) through the Network Simulator 3 (ns-3) and emulates a core network through a set of Virtual Personal Computers (VPCs), which, in this paper, implement the Differentiated Service (DiffServ) protocol [2] to assure QoS. This choice, on one hand, enables the test of access technologies without the use of real networks, and, on the other hand, makes feasible the transmission of real traffic flows from/to the simulated part of the network.
Linux-based operative systems offer a wide variety of network traffic control functions located partially in the kernel-space and partially in the user-space of a PC. In particular, some of them implement the mechanisms required to support the DiffServ architecture [3]. The Linux Traffic Control (tc) software plays a fundamental role because it implements these traffic control operations [4]. In more detail, tc filters each packet and, according to the DSCP value, on which the DiffServ paradigm is based, assigns it to a specific queue that contains all the packets belonging to the same traffic class (identified by the DSCP value). According to the policy adopted to serve the queues in the output interface it is possible to differentiate the service received by each traffic class. The rest of the paper is organized as follows: the next Section presents the structure of HySEP. Section III contains a presentation of HySEP main features and requirements, as well as the structure of the performed tests. Section IV presents a short overview of the state of the art regarding available network simulators together with the explanation for the use of ns-3 within HySEP. Section V describes ns-3, with particular reference to the models used to simulate LTE and Wi-Fi. Section VI and Section VII represent the core of this paper. They describe the HySEP architecture and the results of validation and scalability tests, respectively. The conclusions are discussed in Section VIII.

II. HYSEP STRUCTURE

The scenario taken as a reference for HySEP implementation is shown in Figure 1. It is composed of two heterogeneous wireless access networks: LTE and Wi-Fi, connected to a Core

doi:10.4304/jnw.10.4.265-272

Network which implements the Differentiated Service protocol to manage QoS. Two Edge Routers (ERs) are located at the frontiers of the core domain: one of them communicates with LTE and Wi-Fi networks, the other one is connected to a real remote host. Three different types of terminals (also called nodes in this paper) are included in this scenario: i) Wi-Fi terminals, called Station (STA) nodes; ii) LTE terminals, called User Equipments (UEs); iii) multi-interface nodes (MIN), each of them equipped with both network interfaces. The terminals of the first two types communicate with the remote host by using the supported technology, while the terminals of the third type can use both technologies. The elements composing HySEP will be described in section VI.

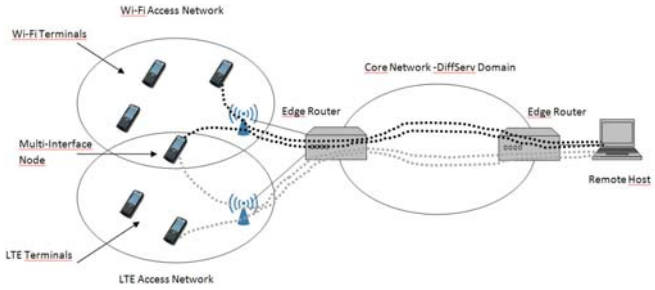

Fig. 1. Reference scenario.

III. HySEP FEATURES, REQUIREMENTS, AND PERFORMED TESTS

As said above the main HySEP characteristic is the power to manage real traffic flows. An important feature supported by HySEP is the capability to define a mobile Multi Interface Node (MIN), defined in the previuos section and shown in Figure 1, equipped with different RAT interfaces and able to execute vertical handover from Wi-Fi to LTE and vice versa, while keeping active the communication. Another important HySEP feature is the capability of adding a delay between the instant when the old network is no longer available and the instant when the new network is available. In practice it is possible to introduce a time period where no connection is available during the handover execution. Varying the duration of this period is possible to evaluate the effects of an "hard handover" on different transmitted traffic flows. In this way HySEP enables the analysis of the vertical handover impact on the end-to-end QoS.

A fundamental HySEP requirement is the ability to keep the synchronization between the simulated access network, called Simulated Network Segment (SNS), and the emulated core network, called Emulated Network Segment (ENS). This synchronization is a keypoint to get reliable results when simulated and emulated network nodes are interconnected. To obtain a correct behaviour of the platform in real time mode, simulation time must correspond to real time. In other words, one second inside the simulation must be equal to one real second. This requirement justifies, as should be clear in the following of the paper, the utilization of the simulation tool ns-3 that enables the use of virtual devices tap and bridge to forward real packets to the simulated network and to forward simulated packets to a real network.

We decided to use only open software, under free license, in order to use it freely and to enable the addition of new modules for network control and management.

A necessary step in HySEP definition is its validation through a test campaign. We have evaluated two different aspects and we show and discuss the results of two different sets of tests:

- First of all we test a set of possible scenarios that can be represented by using HySEP with the aim of checking the limit until the synchronization between SNS and ENS is kept. Different configurations are considered by changing number of nodes and transmitted data rates. The tests are based on the use of Iperf software, which is able to measure the supported data rate between simulated nodes and real remote host. If the obtained throughput and execution time are equal to the real values imposed by the authors during the configuration, the synchronization between the segments is maintained, and, consequently, the platform works appropriately. Otherwise the synchronization is not assured and the results of the tests are not reliable.

- The second group of tests is aimed at getting the performance evaluation of the handover process. In this case a simple UDP traffic flow is transmitted by a multi interface node and some handovers, at different delay values, are executed. Different aspects have been evaluated: the path followed by the traffic flows, to verify the correct handover execution; the obtained throughput; and the packet loss. HySEP can both measure the number of lost packets and identify which packets are lost and when.

IV. State of the Art

Several network simulators are available, each of them assuring different features and capabilities. Among the others: *i)* OMNeT++ and *ii)* OPNET; *iii)* Network Simulator 2 (ns-2) and *ii)* Network Simulator 3 (ns-3). The aim of this section is to present a brief comparison among them (see also [5] and [6]), discussing their strong and weak points, and justifying our decision to use ns-3 within HySEP.

The comparison proposed in this section takes into consideration the main HySEP requirements, summarized again below:

- implementation of LTE and Wi-Fi models; support to the creation of a multi interface mobile node and to the handover execution.
- support of a full integration of simulated and real networks, assuring the capability of handling real traffic flows; schedule of simulation events in real time.
- open software implementation.

A. Important available network simulators

Objective Modular Network Testbed (**OMNeT++**) is a C++, open source framework for discrete event network simulation. It has a modular structure. Modules can be combined together by using an high level language called NEtwork Descriptor (NED) to build complex and realistic network scenarios. OMNet++ supports the simulation of technologies such as Wi-Fi, WiMAX and UMTS. LTE is partially supported

because OMNet++ includes only a limited set of details concerning radio channel, mobile nodes and base stations. It supports the connection between simulated and real networks. Another network simulator largely used in many research fields is OPNET Simulator, called **OPNET Modeler**. It is a discrete event-simulator property of the OPNET Technologies Inc. company. Its hierarchical structure is composed of different models written in C/C++. OPNET Modeler implements many network protocols, functions and architectures such as wired, wireless, and satellite networks; and supports a detailed simulation of LTE and Wi-Fi networks. This tool assures the synchronization of the simulation in a real time environment so assuring the correct interconnection of simulated and emulated networks.

Network Simulator 2 (**ns-2**) is an open source discrete event simulator written in Objective Tool Command Language (OTcl) and C++. Different protocols and technologies are available in ns-2, such as InterServ and DiffServ, satellite network, wireless sensor network, Wi-Fi, WiMAX, UMTS. No official LTE model has been developed up to now. Ns-2 can support the integration of simulated and emulated networks.

Network Simulator 3 (**ns-3**), the new version of ns-2, is a free and open source discrete event network simulator completely written in C++. It can reastically simulate complex network scenarios including LTE/EPC technology, as explained in the next section, and other wireless technologies such as Wi-Fi. It also supports the creation of a node equipped with heterogeneous network interfaces and able to execute the vertical handover. Ns-3 enables also the integration of the simulation within a real network thanks to its support to real time simulation.

Considering the aforementioned requirements and the features of the listed simulators, ns-3 seems to be the most reasonable choice for HySEP.

V. NS-3 FEATURES USED IN HYSEP

This section contains a brief description of the most important ns-3 features used to implement and configure HySEP.

A. The ns-3 simulator

1) Ns-3 real time simulation mode: Using ns-3 it is possible to connect simulated and emulated networks, exchanging packets each other. To do this it is necessary to appropriately configure both the simulator and the host PC running ns-3. On one hand the simulator needs a real-time event scheduler to synchronize the ns-3 clock with the clock of the host PC. The aim is to prevent possible misbehaviours in the packet forwarding process between simulated and real networks due to a lack of synchronization. On the other hand the host PC must use virtual interconnection devices, such as *taps* and *bridges*, to connect the simulation, located in the user space, with the Ethernet interface located in the kernel space. In this way, the traffic generated by the simulated nodes is received by the tap and forwarded to the Ethernet interface by using the virtual bridge. To connect tap and simulated nodes, ns-3 requires the use of a particular node, which is configured through the *TapBridgeHelper* API and acts as a sort of alias of the tap within the simulation.

2) Ns-3 Long Term Evolution - Evolved Packet Core (LTE-EPC) model: Ns-3 supports some models for the simulation of different technologies; one of them is focused on the LTE-EPC network [7], [8]. The first effort to create an LTE module within ns-3 framework is the LENA project, dated 2011 and officially integrated in ns-3 in May 2012. LENA provides two distinct models: *i)* the LTE model which includes the whole radio protocol stack implemented in the mobile nodes, which are called User Equipment (UEs), and in each base station, which is called evolved Node B (eNB); *ii)* the EPC model that represents the core network. This model enables the simulation of the Serving Gateway (SGW) and of the Packet Data Network Gateway (PGW), see [7], [8] for details, whose functionalities are collapsed in a single node, called SGW/PGW, and of the Mobility Management Entity (MME). The model partially integrates the eNB implementation proposed by the LTE model.

A fundamental capability of the LTE module is to create and maintain radio bearers that represent the wireless segments between UEs and eNBs within the whole LTE-EPC bearers. It provides detailed channel models and mobility scenarios, implementing all the layers of the UE and eNB stack: Radio Resource Control (RRC) [9], Packet Data Control Protocol (PDCP) [10], Radio Link Control (RLC), Medium Access Control (MAC) [11], and Physical (PHY).

3) Ns-3 Wi-Fi model: Thanks to its modular structure, ns-3 enables the simulation of many different networks. A generic network node can be equipped with a network interface, such as WiFi, as described in this subsection, by using the following modules: *WiFiNetDevice* and *WifiHelper*.

The ns-3 Wi-Fi network model is composed of two different parts described below:

- The physical layer, which includes detailed channel and terminal mobility models, and is configurable through the *WifiPhyHelper* API.
- The Medium Access Control (MAC) layer, which is divided into lower and higher layer. The former is devoted to manage packets fragmentation, transmission and confirmation, and to execute carrier sense procedures. The latter enables the definition of different node functionalities like Access Point (AP) or non-AP station (called STA). This layer can be easily configured by using API *WifiMacHelper*.

VI. HYSEP STRUCTURE

The Hybrid Simulated-Emulated Platform (HySEP) is composed of the following elements evidenced in Figure 2:

- Simulated Access Network (SAN): mobile nodes/terminals are simulated by using the Network Simulator 3, ns-3. Network nodes use a simulated wireless access network to communicate with the core network and to reach the remote host. SAN is composed of a Personal Computer (PC), PC1, which runs ns-3 simulations and all necessary operations to forward real traffic flows from the simulated to the emulated network and vice versa. PC1 is connected to PC2 by using Ethernet.

- Emulated Core Network (ECN): it is composed of a network of virtual machines (VMs) created and managed by using VirtualBox software [12]. From a structural viewpoint this part is composed of a single PC, PC2, which implements VMs. PC2 is connected to PC3 through Ethernet.
- Real Remote Host: it is a PC (PC3) that acts as a terminal host (or, alternatively, as a server) that communicates with the simulated node inside the ns-3 simulation. It is aimed at representing an end point for the up-link traffic flows generated by the simulated nodes, and at collecting and displaying statistics about these flows.

SAN-ECN Gateway is a logical element, implemented through a set of functions and virtual interfaces acting in the kernel space, aimed at connecting simulated and emulated portions.

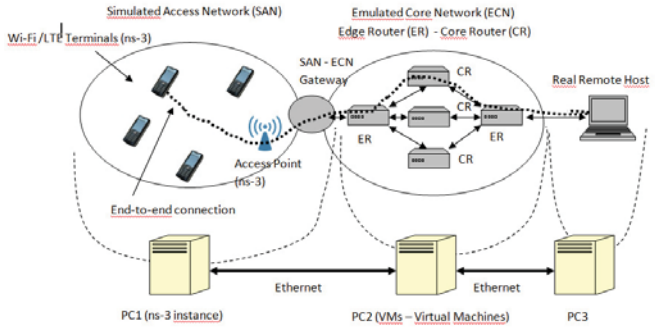

Fig. 2. Hybrid Simulated-Emulated Platform (HySEP) implementation.

A. Simulated Access Network - SAN

The implemented scenario is composed of a single Wi-Fi Access Point (AP) and a single LTE base station, the eNB. Both access points (AP and eNB) are connected to a node called SGW/PGW that implements the functionalities of both SGW and PGW nodes, as said before. The link between AP and SGW/PGW is a point-to-point link, while the correspondent link in the LTE network is implemented through the GPRS Tunnel Protocol (GTP) defined in the LTE standard.

The Simulated Access Network configuration is mainly based on three elements: 1) Network Simulator 3 (ns-3), which is aimed at simulating terminals and access nodes of different wireless access technologies. 2) Direct Code Execution (DCE) tool, which allows the integration of real applications within simulation scripts. 3) Virtual interface and bridges (tun/tap and bridge, respectively) used to connect the simulated network with the physical interface of the PC where the simulation is executed.

An important feature of HySEP SAN is to simulate the Multi Inteface Node (MIN), and, in particular, a MIN equipped with the following two heterogeneous interfaces: *i)* Long Term Evolution (LTE) and *ii)* Wi-Fi. The ns-3 configuration script in C++ defining MIN is reported below. MIN is defined by using a pointer called `Ptr<Node>`. Two different "containers" are used: one `NodeContainer` contains the MIN while two `NetDeviceContainers` contain the two network interfaces. This configuration is necessary to avoid conflicts in the IP address assignation to both interfaces.

```
Ptr<Node>Min = CreateObject<Node>();
...
NodeContainer MinC;
mmUeNodeC.Add(Min);
NetDeviceContainer ueLteDevs =
lteHelper->InstallUeDevice (MinC);
...
NetDeviceContainer staDevices =
wifi.Install (phy, mac, Min);
```

This configuration allows to easily select the interface to be used according to the selected technology. For example, if, at the beginning of the transmission, we decide to use the Wi-Fi interface, the initial route to the remote host is consequently configured as follows:

```
Ptr<Ipv4StaticRouting>multiStaticRouting =
ipv4RoutingHelper.GetStaticRouting(
Min->GetObject<Ipv4>());
multiStaticRouting->AddNetworkRouteTo(
Ipv4Address ("192.168.0.0"), Ipv4Mask
("255.255.255.0"), Ipv4Address
("10.1.3.1"), 1);
```

`AddNetworkRouteTo()` sets the route in the MIN for the network 192.168.0.0/24, where the remote host is located, by using the interface number 1 and the IP address 10.1.3.1 (i.e. the AP address) as a next hop. This route can be changed: the same network can be reached by using the LTE access network. To do this, at a given instant *t*, the third entry in the routing table, the one referred to the remote host, is removed and a new route is inserted as shown below:

```
Simulator::Schedule(Seconds(t),
&removeRoutingEntry, multiStaticRouting,
3);
Simulator::Schedule(Seconds(t),
&addRoutingEntry, multiStaticRouting,
Ipv4Address ("192.168.0.0"),
epcHelper->GetUeDefaultGatewayAddress (),
2);
```

These lines add a route in the MIN where the next hop for the remote host is the default LTE gateway (i.e. the eNB) reachable through the interface number 2.

As highlighted in section III, it is possible not only to execute vertical handover but also to introduce a delay between the cancellation of the old route and the insertion of the new one, so allowing to evaluate the impact of this delay in the transmission of a real traffic flow, as happens for hard handover.

B. Emulated Core Network (ECN)

The Emulated Core Network is composed of virtual machines and virtual links. Their configuration is mainly based on

the following elements: Virtual Box software for the creation and management of virtual machines; traffic control (tc) tool used to characterize the core link in terms of packet rate and delay (in particular the Token Bucket Filter (TBF) is adopted to filter the packets in transit, limiting the achievable rate and so conforming the traffic); virtual interface and bridges (tun/tap and bridge) used to connect the virtual machines, representing the core network links.

We use the Virtual Box to create 3 Virtual Machines (VMs) representing the core network nodes that implement the Diff-Serv protocol. One of them is a Core Router (CR) and two of them are Edge Routers (ERs). Virtual computers are equipped only with the necessary hardware components to ensure the minimization of the hosting PC (i.e. PC2) computational load. The number of possible VMs is limited by the computational capacity of the hosting PC. The advantage of virtualization is the capability of running many computers in a simple, compact and cheap way.

After creating and configuring the core network nodes it is necessary to create links among them. Links are implemented by virtual network devices called tap. The interconnection between taps is achieved through a virtual bridge, whose ports are taps themselves. A rate limiter to emulate real backbone interconnections is set on each link. This emulation is achieved by using a traffic shaper installed in each tap. The traffic shaper is based on the Token Bucket Filter (TBF) algorithm and is configured through the Linux OS tool Traffic Control (tc). So virtual PCs are suitably interconnected by using bridge and tap devices. Two different connection configurations are adopted: *i)* the CR is connected to the ERs (the other virtual PCs) through tap and bridge, *ii)* each ER uses tap and bridge to communicate with the CR on one side and with an Ethernet interface of PC2 on the other side.

Using tc is possible to add details to each link in the transportation network such as bandwidth and propagation delay. tc is a software, developed for Linux operative systems and used to show and manipulate traffic control settings including the queueing discipline (qdisc), defined as the set of rules that define how the packets are handled in a network interface. As previously said, these nodes implement the DiffServ protocol to manage the QoS in the core network.

VII. HySEP Validation Tests

The results of the performed tests are reported in this section. The section is structured into two separate parts: the first one is dedicated to check the limit on the number of nodes that can be simulated without losing the synchronization between SNS and ENS, as introduced in section III. The second one is dedicated to validate the vertical handover execution. Table I contains PC1 characteristics, where ns-3 is run. For each of the two mentioned parts of the tests a different reference scenario is considered but both of them are composed of the same ENS, described in Section VI. They differ each other for the simulated scenario characteristics as described below:

- The first scenario is composed of two different access networks, LTE and Wi-Fi. Both technologies are simulated by varying number of users and required throughput in order to find out the maximum number of nodes that can be simulated maintaining the real time synchronization. Each mobile node implements a single network technology in each test. Two quantities are measured by using the tool *Iperf* [13]: *i)* the duration of *Iperf* data transmission and *ii)* the measured throughput between the simulated nodes and the remote host. The real time simulation is correctly supported if the *Iperf* transmission time duration and the obtained throughput are equal to the set values. Simulation duration and *Iperf* connection duration are constantly equal to $70[s]$ and $60[s]$, respectively. The *Iperf* tool uses a User Datagram Protocol (UDP) flow between the simulated network and the remote host.
- The second scenario is composed of a single Multi Interface Node (MIN), which is the source of the packets, equipped with an LTE interface and a Wi-Fi interface. It communicates with the remote host within PC3 by using alternatively one of the available networks. MIN must execute the handover while keeping active the communication. It transmits a UDP traffic flow to the remote host in PC3. The transmission duration is equal to 15 [s] and the transmission data rate is constantly equal to 1 [Mbps]. The *Tshark* tool is used to collect statistics such as the number of received packets and the data rate, as well as to compute the packet loss.

TABLE I
Characteristics of the PC used for the ns-3 simulations (PC1).

CPU	Intel Core i5-3450S - 2.80GHz x 4
RAM	8 GB - DDR3 - PC3-12800
Operative System	Ubuntu 13.10 - 32 bit
CL	9

A. Synchronization tests

The reference scenario, adopted for this set of tests, is composed of two access networks: LTE and Wi-Fi. A single eNB and a single Access Point (AP) are connected to a common node, the SGW/PGW, which implements the functionalities of the SGW and the PGW. This node receives the traffic flows from the eNB and the AP and forwards them to the core network.

Two different types of tests are carried out:

- In the first test, identified as LTE-EPC, only the LTE access network is considered by varying the number of UE nodes within the range [1 - 8]. In more detail: the network of this test is composed of a special UE, called *Iperf* node, which implements the *Iperf* client functionalities and transmits traffic flows at a variable data rate to a *Iperf* server inside the remote host. The other UEs, called background nodes, communicate each other in pairs with a fixed data rate equal to 800 [kbps].
- In the second test, identified as Wi-Fi, the variability stands in the number of Wi-Fi nodes, called STA nodes, which is varied in the range [1 - 8]. Also in this case there is a single Iperf node and the other STA nodes (background nodes) communicate each other in pairs

with a fixed data rate equal to 800 [kbps]. There are no transmitting UE nodes in this case.

1) LTE-EPC test: as said in Section V the LTE-EPC module implemented within ns-3 is very detailed but, at the same time, it is quite complex and requires a big amount of computational resources. The results proposed in Figure 3 and Figure 4, which show, respectively, the *Iperf* transmission time and the obtained throughput measured by using *Iperf*, confirm this consideration. In particular, only a limited number of background nodes can be simulated in real time mode. Consider, for example, the cases when the *Iperf* node data rate is equal to 2 Mbps: the simulation is reliable up to 5 nodes (i.e., the *Iperf* node and 4 background nodes). Above this number the *Iperf* transmission duration increases with respect to the expected 60 [s] value and, consequently, the obtained throughput decreases making the test unreliable. A similar behaviour characterize all the considered *Iperf* node data rates. These results limit the set of possible LTE-based scenarios in

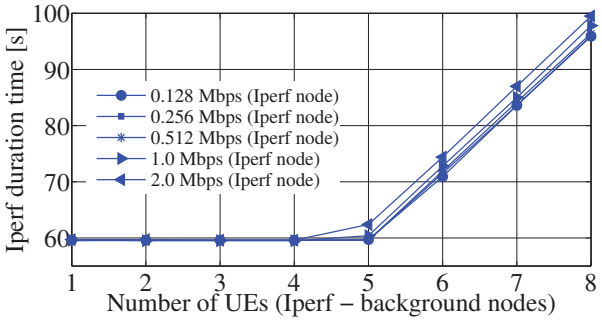

Fig. 3. Iperf transmission time for different transmitting data rates of the Iperf node vs different number of background nodes.

which HySEP can be adopted. Obviously the reliability of the tool can be improved by using a PC with higher performance, so enabling the simulation of a larger number of UE nodes. It is worth noticing that several other similar tests are already carried out through the LTE-EPC network but they are limited to non real time scenarios. According to the authors' best knowledge this paper is one of the first attempts to study and discuss LTE-EPC simulation scalability by using ns-3 in real time mode.

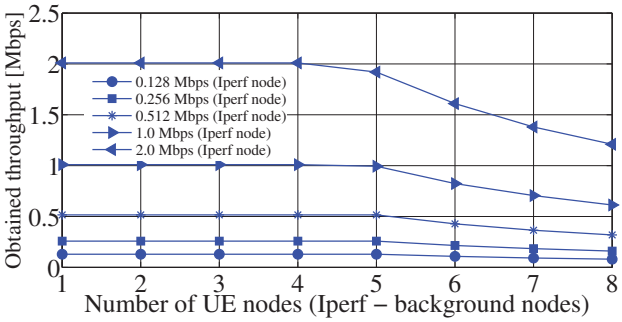

Fig. 4. Obtained throughput for different transmitting data rates of the Iperf node vs different number of background nodes.

2) Wi-Fi test: a simpler module, with respect to the LTE-EPC one, is implemented in ns-3 to support Wi-Fi network simulation. Consequently, a larger number of nodes / higher data rates can be simulated. The main goal of these results is the same of the previous case. Considering the same number of *Iperf* and background nodes as in the LTE-EPC case, Figure 5 and Figure 6 show that higher *Iperf* data rates can be supported in real time. In this case the link between *Iperf* data rate and maximum number of nodes that can be simulated is more evident. Two examples may help undestand: when the *Iperf* data rate is equal to 12 Mbps, up to 7 STA nodes (i.e., the *Iperf* node and 6 background nodes) can be supported; when the *Iperf* data rate is equal to 20 Mbps, the maximum number of supported nodes is 3 (i.e., the *Iperf* node and 2 background nodes). If the number of nodes exceeds these values the *Iperf* transmission time is higher then its expected value (equal to 60 [s]) and the real time synchronization is compromised. Consequently, it is possible to observe that, in these cases, the obtained throughputs are lower than the expected values.

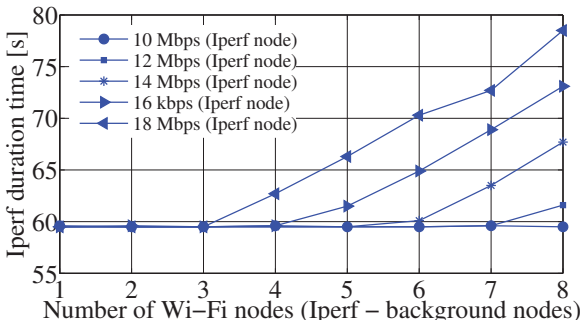

Fig. 5. Iperf transmission time for different transmitting data rates of the Iperf node vs different number of background nodes.

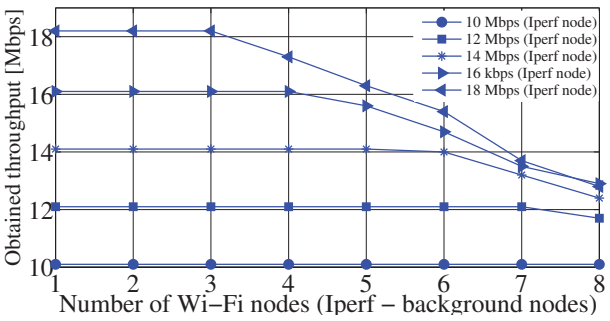

Fig. 6. Obtained throughput for different transmitting data rates of the Iperf node vs different number of background nodes.

B. Vertical Handover test

This subsection is focused on the validation of the multi interface node, with particular reference to the vertical handover execution. As previously said, this node, called Multi Interface Node (MIN), has two different interfaces, Wi-Fi and LTE. Consequently, it can use alternatively both technologies. Handover is implemented by modifying the routing table of the node, as explained in the previous section. As shown in Figure 7 the MIN initially uses the Wi-Fi technology to communicate with the remote host. Its route entry number 3, highlighted in Figure 7, indicates that the remote host, located in the network 192.168.0.0/24, can be reached through the port 1 used by the Wi-Fi interface. The Wi-Fi AP, whose IP

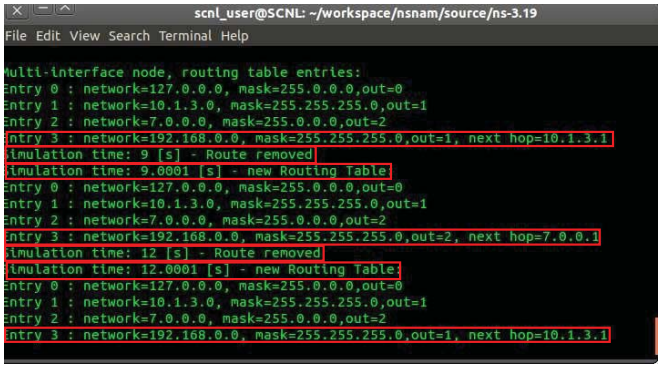

Fig. 7. Simulator visual output: routing table and handover execution.

address is 10.1.3.1, is set as next hop. At instant 9.0001 [s] the MIN executes an handover and the route entry is updated. The Wi-Fi network is no longer used and the communication is carried on through the LTE network. Consequently the MIN communicates with the remote host by using the port 2 (LTE) and the eNB (address 7.0.0.1) as next hop. Finally, at instant 12.0001 [s], another handover is performed: the communication is redirected again through the Wi-Fi network and the routing table is set again to the initial one.

Two different scenarios are used to perform these tests. In both cases the MIN carries out two handovers as previously explained. The difference stands in the time required to complete the handover: in Scenario 1 (S1) a very limited delay (equal to 0.001 [s]) is introduced between the cancellation of the old route and the insertion of the new one. This is the case shown in Figure 7. In Scenario 2 (S2) the delays for the first and second handovers are set to 1.0 [s] and 2.0 [s], respectively.

Figure 8 shows the received data rate measured in the remote host by using *Tshark* for Scenario 1. The transmission starts at instant 4 [s] because the first four seconds are dedicated to set parameters. The first handover is carried out within 0.001 [s] at instant 9 [s], where Wi-Fi rate drops to 0 and LTE rate raises up to 1 Mbps. The opposite happens at instant 12.0 [s]. It is important to note that the received data rate is constantly equal to 1 Mbps, as expected. Due to the very limited handover delay no interruption in the received flow is measured at the remote host. We determine the technology used by the MIN by checking the IP source address of each received packet: 7.0.0.2 for LTE and 10.1.3.2 for Wi-Fi.

Scenario 2 is characterized by higher handover delays that

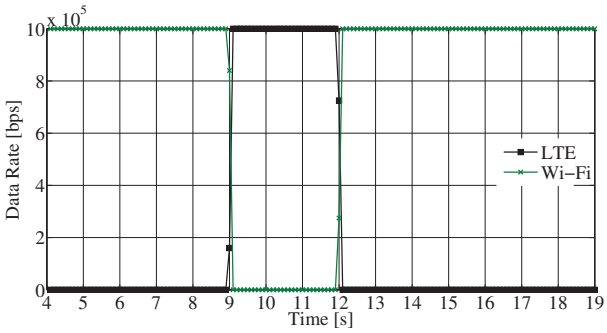

Fig. 8. Received data rate measured by the remote host for Scenario 1 (S1).

heavily affect the data rate measured at the remote host, as shown in Figure 9. In correspondence of the two handovers at instants 9 [s] and 12 [s], the received traffic flow is interrupted (no bit rate is measured at the remote host) for the whole duration of the handover, respectively equal to 1.0 [s] and 2.0 [s].

These tests show the impact of the time required to complete the end-to-end QoS and show the chances offered by HySEP for this type of analysis so opening the door to the test of more complex handover control algorithms, aimed at mitigating the handover delay.

If the application within the MIN is properly configured,

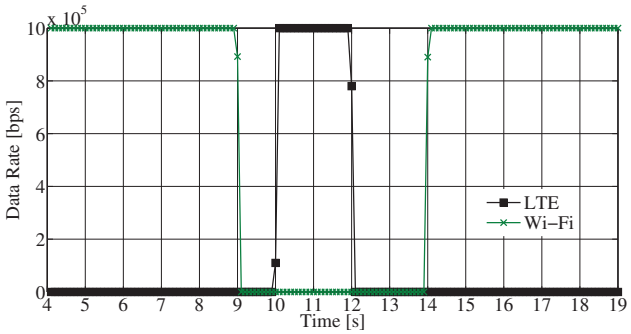

Fig. 9. Received data rate measured by the remote host for Scenario 2 (S2).

HySEP can measure transmitted and received packets, and compute the packet loss. It is also possible to individuate which are the specific lost packets during the handover. To reach this goal, we developed a simple Constant Bit Rate (CBR) application within the MIN and we added a progressive packet ID in each packets payload. The packet ID is extracted from the payload at the remote host. Figure 10 shows the ID from each received packet for both Scenarios: packet IDs are reported on y axis vs the number of received packets, reported in the x axis. 3384 packets are transmitted. All of them are

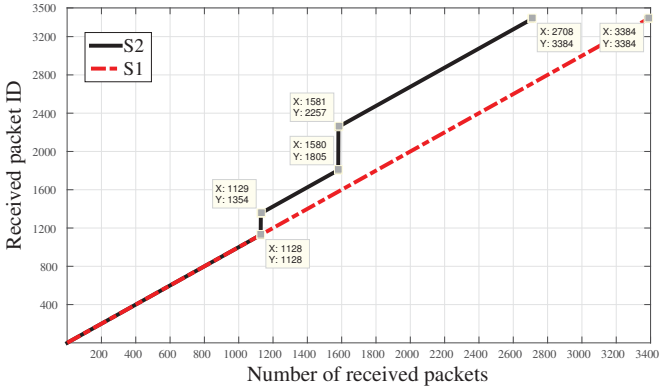

Fig. 10. Packet IDs received by the remote host.

received in S1. This is not true in S2, as already clear in Figure 9. The additional information with respect to Figure 9 is the detail about each single lost packet and the chance to get packet loss metrics. At the execution of the first handover the packets whose ID is in the range [1129-1352] are lost because the last received packet before the handover has ID 1128 and all packets from ID 1129 up to ID 1352 are lost.

Equivalently, at the second handover, the packets whose ID is in the range [1806-2257] are lost. So, during the two handovers 224 and 452 packets are respectively lost getting an overall number of lost packets equal to 676. The same number can be obtained again from Figure 9 by using the overall number of transmitted (3384) and received (2708) packets during the whole transmission. The packet loss rate is $676/3384 \approx 0.2$.

VIII. Conclusions

This paper is focused on the description and validation of a simulation-emulation tool called Hybrid Simulated-Emulated Platform (HySEP), developed by the authors. This platform is composed of two different parts: wireless access network, simulated using ns-3, and emulated core network, implemented through three virtual PCs.

The most important faced challenge has been the proper interconnection of simulated and emulated network: it was necessary to keep the synchronization of the simulation with the real nodes within the emulated network in order to support a correct packet exchange between simulated and emulated nodes.

HySEP can simulate the behaviour of a Multi Interface Node (MIN), which is a mobile terminal equipped with multiple interfaces (Wi-Fi and LTE in the shown results), and of the handover process.

Two sets of tests have been carried out to validate the platform: *i)* the first set is aimed at analysing the scalability of HySEP, getting the limit on the number of mobile nodes and on the transmitted data rate for which the synchronization between simulated and emulated portion can still be kept; *ii)* the second set is focused on the analysis of the handover process and on the chances offered by HySEP in this context.

References

[1] M. Marchese, *QoS over Heterogenuous Networks*. John Wiley and Sons, Ltd, Chichester, England, 2007.

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," RFC 2475, The Internet Engineering Task Force (IETF), December 1998. [Online]. Available: http://www.rfc-editor.org/rfc/rfc2475.txt

[3] W. Almesberger, J. H. Salim, and A. Kuznetsov, "Differentiated services on linux," in *GLOBECOM*, GLOBECOM, Ed., vol. 1, 1999, pp. 831–836.

[4] B. Hubert, "Linux advanced routing & traffic control." [Online]. Available: http://www.lartc.org/

[5] E. Weingartner, H. vom Lehn, and K. Wehrle, "A performance comparison of recent network simulators," in *Communications, 2009. ICC '09. IEEE International Conference on*, June 2009, pp. 1–5.

[6] A. Khan, S. Bilal, and M. Othman, "A performance comparison of open source network simulators for wireless networks," in *Control System, Computing and Engineering (ICCSCE), 2012 IEEE International Conference on*, Nov 2012, pp. 34–38.

[7] M. Olsson, *SAE and the Evolved Packet Core: Driving The Mobile Broadband Revolution*. Academic Press, 2009.

[8] L. Korowajczuk, *LTE, WIMAX, And WLAN Network Design, Optimization And Performance Analysis*. Chichester, West Sussex, U.K.: Wiley, 2011. [Online]. Available: http://isbnplus.org/9780470741498

[9] "Evolved universal terrestrial radio access (E-UTRA); radio resource control (RRC); protocol specification," 3GPP TS 36 331, ETSI, 2014.

[10] "Evolved universal terrestrial radio access (E-UTRA); packet data convergence protocol (PDCP) specification," 3GPP TS 36 323, ETSI, 2013.

[11] "Evolved universal terrestrial radio access (E-UTRA); medium access control (MAC) protocol specification," 3GPP TS 36.321, ETSI, 2014.

[12] Oracle, "Virtualbox." [Online]. Available: https://www.virtualbox.org/

[13] M. Gates, A. Tirumala, J. Ferguson, J. Dugan, F. Qin, K. Gibbs, and J. Estabrook, "Iperf." [Online]. Available: http://sourceforge.net/projects/iperf/