

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—The availability of different access technologies enables the creation of heterogeneous networks supporting users mobility and assuring several different services. Meanwhile these networks require complex control techniques to assure Quality of Service (QoS) to users. Before implementing such networks, a deep performance analysis, through the use of network simulators or real models, is necessary. In particular the first ones (e.g. Network Simulator 3 - ns-3 among the others) are quite simple and easy to manage and configure, while the second ones assure the handling of real traffic flows.

The main contribution of this paper is the description of a hybrid simulated and emulated network evaluation platform, developed by the authors. The platform purpose is to execute a performance analysis of different wireless networks such as Long Term Evolution (LTE) and Wi-Fi, connected to a core network implementing the Differentiated Service (DiffServ) protocol. The paper contains also the results of preliminary validation tests.

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

I. INTRODUCTION

NOWADAYS the development of different network technologies and protocols is able to support a large multitude of services and applications. Typical network architectures are composed of different heterogeneous segments interconnected each other: a user may pass through several different domains to reach its remote destination. Several optimization techniques and protocols are applied to the aforementioned network to support the Quality of Service (QoS).

Within this framework it is fundamental to test and validate the developed protocols and algorithms before applying them in real networks. Consequently, a tool, able to model in details complex networks, realizing realistic scenarios but at the same time limiting complexity, cost and dimension, could assure a useful support in several research activities.

In this paper the authors present the first release of the Hybrid Simulated-Emulated Platform (HySEP). This tool integrates the simulation of different access networks obtained through the Network Simulator 3 (ns-3) and a core network emulated through a set of virtual Personal Computers implementing the Differentiated Service (DiffServ) protocol to assure QoS. This choice, on one hand enables the test of several access technologies without the use of real networks, and, on the other hand, enables the transmission of real traffic flows from/to the simulated part of the network.

An important requirement of this platform is the synchronization between the simulated access network, called Simulated Network Segment (SNS), and the emulated core network, called Emulated Network Segment (ENS). It is necessary to assure the simulated traffic forward inside the emulated network and the real traffic management in the simulated network. This requirement justifies the utilization of ns-3 which enables the utilization of virtual devices (i.e. tap and bridge) to connect real and simulated network nodes. To obtain a correct behaviour of the platform the simulation execution in real time mode has to maintain its prefixed duration. In other words, one second inside the simulation must be equal to a real second.

Before using HySEP for research purpose it is necessary to validate its architecture and to define the scenarios that can be represented. Different configurations are considered starting from Long Term Evolution (LTE) and WiFi networks and for each of them the number of nodes are varied. The tests are based on the use of Iperf software: it is able to measure the supported data rate between the nodes inside the simulation and a real remote host. The considered quantities are the obtainable throughput and the execution time: if both these values are equal to the values imposed by the authors during the tool configuration the synchronization between the segments is maintained, and the platform works appropriately. Moreover it is possible to say that the definition and test of HySEP, define a set of applicative scenarios where this tool can be used. The integration of simulated and emulated networks assures a remarkable versatility and wide applicability of the tool to support testing and validation of several different algorithms and procedures for network control and management. For this reason, according to the authors' viewpoint, this platform represents an important innovative contribution. The rest of the paper is organized as follow: Section II briefly describes the tool adopted, ns-3, with particular reference to the models used to simulate LTE and Wi-Fi networks, and the DiffServ protocol. Section III and Section IV represent the core of this paper, respectively describing the HySEP architecture created by the authors of this paper and reporting the results of the validation and scalability tests. Finally, conclusion are discussed in Section V.

II. THE CONSIDERED FRAMEWORK

A. The ns-3 Simulator

Network Simulator 3 (ns-3) is a free, open source discrete event network simulator, written in C++ [1]. It has a modular structure: the lower layers represent the basic infrastructure of the whole simulator, while the upper layers give the capability to add details to the scenario, realizing different network protocols, devices and channels. The helper modules are useful interfaces that help the user to develop base configuration scenarios. These classes are aimed at facilitating the configuration of many elements in the simulation, as for example the *PointToPointHelper*, the *Ipv4AddressHelper*, the *CsmaHelper*, *TapBridgeHelper*, *EpcHelper* and *LteHelper*. Furthermore ns-3 permits the integration of a simulated network into real network test-bed, transmitting/receiving packets from/to it. Consequently a real-time event scheduler inside the simulator is needed to synchronize the ns-3 clock with the clock of the PC where the simulation is executed. As a matter of fact, without this synchronization ns-3 acts as a Discrete Event Simulator (DES) which determines an alteration in the packets forward process between simulated and real networks.

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

A fundamental capability of the LTE module is to create and maintain the radio bearers, which represent the wireless segments between UEs and eNBs of the whole LTE-EPC bearers. Moreover it furnishes several detailed channel models and mobility scenarios, implementing also all the layers of the UE and eNB stack: Radio Resource Control (RRC) [4], Packet Data Control Protocol (PDCP) [5], Radio Link Control (RLC), Medium Access Control (MAC) [6], Physical (PHY).

2) *The Wi-Fi Model inside ns-3*: Thanks to its modular structure, ns-3 enables the simulation of many different networks. As a matter of fact a generic network node can be equipped with a network interface, for example WiFi as described in this subsection, using the opportune module called *WiFiNetDevice* and the *WifiHelper*.

The Wi-Fi network model, defined within the ns-3 framework, is composed of two different parts as described below:

- The physical layer, that includes detailed channel models and different terminal mobility models. It is configurable using the *WifiPhyHelper* API.

- The Medium Access Control (MAC) layer which is divided into MAC low layer and MAC high model. The first one is deputy to manage packets fragmentation, transmission and confirmation and to execute carrier sense procedures. The second one enables the definition of different node functionalities like Access Piont (AP) or non-AP station, called STA. This layer can be easily configured using the API called *WifiMacHelper*.

B. The Differentiated Service Domain

The term Differentiated Service (DiffServ) identifies a network protocol for IP traffic classification [7]. It is largely used to support QoS over several different type of networks. This protocol uses the first six bits, called DiffServ Code Point (DSCP), of the eight bits which compose the Type of Service (ToS) field inside the header IP [8]. It defines three different macro-classes of service with different priority levels: *i)* Expedited Forwarding (EF) [9] *ii)* Assured Forwarding (AF) [10] *iii)* Best Effort (BE). This protocol defines also two different nodes: *i)* Edge Routers (ERs) that are set at the beginning and at the end of the DiffServ domain and *ii)* Core Routers (CRs) that are inside the domain. In particular, the ERs are deputy to assign the DSCP label to each packet mapping the QoS instruction of the traffic flow used inside the incoming network domain [11].

Recently, Linux operative systems offer a wide variety of network traffic control functions located partially in kernel-space and partially in user-space of a PC. In particular, some of them implement the mechanisms required to support the DiffServ architecture [12]. Moreover, the kernel is able to process the IP packets received from the network, forwarding them into the output queue. The Linux Traffic Control (tc) software plays a fundamental role implementing these traffic control operation [13]. In more detail, tc filters the packet and, according to their DSCP value, puts them in a different queue, that contains all the packets belonging to the same traffic class. According to the policy adopted to serve the queues in the output interface it is possible to differentiate the priority of the traffic flows.

III. THE DEVELOPED TOOL

In Figure 1 is represented the Hybrid Simulated-Emulated Platform (HySEP) structure. It is composed of PCs belonging to two different segments: *i)* the Simulated Network Segment (SNS) that include PC1 and *ii)* the Emulated Network Segment (ENS) that include PC2 and PC3. It is worth noticing that it is necessary to configure opportunely the interconnection between SNS and ENS because it has to assure that real traffic flows are forwarded inside the simulated network and the simulated packets are transmitted through the emulated network. As a consequence the synchronization between the two segments represents a key point for the correct working of the tool. In more detail the HySEP structure is described below:

- PC1 contains the ns-3 simulator and the scenario that is developed to simulate the access network. It is connected, using an Ethernet cable to PC2.

- PC2 implements the virtual PCs, created and managed by the VirtualBox software [14], which emulate the transport network. Each virtual machine represents a node of the network connected to the others using virtual network components that emulate the links and supports the application of the DiffServ paradigm. PC2 is connected to PC1 and to PC3 using Ethernet cables.
- PC3 represents the remote real host that communicates with the simulated nodes inside PC1, using the transport network in PC2.

As previously said, the decision to use ns-3 is taken because it permits the integration of simulated networks into real network test-beds as well as into virtual machine environments. To reach this goal the simulator transmits/receives packets from/to real networks. Consequently a real-time scheduler is needed to synchronize the ns-3 clock with the clock of the PC where the simulation is executed, PC1 in this case.

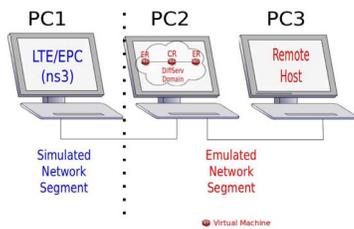


Fig. 1. The Hybrid Simulated-Emulated Platform (HySEP) structure

A. The Simulated Network Segment

The SNS structure is shown in Figure 2: it includes the network, simulated using ns-3 and the virtual network devices tap and bridge, used to connect the SNS and the ENS. Several

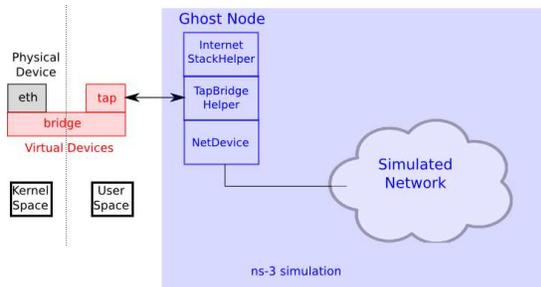


Fig. 2. Simulated Network Segment (SNS) Architecture

networks can be simulated using ns-3, for example LTE-EPC and Wi-Fi as done in this paper, or WiMAX and Bluetooth. For all these examples the steps necessary to connect the simulated nodes with a real network are the same: a particular type of node, called “Ghost Node”, is created inside the simulate scenario. As shown in Figure 2, this node acts as an alias of a tap inside the ns-3 framework. As previously said, the tap is an interface of a virtual bridge connected to the Ethernet interface (*eth*) of PC1. In this way, the traffic generated by the simulated nodes is received by the Ghost Node (i.e., the tap) and subsequently forwarded to *eth* using the virtual bridge. To use tap and bridge utilities it is necessary to appropriately

configure the ns-3 simulation, using the *TapBridgeHelper*, a ns-3 API whose task is to enable the communication with the tap.

Finally it is necessary to disable the Ethernet filters inside the kernel space of PC1 because when those filters are active no packet from ns-3 nodes can pass through the virtual bridge.

B. The Emulated Network Segment (ENS)

The ENS is composed of two PCs as previously said: PC2 which implements the transport network using virtual PC and PC3 that is the remote host which communicates with the simulated mobile nodes.

A transport network may contain several nodes and links and could be difficult and very expensive to build it up. Consequently adopting an emulated test-bed could be a useful simplification, assuring at the same time the capability of handling real traffic flows. The authors propose in this work a network composed of virtual PCs interconnected each others inside a host PC (i.e. PC2), implementing also the DiffServ protocol to support the QoS.

The configuration adopted inside PC2, shown in Figure 3, to emulate the transport network is composed of three virtual machines: two of them are the Edge Routers (ERs) set at the boundary of the network and the other is a Core Router (CR) inside the network. Each virtual PC is also connected to Internet, consequently it is possible to configure and use each of them remotely. The virtual PCs are suitably

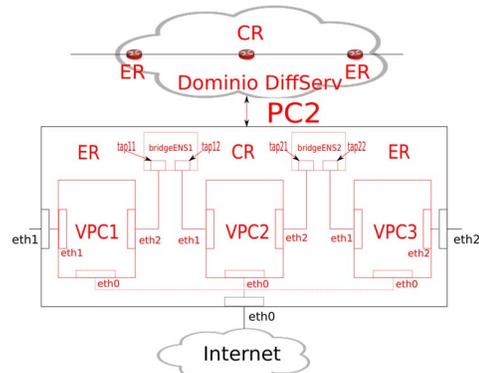


Fig. 3. Transport network architecture composed of virtual PCs

interconnected, using bridge and tap devices. Two different connection configurations are adopted: *i*) the CR is connected to the ERs (i.e., the other virtual PCs) using tap and bridge, *ii*) each ER uses tap and bridge to communicate with the CR on a side and with a Ethernet interface of PC2 on the other. Using tc is possible to add several details to each link in the transportation network such as for example its bandwidth and its propagation delay. Moreover, tc is a software, developed for Linux operative systems, which is used to show and to manipulate traffic control settings managing also the queueing discipline (qdisc), which are the rules that define how the packets are handle in a network interface.

IV. PRELIMINARY TEST RESULTS

The results of the preliminary tests are reported in this section. The authors conduct these tests to validate the HySEP functionality and to define the characteristics of the scenarios that can be simulated through ns-3. As a matter of fact the ns-3 real time modality, necessary to use the simulator inside an emulated test-bed, is computationally heavy. Consequently if the number of nodes in the network increases the computational load increases too and it could overflow the sustainable load of the PC. It is worth noticing that different network models are characterized by different computational loads, so different scenarios are evaluated by the authors.

The next subsection describes the adopted test-bed, the tools used to validate the HySEP and the characteristics of the PC used for the simulations. Indeed different PCs assure different performance in terms of number of nodes which keeps the simulation synchronization.

A. Test-bed characteristic and configuration

The characteristics of the PC used to run ns-3 simulations are reported in Table I to assure the replicability of the results. Two different quantities are measured, using *Iperf* in order to

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

validate HySEP and to define the maximum number of sustainable nodes: *i*) the duration of *Iperf* data transmission and *ii*) the measured throughput between the simulated nodes and the remote host measured using *Iperf* tool [15]. The real time simulation is correctly supported if the *Iperf* transmission time duration and obtained throughput are equal to the predefined values, set by the authors.

Moreover two different scenarios are considered in this work; both are composed of the same ENS described in Section III and differ each other for the simulated scenario characteristics as describe below:

- The single node configuration scenario is composed of a single simulated node directly connected to the Ghost Node within the ns-3 scenario. This scenario is mainly focused on the verification of the connection between SNS and ENS.
- A more realistic scenario considered in this work is composed of two different networks, LTE-EPC and Wi-Fi, connected to the Ghost Node. Both technologies are simulated considering a different numbers of users and a different required throughput in order to find out the maximum number of nodes that can be simulated maintaining the real time.

For both scenarios the simulation duration and *Iperf* connection duration are constantly equal to 70[s] and 60[s] respectively. The *Iperf* tool uses an User Datagram Protocol (UDP) flow between the simulated network and the remote

host. More details about these scenarios are included in the next subsections.

B. Single node configuration

This scenario is composed of a single node that implements *Iperf* in client mode, which is connected to *Iperf* server in the remote host. Table II contains the simulations results in terms of execution time and obtained throughput. It is possible to view that in this configuration the maximum data rate supported is equal to 90 Mbps. Transmitting data rate higher than this value (i.e., 100 Mbps in Table II) causes a simulation time increase. Consequently the *Iperf* client and server are not synchronized, the transmission time is higher and consequently the measured throughput is lower. In practice these results shown the applicability of the developed tool in a single simulated node scenario with a transmit data rate up to 90 Mbps. This limitation is due to the ns-3 simulation computational load that is not supported by PC for 100 Mbps. Obviously different results are obtained using PCs with different characteristics with respect to the ones reported in Table I.

It is worth noting that the throughput values in Table II are slightly higher than the transmission data rate generated by the *Iperf* client. This happens because, even if the simulation is synchronous with respect to the clock of the PC, the synchronization between simulated client and the remote real server *Iperf* shows slight lags. Therefore the transmission length measured by the server is slightly smaller and, consequently, the measured throughput is slightly higher. Despite this small variation in the measured throughput, it possible to determine an upper limit in which the real-time constraint is respected and the instrument can be correctly used.

TABLE II
MEASURED SIMULATION TIME AND OBTAINED THROUGHPUT FOR DIFFERENT TRANSMITTING DATA RATE.

Data Rate	Measured Simulation Time	Obtained Throughput
50 Mbps	59.6 s	50.4 Mbps
60 Mbps	59.6 s	60.8 Mbps
70 Mbps	59.6 s	70.8 Mbps
80 Mbps	59.5 s	80.6 Mbps
90 Mbps	59.7 s	90.5 Mbps
100 Mbps	65.0 s	92.1 Mbps

C. The LTE-EPC and Wi-Fi Networks scenario

The scenario adopted is composed of two access networks, one LTE and one 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 forward them to the core network using the Ghost Node.

Two sets of tests are conducted using this scenario, in the first only the LTE access network with a variable number of UE nodes, within the range [1 - 8], is considered. In more detail this scenario is composed of a UE, called *Iperf* node, that

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]. Similarly, in the second set of tests there is a variable number of Wi-Fi nodes, called STA nodes, whose number is 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]. No transmitting UE nodes are considered in this case.

1) *LTE-EPC case test*: as reported in Section II the LTE-EPC module implemented within ns-3 is very detailed but at the same time it is quite complex and require large computational resources. The results proposed in Figure 4 and Figure 5, which show respectively the *Iperf* transmission time and the obtained throughput measured using *Iperf*, confirm this consideration. In particular a limited number of background nodes can be simulated in real time mode. Considering for example the cases when the *Iperf* node data rate is equal to 2 Mbps: in this case up to 5 nodes (i.e., the *Iperf* node and 4 background nodes) can be simulated. After this number the *Iperf* duration increases and consequently the obtained throughput decreases. Moreover a similar behaviour characterize all the considered *Iperf* node data rates. These results define a set of possible

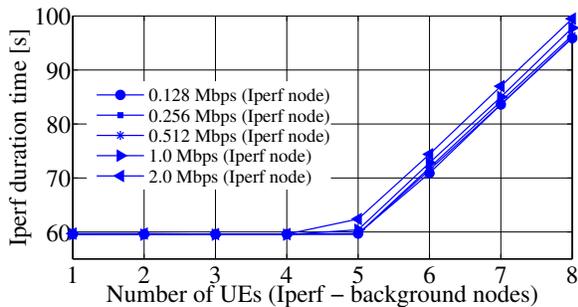


Fig. 4. Iperf transmission time over different transmitting data rates for the Iperf node and over different number of background nodes.

LTE based scenarios in which the HySEP can be adopted. Obviously with a PC with higher performance this set can be enlarged, enabling the simulation of a larger number of UE nodes. It is worth noticing that several scalability tests are already conducted considering the LTE-EPC network but they are limited to non real time scenarios. According to the authors' best knowledge this paper is one of first attempts to study and discuss the LTE-EPC simulation scalability, using ns-3 in real time modality.

2) *Wi-Fi case test*: a simpler module whit respect to the LTE-EPC one is implemented in ns-3 to support the Wi-Fi network simulation. Consequently a larger set of possible scenarios can be simulated overcoming the limitation introduced by the LTE-EPC module. The main goal of these results is to find out the *Iperf* data rate limitations that can be supported in a real time simulation using ns-3. Considering the same number of *Iperf* and background nodes as in the LTE-EPC case Figure 6 and Figure 7 shows that higher *Iperf* data rates can be supported in the real time simulation. Moreover in this case is more evident the link between *Iperf* data rate and

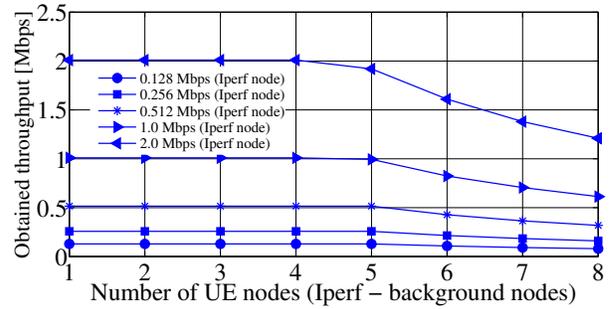


Fig. 5. Obtained throughput over different transmitting data rates for the Iperf node and over different number of background nodes.

the maximum number of nodes that can be simulated. This consideration is confirmed considering two examples: when the *Iperf* data rate is equal to 12 Mbps up to 7 STA nodes (i.e., the *Iperf* node and 6 background nodes) are supported; on contrary when the data rate is equal to 20 Mbps the maximum number of nodes is equal to 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 considered values.

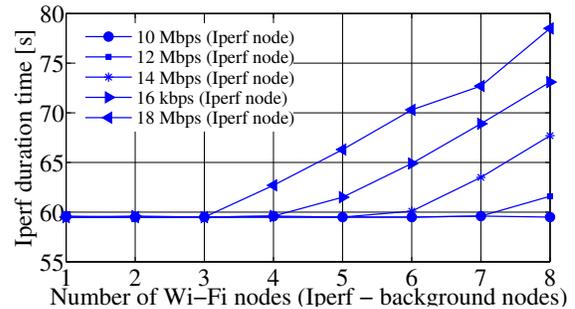


Fig. 6. Iperf transmission time over different transmitting data rates for the Iperf node and over different number of background nodes.

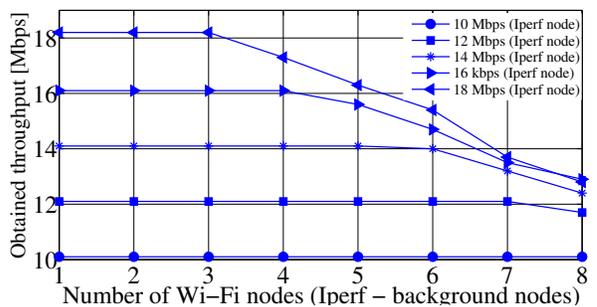


Fig. 7. Obtained throughput over different transmitting data rates for the Iperf node and over different number of background nodes.

V. CONCLUSIONS

This paper is focused on the Hybrid Simulated-Emulated Platform (HySEP) description and its preliminary tests. This

platform has been created by the authors integrating the ns-3 simulator and an emulated DiffServ domain, implemented using three virtual PCs. The most important challenge faced by the authors is the connection of the two aforementioned network segments that has to be properly configured in order to support the communications between simulated and emulated network nodes. In this way it is possible to forward the traffic flows generated by the simulated nodes inside the emulated network and the real traffic flows inside the simulated network. This requirement justifies the utilization of ns-3: it enables the utilization of virtual devices (i.e. tap and bridge) to connect the simulated network with a real network.

The executed tests are aimed at verifying the simulation synchronization maintenance, considering different network scenarios based on different access technologies (i.e. Wi-Fi and LTE) with a variable number of transmitting nodes. Using the software tool *Iperf* the transmission duration and the achievable data rate are measured. The reported numerical results enable to define which are the scenario configurations that respects the real time execution requirement.

REFERENCES

- [1] ns3 Project, "Network simulator 3 (ns-3)." [Online]. Available: <http://www.nsnam.org/>
- [2] M. Olsson, *SAE and the Evolved Packet Core: Driving The Mobile Broadband Revolution*. Academic Press, 2009.
- [3] 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>
- [4] "Evolved universal terrestrial radio access (E-UTRA); radio resource control (RRC); protocol specification," 3GPP TS 36 331, ETSI, 2014.
- [5] "Evolved universal terrestrial radio access (E-UTRA); packet data convergence protocol (PDCP) specification," 3GPP TS 36 323, ETSI, 2013.
- [6] "Evolved universal terrestrial radio access (E-UTRA); medium access control (MAC) protocol specification," 3GPP TS 36.321, ETSI, 2014.
- [7] 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>
- [8] K. Nichols, S. Blake, F. Baker, and D. Black, "Definition of the differentiated services field (ds field) in the ipv4 and ipv6 headers," RFC 2474, The Internet Engineering Task Force (IETF), December 1998. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc2474.txt>
- [9] B. Davie, A. Charny, J. Bennet, K. Benson, J. L. Boudec, W. Courtney, S. Davari, V. Firoiu, and D. Stiliadis, "An expedited forwarding phb (per-hop behavior)," RFC 3246, The Internet Engineering Task Force (IETF), March 2002. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc3246.txt>
- [10] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured forwarding phb group," RFC 2597, The Internet Engineering Task Force (IETF), June 1999. [Online]. Available: <http://www.rfc-editor.org/info/rfc2597.txt>
- [11] K. Nichols and B. Carpenter, "Definition of differentiated services per domain behaviors and rules for their specification," RFC 3086, The Internet Engineering Task Force (IETF), April 2001. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc3086.txt>
- [12] W. Almesberger, J. H. Salim, and A. Kuznetsov, "Differentiated services on linux," in *GLOBECOM*, GLOBECOM, Ed., vol. 1, 1999, pp. 831–836.
- [13] B. Hubert, "Linux advanced routing & traffic control." [Online]. Available: <http://www.lartc.org/>
- [14] Oracle, "Virtualbox." [Online]. Available: <https://www.virtualbox.org/>
- [15] M. Gates, A. Tirumala, J. Ferguson, J. Dugan, F. Qin, K. Gibbs, and J. Estabrook, "Iperf." [Online]. Available: <http://sourceforge.net/projects/iperf/>