Context-aware Heterogeneous Network Performance Analysis: Test-bed Development

Zhenhui Yuan, *Member, IEEE,* John Keeney, Sven van der Meer, *Member, IEEE,* Gabriel Hogan, *Member, IEEE* and Gabriel-Miro Muntean, *Member, IEEE*

Abstract—The rapid growth in mobile and heterogeneous network has led to a massive increase in data traffic adding to the already significant challenges in network performance. Operators already face the problem that thousands of network incidents (e.g. dropped calls or packet delivery failures) remain unresolved due to the technological limitations. It is recognized that the use of network emulation/simulation test-beds are an valid approach to validate the network performance prior to rollout. This paper introduces the development of a context-aware heterogeneous network test-bed which allows network administrators to flexibly control both network topology and network traffic, as well as analyze network performance through output network events. The developed test-bed includes both open-source simulators and low-cost real life equipment. In order to demonstrate the benefit of the test-bed, cell congestion and handover use cases are implemented. SIP services are delivered over various radio network technologies including LTE, UMTS, WiMAX, GSM and WiFi. Various types of network events are provided as output for potential users, i.e. operators, network administrators, etc.

Index Terms—heterogeneous network, LTE, UMTS, WiMAX, WiFi, GSM, multi-RAT, open-source.

I. Introduction

THE significant increase in mobile data traffic places huge pressure on both operators and service providers. Different solutions have been proposed to increase network capacity, including using new radio technologies. However, the emerging new radio access technologies increase the complexity of resource management in the heterogeneous network. In this context, efficient network troubleshooting and management solutions are essential to network operators, subscribers and service providers [1] [2]. Network troubleshooting and management techniques mainly focus on identifying and diagnosing communication system bottlenecks. For instance, mobile data can be offloaded from high congested networks to low congested networks (e.g. from WiFi to LTE) to enhance the link resource utilization [3]. Additionally, in order to efficiently use the radio resources, 4G/WiFi networks can be planned to handle the majority of

internet data traffic while 3G/2G technologies are configured to mainly provide voice services including SIP [4].

System failures and problems are critical factors in analyzing and troubleshooting heterogeneous network performance. For instance, many data mining techniques (e.g. [5] [6]) have been employed to analyze monitored network events in order to detect network faults and also predict the network performance. In this context, accurate network activity diagnostics rely on large amount of network events collected. In general, collecting network events in practical heterogeneous network are expensive and time consuming. There is a need for a low-cost flexible heterogeneous network test-bed which is able to generate large number of practical network events.

This paper presents a software defined heterogeneous network test-bed which has implemented both virtual SIP services and various radio access technologies. The test-bed enables network administrators to have programmable control of both network hardware and network traffic. The experiences and lessons learnt from building the test-bed are summarized which are useful for repeating the test-bed if wished so. Two common telecom use cases, cell congestion and handover, are deployed in the developed test-bed in order to demonstrate its usage. A generic format of network events is given as an example. These events are beneficial for network performance optimization including for example data mining, recommender system, etc.

This paper is organized as follows. Section II introduces the test-bed implementation, experiences and lessons. Section III describes the use cases deployment and shows the output network events. Section IV concludes the paper.

II. Test-bed Implementation, Experience and Lessons

Fig. 1 shows the overall architecture of the developed software-defined heterogeneous network test-bed which consists of three separate test-beds including radio technologies such as LTE, UMTS, WiMAX, WiFi and GSM. The LTE test-bed is established using the LTE-Sim simulator [7] and the UMTS/WiMAX/WiFi test-bed is setup using the NS2 [8] simulator. OpenBTS [9] and a Universal Software Radio Peripheral (USRP) device[1] are used to build a real lab GSM test-bed. In this context, the 2G technology is integrated

[1] Ettus Research, http://www.ettus.org
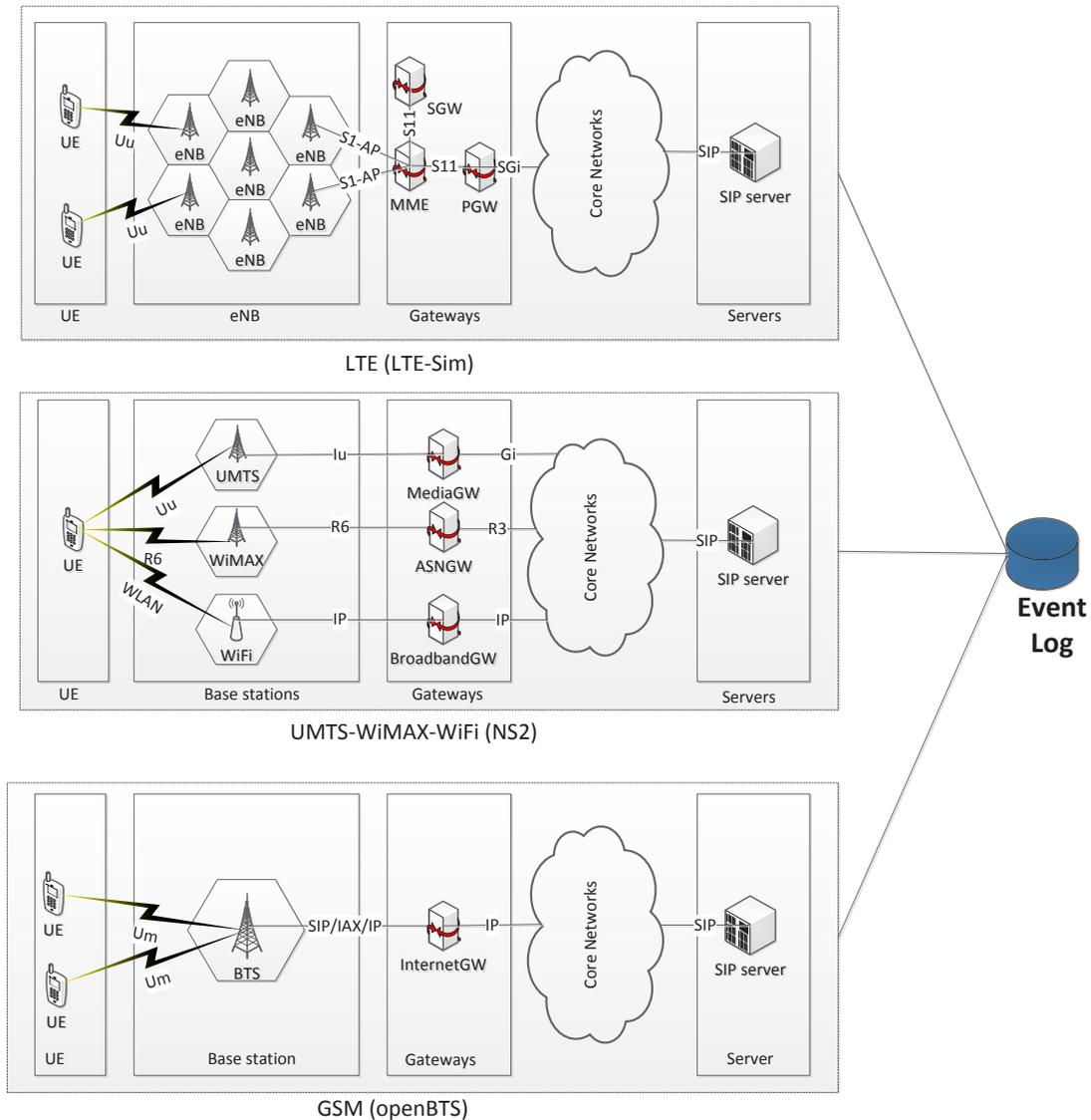
Fig. 1 Overview of the software defined heterogeneous network

into the IP-based test-bed as each test-bed has configured a SIP server to provide voice service. Details about the three test-beds implementation are introduced as follows.

A. LTE Test-bed

LTE-Sim [7] is designed as an event-driven simulator written in C++. It is open-source and includes both the Evolved Universal Terrestrial Radio Access (E-UTRAN) and the Evolved Packet System (EPS). In particular, LTE-Sim provides key features such as flow management (i.e. radio bearers, QoS requirements, etc), feedback control, multiple types of downlink scheduler, etc. LTE-Sim has four main components: Simulator, NetworkManager, FlowsManager, and FrameManager. Each of the components is related to a dedicated class. A singleton design pattern is used to ensure that only one instance of these classes are initiated. LTE-Sim is installed in the Ubuntu 12.10[2] linux system.

---

[2] Ubuntu, http://www.ubuntu.com

Table 1 LTE-Sim Test-bed Configuration for the Handover Scenario

Carrier frequency	2GHz
Radio Access Mode	FDD
Uplink bandwidth	5MHz (25RBs)
Downlink bandwidth	5MHz (25RBs)
Downlink scheduler	Proportional Fair (PF)
Propagation Loss Model	Marco Cell –Urban Areas
Cell radius	1km
Speed of UEs	30km/h
Application	ITU-T G.729

Table 1 shows the parameter configuration of LTE-Sim. There are two types of radio access modes supported: FDD and TDD. In this test-bed, FDD is selected. LTE signal operates at 2000MHz and downlink and uplink bandwidth are both set as 5MHz. This bandwidth is selected in order to allow practical simulation process. Larger bandwidth will increase the simulation complexity. Proportional Fair scheduler is used as

Table 2
Component configuration in NS2

Component	WiFi	UMTS	WiMAX
Carrier frequency	2.4GHz	2GHz	3.5GHz
Bandwidth (downlink)	1, 2, 5.5, 11 (Mbps)	384Kbps	1.25MHz-20 MHz
Coverage	160m	1000m	1000m
Propagation model	Two Ray Ground	Two Ray Ground	Two Ray Ground
Antenna type	Omni Antenna	Omni Antenna	Omni Antenna
Interface queue length	50packets	5000packets	5000packets
Routing protocol	NOAH	Adhoc routing	Adhoc routing
UE speed	1m/s		
Link Queue Type	Droptail		
Number of UE	1		
VoIP codec	ITU-T G.729		

the downlink scheduler. Cell radius is set as 1km and the propagation loss model uses Marco Cell-Urban Areas. UE receives ITU-T G.729 [10] VoIP service and moves in the cell at the speed of 30km/h.

By default, LTE-Sim prints out basic information of packet transmission only. LTE-Sim needs to be modified in order to output more comprehensive network events such as data and control flows at different layers (RRC, PDCP, RLC, MAC, PHY, etc), resource blocks allocation process, SINR and CQI information, handover, etc. The file located in "/*home/lte-sim-version/CONFIG/global_config*" needed to be updated.

B. UMTS/WiMAX/WiFi Test-bed

The original NS2 [8] simulator does not support the multiple interfaces of a mobile node (MN). In order to develop a heterogeneous network, NIST[3] introduces "supernode", in which an individual node is configured with a single radio interface and nodes are linked to each other. The NIST package includes several enhancements to the original NS2 in order to support seamless communications in heterogeneous networks. It allows the simulation of WiFi, UMTS and WiMAX technologies and incorporates an IEEE 802.21 [11] based handover mechanism. The IEEE 802.21 MIH module provides seamless UMTS/WiFi/WiMAX handover based on the Make-Before-Break (MBB) algorithm.

NS2 with NIST package is installed on Ubuntu 12.10. Table 2 lists the configuration of the simulations. A single UDP-based VoIP flow is delivered to the mobile node. Constant Bit Rate (CBR) VoIP traffic is generated using ITU-T Rec. G.729, with payloads of 160 bytes/packet. The bit-rate of CBR was

set to 128kbps representing an inter packet transmission interval of 10ms. In the UMTS network simulation, acknowledgement mode (AM) radio link control protocol (RLC) is used. In general, unacknowledgement mode (UM) and transparent mode (TM) are used for RRC connection update and cell update process. For the wireless environment, the No Ad-Hoc (NOAH) patch[4] is implemented in the NS-2 simulator in order to allow direct communication between mobile users and the base station only.

The NIST package for supporting 802.21 MIH framework has to be installed in earlier versions of NS2 (e.g. NS-2.28 and NS-2.29). There might be many compiling errors and warnings when installing NIST package in Ubuntu 12.10 due to system incompatibility. These errors need to be fixed according to the error messages shown.

C. GSM Test-bed

This section presents a real GSM network test-bed using an open-source platform, OpenBTS [9]. The latter provides GSM base station functionalities and allows any GSM-compatible mobile phones to be used as SIP endpoints. The public release of OpenBTS implements the lower three layers of the industry-standard GSM protocol stack. It is written in C++ and used as free software under the GNU Affero General Public License (AGPL). OpenBTS uses a SIP switch or PBX to perform the call control functions that would normally be performed by the mobile switching center in a conventional GSM network. The OpenBTS framework is installed on Ubuntu 12.10 system in a desktop (Dell Studio 540s, Intel core 2 CPU Q8300@2.5GHz, 4GB RAM). The SIP service is provided by Asterisk v11.0[5]. Fig. 2 presents the test-bed for the GSM network. The GSM frequency is set to 900MHz. As indicated in Fig. 2 (a), two mobile phones (caller and callee) communicate via the GSM spectrum generated by the Universal Software Radio Peripheral (USRP) N210. The USRP provides computer-hosted software radios and is widely used by research labs and academic institutions due to its low cost. GNURadio v3.6.5 [6] is used to connect USRP and OpenBTS.

GNU radio is open-source software that provides signal processing blocks to implement software radios. GNU radio can be used with external RF hardware to generate software-defined radios or without hardware in a simulation environment. GNU radio applications are primarily written using Python and C++. The USRP in this test-bed has output power up to 15dBm and is equipped with a SBX daughter board[7], which supports 400 MHz to 4.4 GHz frequency. Two VERT900 TX/RX antennas[8] are installed on the USRP N210 device to support 824 to 960 MHz and 1710 to 1990 MHz Quad-band. USRP requires a gigabyte switch to connect to the

---

[3] NIST package,
http://http://www.nist.gov/itl/antd/emntg/ssm_seamlessandsecure.cfm

[4] NOAH package, http://icapeople.epfl.ch/widmer/uwb/ns-2/noah/
[5] Asterisk, http://www.asterisk.org
[6] GNURadio, http://www.gnuradio.org
[7] Ettus Research, http://www.ettus.com
[8] Vert 900, https://www.ettus.com/product/details/VERT900
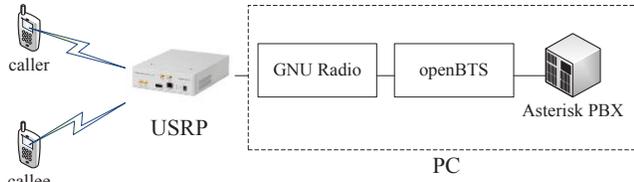
(a) Topology of the GSM test-bed
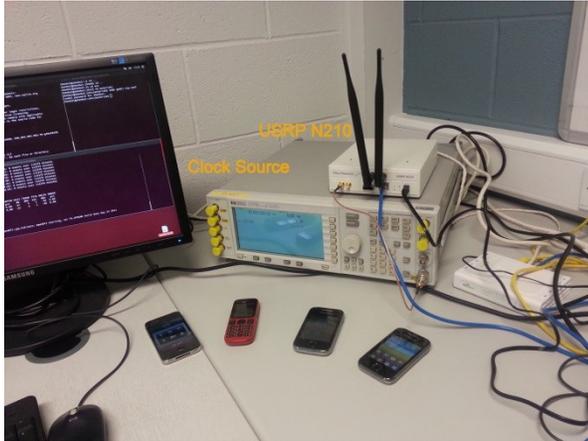
(b) Screenshot of test-bed

(c) Connect to the Public Land-Mobile Network (PLMN) on mobile

Fig. 2 GSM test-bed topology

PC. A signal generator, HP E4437B, outputs a reference signal with 52MHz and 5dBm and is required to act as external clock source for the USRP. This provides compatibility between the USRP N210 provided 64MHz default clock and the GSM specification 52MHz clock. Therefore, the clock must be synchronized. Four GSM 900MHz-supported mobile handsets (one iPhone 4s, two Samsung Galaxy Y, and one Nokia 6021) are used to connect to the GSM station. Fig. 2(c) indicates that a mobile has been successfully registered to the OpenBTS network (operator name is Test PLMN 1-1).

Several key lessons obtained are summarized as follows. 1) USRP N210 Clock must be synchronized; otherwise mobile handsets will never detect the GSM base station. Traditionally,

Table 3 An example of output network events in cell congestion scenario (LTE-Sim)

Time	Event type	Event ID	Cell ID	Number of UE
1373029409	UEJoin	2181	99	1
1373029411	UEJoin	465	21	1
1373029413	UELeft	464	21	0
1373029417	UEJoin	1983	90	1
1373030993	CellFull	2224	23	10

many solutions on synchronizing the clock are based on soldering the mother board. This paper shows a simple solution which is easy to use. 2) It is common that the mobile handsets cannot immediately find the OpenBTS base station. In this case, multiple manual search are required in order to succeed. 3) As the system has one daughterboard only, it can support large amounts of SMS text messages, but it can only handle two voice calls between four mobile phones. Three SIP calls (between six mobile handsets) will cause a system failure. 4) The phone registration information (e.g. extension number, caller id, port number, etc) is saved in a Sqlite database (/etc/asterisk/*sqlite3.db*) under the table *sip_buddies*. 5) It is strongly recommended after the successful installation to prevent the optional update of Ubuntu. As this can cause serious software conflicts and cause the OpenBTS system to crash.

III.   TELECOM USE CASES AND EVENT GENERATION

The cell congestion use case is implemented in LTE-Sim and OpenBTS test-beds, separately. Handover between UMTS/WiMAX/WiFi is deployed in NS-2.

A. Cell Congestion in LTE

ITU-R [12] requires an LTE systems to support more than 40 voice calls per MHz bandwidth and cell. Therefore, more than 200 VoIP calls can be supported by a cell configured at 5 MHz bandwidth. The cell congestion scenario is setup in the test-bed configured based on Table 1. 100 cells and 100 eNBs are created. Each cell is configured with a UE capacity equals 10. A UE randomly joins one of the 100 cells if a cell is not full. Additionally, a UE can randomly leave a cell if there is at least one UE in that cell. UEs will be rejected from joining a cell if the cell has reached its capacity. The interval between two join events is determined using two values *shortJoinIntervel* and *longJoinInterval*, which are set to 1 second and 3 second, respectively. The interval between two left events is computed based on the sum value of *shortJoinIntervel* and *longJoinInterval*, which is 4 second. All the UEs are static and uniformly distributed in each cell. A remote host is created and transmits ITU-T G.729 VoIP traffic to UEs. The VoIP data rate is 8Kbps and the inter-packet interval is 20ms. There is no specific limitation on the number of cells and UEs supported by LTE-Sim. In general, the increasing number of network nodes increases the hardware processing cost.

Three types of events are defined:

1) *UEJoin* - an UE joins a cell
2) *UELeft* - an UE left a cell
3) *CellFull* - a cell has reached its capacity

These network events are recorded into log files including five fields: 1) *time stamp*; 2) *event type*; 3) *event ID*; 4) *cell ID*; 5) *the active number of UE in the current cell*. Time stamps are the current system time in seconds. Event type can be *UEJoin*, *UELeft*, or *CellFull*. *Cell ID* is the identifier of the cell a UE is attaching to. *Event ID* is unique for each *cell ID*. The *event ID* is calculated depending on the use case and network administrator. Computation of *event ID* is not shown in this paper. An example of the output events log file is shown in Table 3. The values of *event ID* are the same for the same cell and the same event type and changes according to the active number of UEs in that cell. Stress test shows that it takes around 11 hours to generate 30,000 events in our test-bed (about 0.76 event/second in real time).

B. Cell Congestion in GSM

As shown in Fig. 2, four mobile phones were connected to the openBTS-based GSM base station. The configuration file needs to be properly setup in Asterisk in order to make SIP calls between the mobile handsets. Table 4 shows the SIP service configuration in Asterisk.

In this test-bed, only two SIP calls (four mobile handsets) are supported since the USRP is equipped with a single mother board. The cell congestion scenario is implemented by increasing the number of mobile handset from one to four and initiating two SIP calls between the four mobile handsets. The test includes two steps. First, connect two mobile phones to the GSM network and establish voice call between two mobile phones. Second, add another SIP call (another pair of mobile handsets) to the GSM network. It is noticed that the second voice call last around 30 seconds and then is dropped. Meanwhile, the first call is also dropped.

The call drop events are caused by high interference levels of the GSM channel. Syslog file (*var/log/syslog*) is investigated showing that two error messages are output: "*radio link failure, dropped call*" and "*Pager blocking for signal*". This means that the mobile handsets cannot be paged into the base station due to radio link failure. OpenBTS also provides command interface to monitor the physical status of active channels by running the command *OpenBTSCLI> chans* in the OpenBTS control interface. Table 5 shows the events information transformed from the syslog file. Two types of event are defined: SIP call start and SIP drop. *UPFER* is the uplink frame erasure rate (in percentage). *DNBER* is downlink bit error rate (in percentage). *Phone ID* refers to the identity of the mobile handsets, ranging from 1 to 4. *Cell ID* is the identity of the GSM cell, which always equals 1 in this test-bed. It is noticed that, when there is one SIP call only, downlink bit error rate and uplink frame error rate are zero indicating perfect channel condition. When two SIP calls exist, the downlink bit error rate and uplink frame error rate increases significantly indicating worse channel quality.

Table 4. SIP configuration in Asterisk

etc/asterisk/sip.conf
[iphone] username=iphone ; usually the same with the section title callerid=1111 canreinvite=no ;allow RTP voice traffic go through Asterisk type=friend context=sip-local host=dynamic disallow=all ;use disallow=all before using allow=.. allow=gsm allow=g729 qualify=no nat=no [Sumsung1] username=Sumsung1 callerid=2222 canreinvite=no type=friend context=sip-local host=dynamic disallow=all allow=gsm allow=g729 qualify=no nat=no [Sumsung2] username=Sumsung2 callerid=3333 canreinvite=no type=friend context=sip-local host=dynamic disallow=all allow=gsm allow=g729 qualify=no nat=no [Nokia] username=Nokia callerid=4444 canreinvite=no type=friend context=sip-local host=dynamic disallow=all allow=gsm allow=g729 qualify=no nat=no
etc/asterisk/extension.conf
exten => 1111,1,Dial(SIP/IMSIXXX@127.0.0.1:5062,10,t) exten => 2222,1,Dial(SIP/IMSIXXX@127.0.0.1:5062,10,t) exten => 3333,1,Dial(SIP/IMSIXXX@127.0.0.1:5062,10,t) exten => 4444,1,Dial(SIP/IMSIXXX@127.0.0.1:5062,10,t)

Table 5 An example of output network events in cell congestion scenario (OpenBTS)

Time	Event type	DNBER	UPFER	Phone ID	Cell ID
Jun 4 14:05:52	SIP call start	0.00	0.00	1	1
Jun 4 14:06:11	SIP call start	0.00	0.00	2	1
Jun 4 14:06:33	SIP call start	1.13	4.75	3	1
Jun 4 14:06:56	SIP call start	1.13	4.75	4	1
Jun 4 14:07:25	SIP Call Drop	1.13	4.75	3	1
Jun 4 14:07:25	SIP Call Drop	1.13	4.75	4	1
Jun 4 14:07:27	SIP Call Drop	0.57	0.01	1	1
Jun 4 14:07:27	SIP Call Drop	0.57	0.03	2	1

Table 6 An example of output network events in handover scenario (NS2)

Time	Network Technology	Cell ID
256	UMTS	1
257	UMTS	1
401	WiFi	2
403	WiFi	2
601	WiMAX	3
603	WiMAX	3

C. Handover in UMTS/WiMAX/WiFi

The need for traffic offload and handover between UMTS, WiMAX and WiFi networks is increasing significantly in order to provide higher Quality of Service levels and lower costs. This section presents a handover scenario between UMTS, WiMAX and WiFi networks using the test-bed configured in section II. A UE is created and moves across UMTS, WiMAX and WiFi networks at the speed of 1m/s. The UE is configured with three radio interfaces (i.e. UMTS, WiMAX and WiFi) and will automatically connect to the closest network base station when it moves. Unique network identity number is given to the three networks: ID=1 for UMTS, ID=2 for WiFi and ID=3 for WiMAX. An example of the network events generated from this handover scenario is shown in Table 6. The event log includes three fields: time, network technology and cell ID (network identity number).

IV. CONCLUSIONS

This paper discusses the implementation experience and lessons learnt from building a low-cost heterogeneous network test-bed. Five types of radio technologies are included: LTE, UMTS, WiMAX, WiFi, and GSM. To demonstrate the usage of the developed test-bed, two common telecom use cases (i.e. cell congestion and handover) are implemented. Examples of the output network events are presented and can be utilized by the network administrator for analyzing the network performance. Future works will focus on the multi-radio access technology (multi-RAT) handover between different simulators and real test-beds.

REFERENCES

[1] W. Zhang, M. Kamgarpour, D. Sun, and C. J. Tomlin, "A Hierarchical Flight Planning Framework for Air Traffic Management," Proceedings of the IEEE, vol. 100, no. 1, pp. 179–194, 2012.

[2] M. Charalambides, G. Pavlou, P. Flegkas, N. Wang, and D. Tuncer, "Managing the future internet through intelligent in-network substrates," *IEEE Network*, vol. 25, no. 6, pp. 34–40, 2011.

[3] S. Chen, Z. Yuan, and G.-M. Muntean,"An Energy-aware Multipath-TCP-based Content Delivery Scheme in Heterogeneous Wireless Networks", IEEE Wireless Communications and Networking Conference (WCNC), Shanghai, China, Apr. 2013, pp.1291-1296.

[4] J. Cartmell, J. McNally, and B. Balazinski, "Local selected IP Traffic Offload Reducing traffic congestion within the mobile core network," in 2013 IEEE Consumer Communications and Networking Conference (CCNC), 2013, pp. 809–812.

[5] A. Gainaru, F. Cappello, M. Snir, and W. Kramer, "Fault prediction under the microscope: A closer look into HPC systems," IEEE International Conference in High Performance Computing, Networking, Storage and Analysis (SC), 2012, pp. 1–11.

[6] T. Lindhorst, G. Lukas, E. Nett, and M. Mock, "Data-Mining-Based Link Failure Detection for Wireless Mesh Networks," 29th IEEE Symposium on Reliable Distributed Systems, 2010, pp. 353–357.

[7] LTE-Sim, available at https://lte-sim.googlecode.com/files/lte-sim-r5.zip

[8] Network Simulator 2, available at http://www.isi.edu/nsnam/ns/.

[9] OpenBTS, available at http://www.openbts.org.

[10] ITU-T Rec. G.729, "Coding of Speech at 8 Kbit/s using Conjugate Structure Algebraic Code Excited Linear Prediction (CS-ACELP)," Mar. 1996.

[11] IEEE 802.21-2008, Standard for Local and Metropolitan Area Networks-Part 21: Media Independent Handover Services, IEEE Computer Society, January 2009.

[12] "Guidelines for Evaluation of Radio Interface Technologies for IMTAdvanced," ITU-R, Tech. Rep. ITU-R M.2135, November 2008.