Measurement Based Mobility Emulation Platform for Next Generation Wireless Networks*

Parishad Karimi, Shreyasee Mukherjee, Jakub Kolodziejski, Ivan Seskar, and Dipankar Raychaudhuri

WINLAB, Rutgers University, {*parishad, shreya, jkol, seskar, ray*}*@winlab.rutgers.edu*

***Abstract*—The exponential growth of the number of mobile users and their traffic generation makes mobility an intrinsic element to be considered in the design of new protocols and network architecture. In order for future Internet protocols to meet the requirements of advanced mobility services, they should be designed with mobility considerations for various signal propagation scenarios, i.e. various wireless technologies operating in different frequency bands, diverse mobility patterns in different environments ranging from vehicular mobility to pedestrian in urban and suburban scenarios. In this paper we propose "EmuWNet", an Emulation framework for Wireless Networks, using real-world signal propagation measurements in a controlled testbed environment. This framework will enable network architecture and protocol designers to evaluate the performance of novel Internet protocols in a controlled environment for various mobility scenarios and perform fair comparisons with existing protocols for the same emulated mobility scenarios. In this paper we will demonstrate this functionality by showing how a set of signal measurement traces, whether collected from real-world experimentation or simulation can be replayed using the emulation framework. As a demonstrative example we input real-world signal strength (RSSI) traces for Wi-Fi and LTE from a smartphone into the framework and conduct mobility experiments on the ORBIT testbed, comparing a future Internet non-IP based protocol with TCP/IP based communication.**

I. Introduction

The number of mobile devices in the Internet has been growing exponentially over the past decade. The Cisco VNI Global Mobile Data Traffic Forecast 2021 [1] predicts that mobile data traffic alone will account for 48.3 exabytes/month by 2021, growing twice as fast as fixed IP. However, the core building blocks of the Internet are still geared towards static connected hosts. It is therefore imperative to propose and evaluate newer protocols that cater to mobile users with varied throughput, latency and connectivity requirements. For example, there has been a body of protocols proposed recently for 5G [2], [3], LTE in unlicensed band (LTE-U) [4], [5], highly-mobile vehicular connectivity [6], [7], content delivery [8], [9] as well as a parallel effort in alternative architectures for mobile hosts that offer non-IP based protocol solutions [10], [11]. However, without changing the entire ecosystem, it is challenging to evaluate such proposed protocols and techniques through real-world experimentation that is meaningful as well as repeatable, especially in the research community with fewer resources or access to the network infrastructure at-scale.

*Research supported under NSF CI-EN: ORBIT GEN 3 grant, CNS-1513110 and NSF FIA-NP grant, CNS-1345295

Keeping in mind the above limitation, our aim is to provide an emulation framework, with the following features for mobile wireless experimentation: (i) repeatability of the experiments, (ii) reconfigurability and programmability of the nodes, (iii) automation and remote management, and, (iv) realism in different layers of the protocol stack. There have been multiple projects aimed at emulating wireless scenarios through different strategies, mostly using wired networks. One class of emulator frameworks used a hybrid approach of running real protocols on top of simulated physical layer [12], [13]. Emulators like [14] use virtualization of multiple machines with different wired connectivity configurations to emulate wireless environment. Similarly, in [15], [16] the authors proposed frameworks to emulate 802.11 ad-hoc networks. Unlike these frameworks, EmuWNet is designed on the ORBIT testbed [17] which is an open-access wireless testbed with fully programmable nodes and software defined radios (SDRs). It takes as input real-world or simulation-based signal strength measurements [18] for a characteristic end-host, maps these measurements to operation range of wireless links based on the technology aimed to use (such as Wi-Fi, LTE) and programs the attenuation of wireless links in a dynamic manner to emulate mobility. Being fully programmable with SDRs, EmuWNet can be utilized to evaluate any existing or proposed protocols. The contributions of this work are as follows:

- A novel framework, EmuWNet, is proposed, that emulates wireless link mobility for real-world experimentation.
- EmuWNet runs on ORBIT with fully-programmable nodes and SDRs that can be utilized to perform repeatable experimentation of any software implementation of protocols at all layers of the architecture, from physical to application layer.
- A representative use-case of comparing an existing transport protocol with a future Internet architecture protocol on a mobile wireless device is presented.

II. System Architecture

EmuWNet is an emulation framework built on top of the ORBIT testbed. In this section we present the hardware and software capabilities of it.

A. Hardware

EmuWNet runs on a sandbox [19] on ORBIT with RF controllable attenuators. It consists of 9 ORBIT nodes (general purpose computers) in RF shielded boxes. Each of the nodes is located inside RF enclosure providing 80 dB of RF isolation. All of the nodes have Intel Wi-Fi/WiMAX cards. In addition to these wireless interfaces, four of the nodes have Ettus USRP B205mini-i SDRs [20] to experiment with programmable RF physical layer, while three others have commercial USB LTE client adapters to permit LTE based experiments. Two NEC WiMAX basestations and two Airspan LTE basestations are deployed in the sandbox for WiMAX and LTE experiments. All of the nodes are also equipped with Atheros AR928X wireless network adapters (2.4/5 GHz) [21] to perform Wi-Fi based experiments. The RF connections of the enclosures housing the nodes as well as the antenna connections of the LTE and WiMAX base stations are all connected to the RF transceiver test system, as shown in Fig. 1, which is composed of an array of RF switches and two attenuator matrices. The tight control of attenuation between each pair of nodes is enabled by the JWF attenuator matrix [22] and a custom attenuator matrix. The JWF test system ("RF Matrix 1" in Fig. 1) allows for 0 to 63 dB of attenuation between each pair of nodes. The custom 5 port attenuator matrix ("RF Matrix 2" in Fig. 1) is based around a series of digital step attenuators controlled via an Ethernet connected Raspberry Pi. Two attenuators in series are installed per port pair giving 0 to 63.5 dB of attenuation range. Therefore, depending on the experimenter requirements, the sandbox can be setup with two independent topologies of 5 or 9 nodes consisting multiple LTE or WiMAX basestaions, Wi-Fi APs and one or more clients. This makes it suitable for controlled tests of handoffs, interfering devices, and multi-homing across different access technologies. There is an additional tenth node which is unshielded and has no radios. It can be utilized for network services such as LTE mobility management (MME), a webserver, a network cache or a network router. Fig. 1 provides and overview of the overall sandbox hardware setup.

B. Software

The ORBIT management framework (OMF) instrumentation service [23] provides user control of the RF interconnect topology and individual link attenuations in the sandbox. Through a simple REST API, a user can change the system's parameters at will. At the simplest level, the service provides the ability for certain nodes to be replaced with WiMAX or LTE basestations and permits the user to set the attenuation values (0 to 63dB) between port pairs on the JFW attenuator matrix. Actual attenuations between devices are greater due to specifics of the system architecture. Interconnect topologies can be defined by setting unconnected link attenuations to their maximum possible value (63dB). A subset of the ISGCI Smallgraphs [24] topologies is preloaded into the system.

More complex control of the attenuator matrix is also possible through the OMF instrumentation service. For example, the attenuations of multiple port pairs can be synchronously set with a single call to the service. This allows for a user's experiment script to more accurately emulate mobility of clients between basestations. Additionally, the service permits attenuations to be scheduled with respect to UNIX epoch

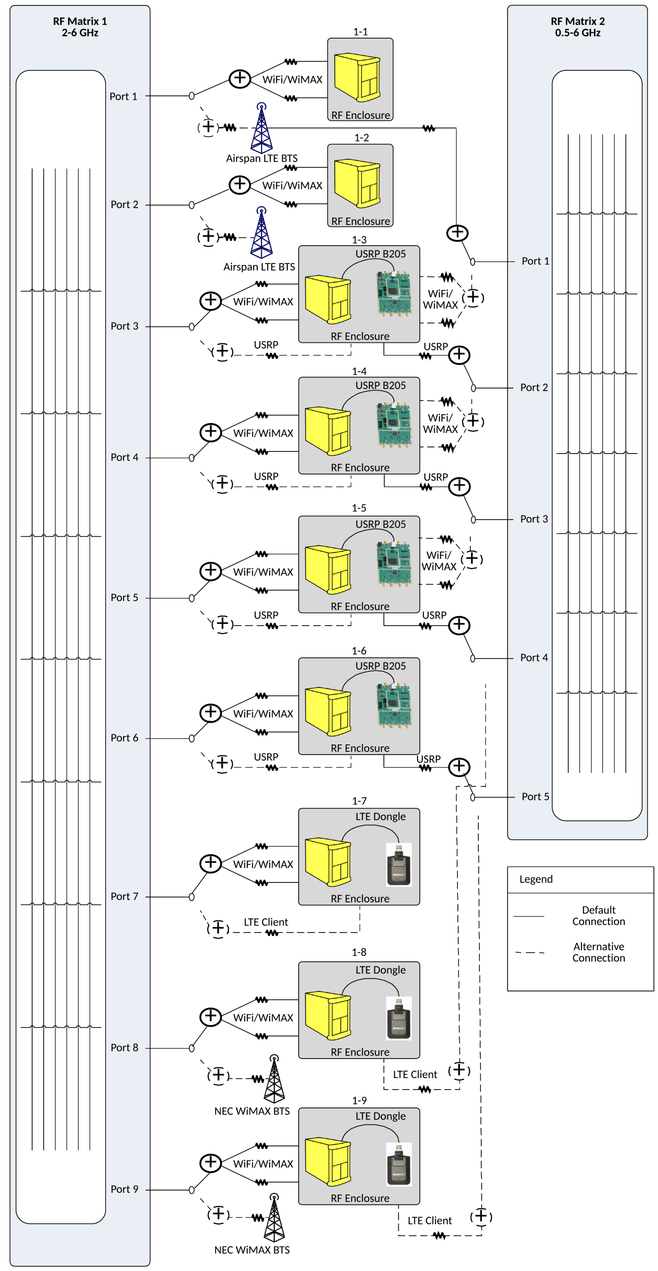

Fig. 1. Overview of the sandbox with RF attenuation matrices in ORBIT

time to facilitate synchronization between the experiment and changes in link attenuation. Given the limitation of 1dB resolution in the attenuator matrix, this method combined with NTP time synchronization between devices is sufficient for most experiments. EmuWNet is built on top of these described capabilities of the OMF instrumentation service to allow for "replay" of discrete time based traces. At a fundamental level, EmuWNet automatically steps through a given set of time indexed port pair attenuator values. This automatic replay can be scheduled to begin just like the individual attenuator setting commands described earlier to facilitate synchronous execution of the experiment. Alone, this capability is no different

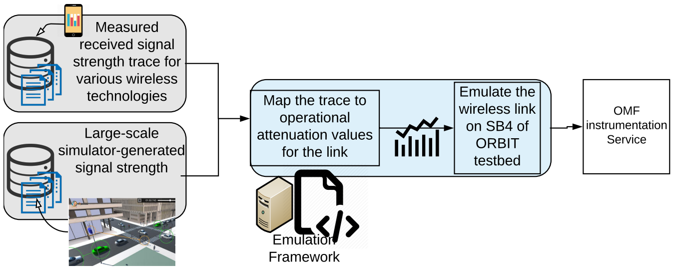

Fig. 2. Overview of the emulation framework on ORBIT

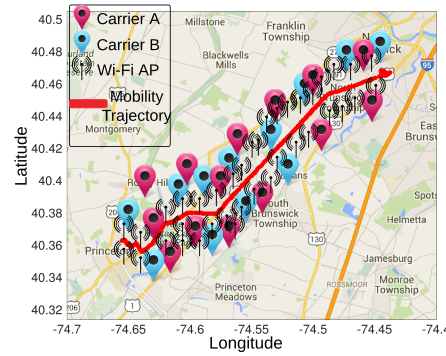

Fig. 3. The route on which the measurements have been conducted along with estimated location of a subset of LTE base stations and Wi-Fi APs

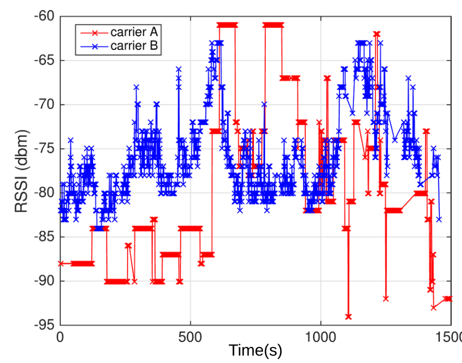

Fig. 4. Example of signal strength measurements, for two different cellular carriers

than what a user's experiment script can achieve. However, the framework is expanded to accept different user provided datasets to produce the required time indexed attenuator traces. For example, EmuWNet can generate a mapping table to translate a user provided RSSI trace (e.g. cellular RSSI data collected in the field or via realistic simulations) into a time indexed attenuation trace that the service can then replay. Taking this a step further, a mobility trace with a set of known basestation coordinates can be used to calculate RSSI values that the mapping module can then translate into a dataset usable by the service. This simplifies running the experiments, since trace data processing can be offloaded to the service and allows for repeatable experiments using not only real world data, but also purely simulation-generated data, as well as any combination thereof. Fig. 2 shows the system design of the framework.

III. Wireless Link Emulation

In order to design realistic emulation of real-world mobility in a testbed setup, we first looked into sample measurements and collected fine-grained wireless connectivity data using two smartphones. The details of these measurements are described in Sec. III-A. These measurements provide us with insights on different schemes to emulate real-world wireless links. In Sec. III-B, we discuss how various measurement categories can be replayed based on possible mappings. Some examples of these possible mappings are depicted using our sample measurements.

A. Measurement Procedure

We developed "NetMobilityTracker", an application using the android telephony API in order to capture wireless connectivity information from smartphones. This application performs both passive and active measurements in the background. For passive measurements, it periodically logs the following information: (i) the current timestamp, (ii) cell-ID and local area code (LAC) of the cellular basestation the phone is connected to, (iii) the cellular technology used, such as LTE, UMTS, HSPA+, 3G, etc., (iv) the received signal strength indicator (RSSI) of the cellular connection, (v) the service set identifiers (SSIDs) of available Wi-Fi access points, and, (vi) the RSSI of each of the Wi-Fi APs. For active measurements (an optional feature), the app periodically tries to download the ORBIT webpage and logs the latency and throughput it observes. NetMobilityTracker was installed on two phones (Samsung Galaxy S3 and S5) with different cellular carriers and they monitored the wireless channel conditions over a predefined vehicular commuter route (see Fig. 3) twice a day (morning and night) for 5 months. The two phones were always co-located and collected the measurements simultaneously to record spatially correlated channel conditions.

B. Real-World Trace Emulation

• **RSSI-based Emulation:** According to the architecture of our system and framework explained previously, the most straight-forward way of replaying traces is having access to signal strength (RSSI) measurements. It should be taken into account that due to different receiver sensitivity (which is governed by noise floor for different devices) and different technologies, the raw signal strength data cannot be directly injected into the emulation environment. In order to address this issue, the measurements were mapped to the operating range of our system based on the technology used. This is shown in Fig. 4, for a sample piece of signal strength measurement we conducted for two different cellular carriers. As can be seen, the range of the reported RSSI varies with the phone, and the carrier it is connected to, and this should be noted when replaying these traces.

• **Download Rate-based Emulation:** One of the most common ways to emulate real-world wireless systems is to replay download traces obtained from periodic retrieval of constant-sized objects through the wireless medium. Compared with

RSSI-based emulation, download rate-based emulation takes into account additional information about the system, for example, the load on the system (number of concurrent UEs being served by a base station, the MAC scheduling at the base station in a cellular network), in-network latency (caused by huge buffers deployed at the LTE base stations, core network conditions and congestion at each routing hop, etc.), interaction of the application layer with wireless channel conditions, etc.

There are various ways to use the download rate traces in an emulation environment. In [25], packet arrival times are used to inject delays in the emulator to replicate cellular network conditions. Since our emulation environment is based on setting attenuation levels, we are interested to investigate the relationship between download rate and signal strength. To do so, we parametrize the correlation between download test rates and RSSI for our measurements, by calculating the Pearson's product-moment coefficient for them. The closer the correlation coefficient is to 1, greater is the linear dependence between the two variables. One example of highly correlated measurement (with a correlation coefficient of 0.72) for a commercial cellular network is depicted in Fig. 5.[1]

Not all the RSSI-bitrate measurements we conducted show this level of correlation. As a matter of fact, the average correlation coefficient for each of the cellular networks over all of the traces is 0.36. Considering all the aforementioned factors that have an impact on the value of download bitrate, only when those factors are at rest we can see the download bitrate to be a function of RSSI. For example, the measurements shown in Fig. 5 have been recorded during off-peak time[2]. In order to investigate the role of other parameters involved in mapping between download bitrate and RSSI (which will be ultimately fed to our system to replay the trace), we plotted the correlation coefficient between RSSI and measured download test bitrates, as a function of time of day and average speed of the vehicle in Fig. 6. As can be seen, the correlation is highest for off-peak times and higher average vehicle speed (which is an indication of less traffic on the road and less load on the cellular network), and lowest for rush hour times with traffic on the road. Similar analysis can be done to extract statistical information regarding the mapping of download bitrates to RSSI measurements based on parameters at hand (ranging from the most basic ones like time of day, or traffic on the road, to fine-grained information from the base stations or the network).

• **Location/Distance-based Emulation:** Another type of trace that can be replayed in our system is the mobility (location) trace of a user. There exist databases [26] of signal quality for different cellular carriers. By specifying the route of the user the signal strength information can be extracted

[1]In the RSSI plot, for consistency purposes only the RSSI measurements denoting LTE technology have been shown. It can be seen that the download rate vastly decreases when the connectivity downgrades from LTE to a different technology (discontinuity in the RSSI plot at 3000 seconds)

[2]The average speed of the car driven while making the measurements shows no effect of traffic, which corresponds to a lightly-loaded system.

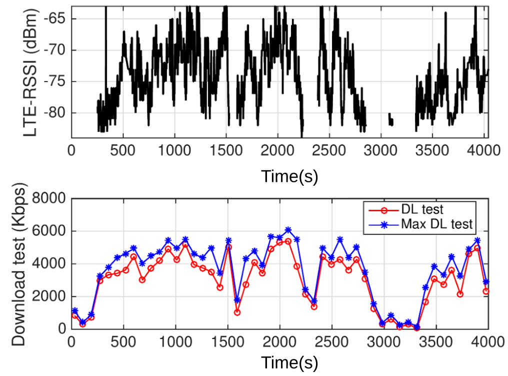

Fig. 5. Example of a highly correlated RSSI and download test bitrate

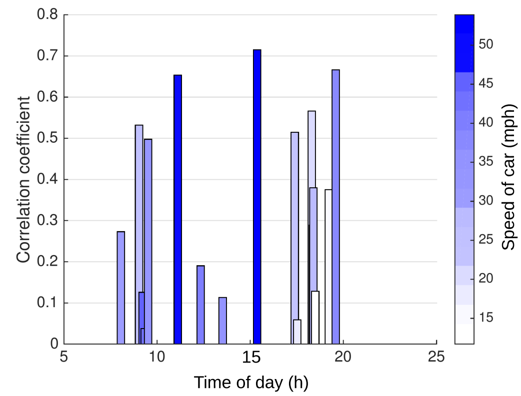

Fig. 6. Correlation Coefficient between RSSI and download test bitrate. The color of each bar denotes the average speed of the vehicle (mph), which can be an indicator of traffic along the route, hence the load on the system

and then after mapping the data to the operating range of our system, the mobility trace can be replayed. In our set of measurements, we investigated the variance of RSSI values for different times, but constant locations on a route for a commercial cellular network. The route is nearly 14 miles, as shown in Fig. 3, and the subset of traces considered is 13. We averaged out all the measurements in a trace, in steps of 0.1 mile and then computed the mean and standard deviation across the 13 traces for each 0.1 mile interval. The result is shown in Fig. 7. The average of standard deviation is 1.41 and as can be seen, the RSSI variance is around 4-5 dBm for each 0.1 mile step. This confirms some level of correlation between RSSI values for specific locations and the possibility of replaying mobility traces in our system.

If databases for mapping specific locations to corresponding signal strength is not available, there exist also the opportunity to conduct distance-based emulations. By knowing the route of the user and obtaining base station distribution from available databases [27], [28], one can apply mappings to translate the distance values to signal attenuation values (following mainstream path-loss and fading models). These attenuation values can be injected to our system, to replay the mobility trace of a user.

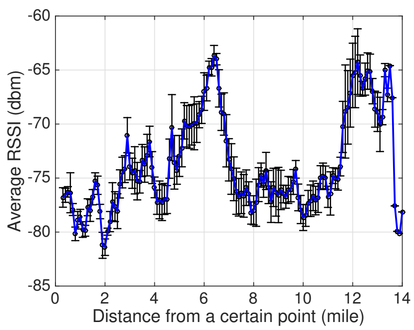

Fig. 7. Mean and Variance for RSSI values, shown for specific locations on a route.

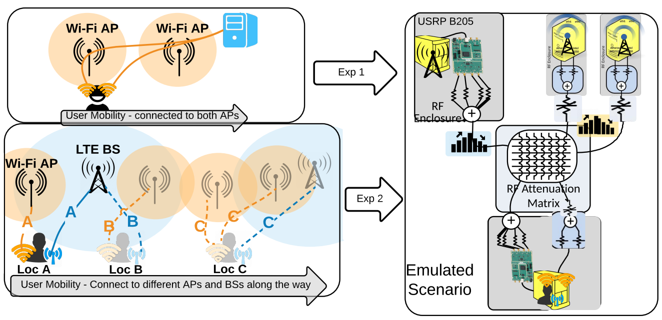

Fig. 8. Proof-of-concept experiments demonstrating the utilization of EmuWNet framework

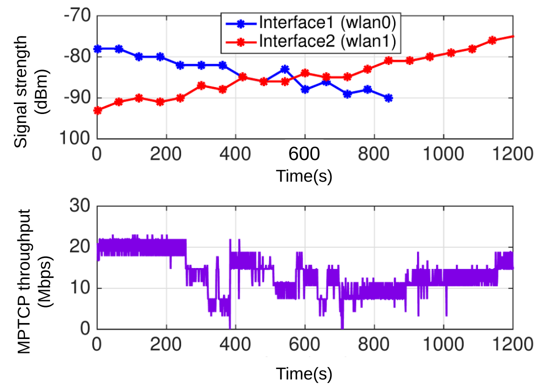

Fig. 9. Emulating the mobility of a user across two Wi-Fi APs. (a) RSSI levels for one connection decreases as it increases for the other one (The client looses its connection on Interface 1 after 800 secs).(b) The downlink throughput measured by *iperf* over MPTCP through the two Wi-Fi interfaces.

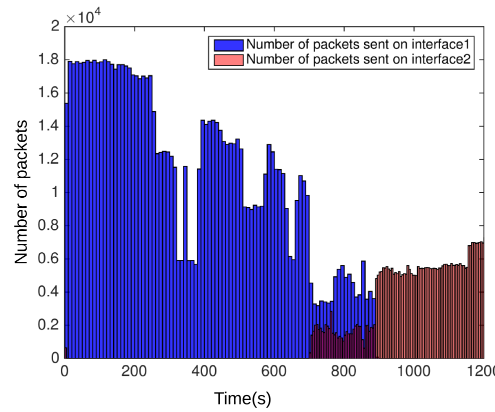

Fig. 10. Utilization of interfaces as the RSSI values varies in time

IV. Experimental Evaluation

In this section we present the results of two proof-of-concept experiments we performed using EmuWNet. As shown in Fig. 8, in the first experiment a baseline mobility scenario is emulated, where a Wi-Fi user is connected to 2 Wi-Fi Access Points (APs) simultaneously and maintains Multi-Path TCP [29] connection to a remote host. In this baseline setup, a Wi-Fi handover scenario is emulated, illustrating how the data flow remains active using MPTCP. In the second experiment, shown in Fig. 8 we replay 300-400 seconds of measured RSSI traces from the RSSI database obtained using our app for Wi-Fi and LTE links. Through this experiment we show the repeatability of the emulated scenarios by comparing throughput of TCP/IP based data transmission with ICN-based (MobilityFirst [10]) data transmission.

A. MPTCP Wi-Fi handover experiment

As shown from the setup of the experiment in Fig. 8, the user is equipped with two Wi-Fi interfaces and associated with both of the APs simultaneously. As the user moves from one AP to the other, the RSSI it senses for each AP changes accordingly. Our goal was to emulate this scenario by linearly increasing the attenuation on one link, and decreasing it on the other. We enabled MPTCP on both the server and the user, and ran *iperf* to investigate the downlink throughput of the user in this emulated mobile scenario. The results are shown in Fig. 9. The results illustrate how the user's throughput degrades as it supposedly moves away from one AP, and then increases as it supposedly gets close to the other AP. In order to better clarify the user's connection to each of the APs, we investigated interface utilization on the user by looking into the number of packets it is receiving on each interface, shown in Fig 10.

B. TCP/IP vs. ICN-based data transfer experiment

In this experiment, we demonstrate the repeatability of the emulation scenarios, by taking as input to the emulation framework a sample of measured RSSI traces for LTE and Wi-Fi connections and test the data transmission throughput for TCP/IP and ICN-based protocol stack, shown in Fig. 8. As an example for ICN-based communication we use MobilityFirst protocol stack and network components (routers and global name resolution service (GNRS)) [30] to run this experiment. Hostapd daemon [31] is used for Wi-Fi AP implementation and management. For LTE, OpenAirInterface (OAI) [32] on USRP nodes is used, which is an open-source software implementation, fully compliant with 3GPP LTE standard (Rel14). The input RSSI measurements and throughput results measured by *iperf* [3] for LTE and Wi-Fi links are shown in Fig. 11 and Fig. 12. As can be seen for both runs of

[3]In the case of MobilityFirst a modified version of iperf which uses MF API calls is used

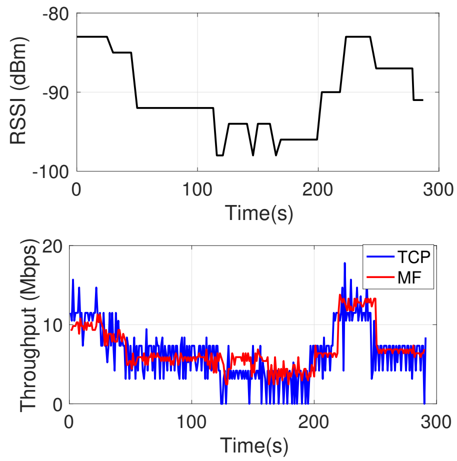

Fig. 11. LTE RSSI measurement emulation and throughput results

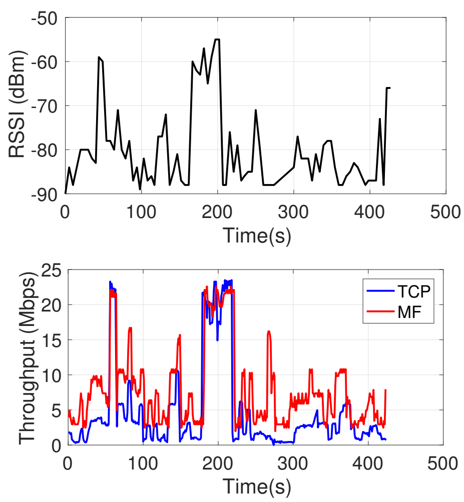

Fig. 12. Wi-Fi RSSI measurement emulation and throughput results

the experiment, the MF and TCP throughput both follow the input Wi-Fi and LTE RSSI patterns. Higher MF throughput and better utilization of the wireless links have been reported in [33] and using the EmuWNet emulation framework we observed similar trends. This shows EmuWNet provides a useful research tool for fair comparison of novel protocols with existing ones, and this can contribute to better design and tuning of protocols on an end-to-end basis for next-generation of wireless networks.

V. Conclusions

In this paper we proposed EmuWNet, a framework to emulate realistic mobile wireless networks on an open-access testbed. We described in detail how real world wireless channel conditions as well as simulated data can be mapped onto attenuator matrices to experiment with various topologies and multiple access technologies. Representative traces and selective experimentation results of a mobile smartphone using Wi-Fi and LTE were provided. As part of our future work we plan to utilize EmuWNet for experimenting with other advanced mobile use-cases such as novel multi-network access protocols and LTE/Wi-Fi coexistence.

References

[1] C. V. N. Index, "Global mobile data traffic forecast update, 2015-2020," *White Paper, February*, 2016.

[2] A. Osseiran *et al.*, "Scenarios for 5g mobile and wireless communications: the vision of the metis project," *IEEE Communications Magazine*, 2014.

[3] P. Rost *et al.*, "Cloud technologies for flexible 5g radio access networks," *IEEE Communications Magazine*, 2014.

[4] R. Zhang *et al.*, "Lte-unlicensed: the future of spectrum aggregation for cellular networks," *IEEE Wireless Communications*, 2015.

[5] F. Abinader *et al.*, "Enabling the coexistence of lte and wi-fi in unlicensed bands," *IEEE Communications Magazine*, 2014.

[6] H. Hartenstein *et al.*, "A tutorial survey on vehicular ad hoc networks," *IEEE Communications magazine*, 2008.

[7] C. Lochert *et al.*, "A routing strategy for vehicular ad hoc networks in city environments," in *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, 2003.

[8] T. Leighton *et al.*, "Content delivery network using edge-of-network servers for providing content delivery to a set of participating content providers," 2003, uS Patent 6,553,413.

[9] M. Hudson *et al.*, "Mediated multi-source peer content delivery network architecture," 2003, uS Patent App. 10/132,954.

[10] D. Raychaudhuri *et al.*, "Mobilityfirst: a robust and trustworthy mobility-centric architecture for the future internet," *ACM SIGMOBILE Mobile Computing and Communications Review*, 2012.

[11] L. Zhang *et al.*, "Named data networking (ndn) project," *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC*, 2010.

[12] J. Zhou *et al.*, in *IEEE INFOCOM*, 2006.

[13] R. Beuran *et al.*, "A multi-purpose wireless network emulator: Qomet," in *aina workshops*, 2008.

[14] P. Zheng *et al.*, "Emwin:: Emulating a mobile wireless network using a wired network," in *ACM WOWMOM*, 2002.

[15] P. Mahadevan *et al.*, "Mobinet: A scalable emulation infrastructure for ad hoc and wireless networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, 2006.

[16] Y. Zhang *et al.*, "An integrated environment for testing mobile ad-hoc networks," in *ACM MobiHoc*, 2002.

[17] D. Raychaudhuri *et al.*, "Overview of the orbit radio grid testbed for evaluation of next-generation wireless network protocols," in *IEEE WCNC 2005*.

[18] G. Huang *et al.*, "Metis-ii," *Deliverable D7.3 Final 5G visualization*, 2017.

[19] "Hardware configuration of sandbox 4," http://www.orbit-lab.org/wiki/Hardware/bDomains/cSandboxes/dSB4#SB4, Accessed: 2018-01-21.

[20] "Ettus research, "usrp b200 and b210"," http://tinyurl.com/k7w6zh2.

[21] "Linuxwireless ath9k," https://wireless.wiki.kernel.org/en/users/drivers/ath9k.

[22] "Jfw industries, inc. manufacturer of rf components and automated rf test systems," http://www.jfwindustries.com, Accessed: 2018-01-21.

[23] T. Rakotoarivelo *et al.*, "Omf: a control and management framework for networking testbeds," *ACM SIGOPS Operating Systems Review*, 2010.

[24] "Information System on Graph Classes and their Inclusions," http://www.graphclasses.org/smallgraphs.html, Accessed: 2018-01-21.

[25] W. K. Winstein *et al.*, "Stochastic forecasts achieve high throughput and low delay over cellular networks," in *NSDI*. USENIX, 2013.

[26] "G-NetWorld," Accessed: 2018-01-21. [Online]. Available: http://www.gyokovsolutions.com/G-NetWorld/G-NetWorld.php

[27] "The world's largest Open Database of Cell Towers," Accessed: 2018-01-21. [Online]. Available: http://opencellid.org/

[28] "CellID Finder," Accessed: 2018-01-21. [Online]. Available: http://cellidfinder.com/

[29] "MultiPath TCP - Linux Kernel implementation," Accessed: 2018-01-21. [Online]. Available: http://www.multipath-tcp.org/

[30] F. Bronzino *et al.*, "Network service abstractions for a mobility-centric future internet architecture," in *ACM MobiArch*, 2013.

[31] "hostapd: Ieee 802.11 ap, ieee 802.1x/wpa/wpa2/eap/radius authenticator," http://w1.fi/hostapd/.

[32] N. Nikaein *et al.*, "Openairinterface: A flexible platform for 5g research," *SIGCOMM Comput. Commun. Rev.*, 2014.

[33] P. Karimi *et al.*, "Evaluating 5g multihoming services in the mobilityfirst future internet architecture," in *IEEE VTC*, 2017.