

An LTE-WiFi Interworking Platform with Real-Time PHY Layer Interface

Walter P. Nitzold
National Instruments Corp.
Austin, TX, USA
walter.nitzold@ni.com

Clemens Felber
National Instruments Corp.
Austin, TX, USA
clemens.felber@ni.com

Vincent Kotzsch
National Instruments Corp.
Austin, TX, USA
vincent.kotzsch@ni.com

Abstract—New use cases and requirements of future communication systems have driven the focus of research towards the coordination and coexistence of heterogeneous wireless technologies such as LTE, WiFi as well as 5G. In this paper we present a prototyping system focused on the joint real-time experimentation of LTE and WiFi systems utilizing a single software-defined radio platform. We present a new general API architecture that connects FPGA based PHY layer implementations to higher layers. End-to-End throughput measurements underline the capabilities of this approach. This platform allows new experimentation results that will lead to a better understanding of the trade-offs when using different radio access technologies *together* as it would be the case in practical wireless network deployments.

Index Terms—LTE, WiFi, FPGA, 5G, SDR, prototyping, Network, API

I. INTRODUCTION

In the 3GPP LTE standard there are multiple options to split the traffic over different air interfaces to increase the system capacity e.g. using dual connectivity or carrier aggregation between LTE base stations or using LTE-WLAN aggregation (LWA/LWIP) [1]. Similar techniques are also used for combining 4G LTE with the new 5G NR standard in a non-standalone mode. In order to understand the trade-offs between these radio access technology (RAT) options joint interworking simulations and experiments are required. With existing Software-Defined Radio (SDR) platforms and corresponding open source protocol stacks it is either possible to prototype LTE *or* WiFi systems. In the context of the H2020 project "ORCA" [2], a platform for real-time multi-RAT experimentation is currently being developed that allows for prototyping LTE *and* WiFi systems and will integrate a 5G link in the future. A first attempt to implement these interworking techniques with SDR hardware was shown in [3] with the help of Open Air Interface [4]. In this paper, we introduce this multi-RAT prototyping system which utilizes the ns-3 simulator as well as the NI SDR platform. We discuss how to use this system for joint LTE and WiFi experimentations including RAT interworking techniques. A key ingredient of the platform is a newly developed general API that connects the PHY and MAC layer. This API is used to interface the ns-3 simulator to the FPGA based real-time PHY implementation from the NI Application Frameworks. An assessment of the performance of the system is done by measuring end-to-end throughput and compare it to theoretical values.

The paper is structured as follows. Section II introduces the LTE-WiFi interworking scenario implementation in ns-3. In Section III, we present the components of the NI SDR prototyping system that are used as PHY layer implementations in the platform. Section IV gives insight into the concept and functionality of the general API that was implemented for LTE as well as WiFi to interface PHY and MAC layer. Some results on end-to-end throughput measurements are presented and discussed in Section V. Finally, the paper is concluded in Section VI.

II. LTE-WiFi INTERWORKING USING NS-3

To model the upper layers of the multi-RAT system the open source ns-3 simulator is used. This simulation environment includes a broad set of functionalities and modules for testing wired and wireless communication systems in a single simulation. Provided communication technologies include LTE

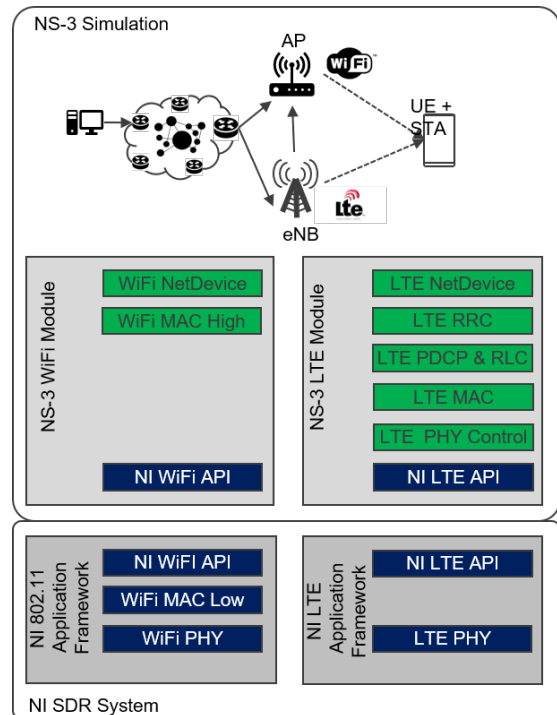


Fig. 1. ns-3 based LTE-WiFi Interworking System

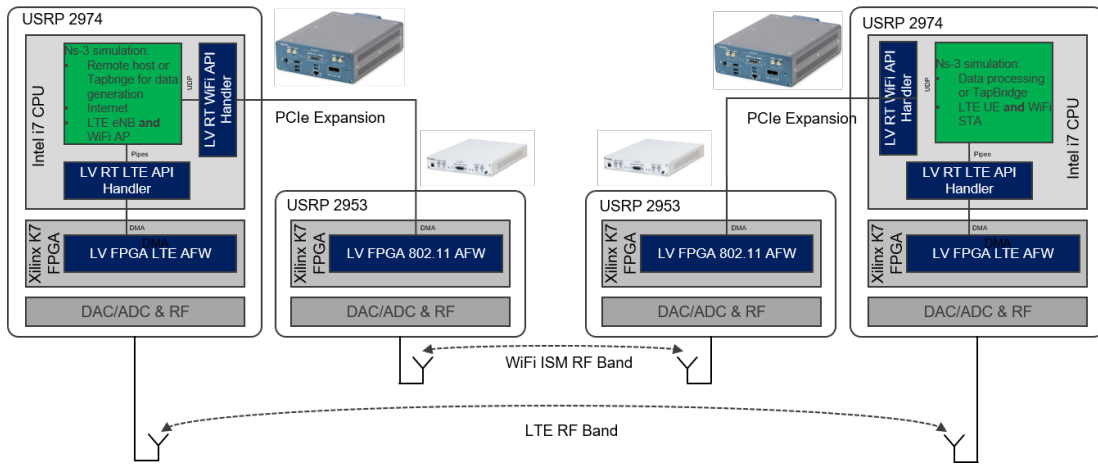


Fig. 2. LTE-WiFi Prototyping Platform using NI SDR

and WiFi as well as CSMA/Ethernet and even point-to-point connections comparable, e.g., to USB. More information about the capabilities of ns-3 can be found in [5]. The ns-3 WiFi model [6] offers a full stack implementation of the 802.11 standard for Access Point (AP) and Station (STA). The ns-3 LTE model [7] includes functionality to instantiate a complete core network with attached base stations (eNB) and user terminals (UE). To accommodate for simulations over real wireless channels, ns-3 uses physical layer abstraction techniques based on typical SINR models. However, ns-3 is already equipped with a real-time simulator mode where the simulator time is coupled to the CPU time. This is beneficial for our goal of real-time prototyping with ns-3.

In order to exploit the multi-RAT functionalities provided in ns-3 in a real-time end-to-end prototyping environment we implemented an NI API between the ns-3 LTE and WiFi modules. This allows to connect the ns-3 simulator to the NI real-time SDR platform described below including real-time capable PHY implementations running on an FPGA. A sketch of the system partitioning is shown in Fig. 1. As one can see, for the LTE module a split between the MAC and PHY is used which is intuitive as we want to bypass the PHY emulation model in ns-3. However, for the WiFi module we implemented a split between the higher MAC and lower MAC because of the stricter timing requirements of the listen-before talk mechanism given by the standard that cannot be fulfilled by the ns-3 WiFi module. Details regarding the API are discussed in Section IV.

To enable real-time wireless network prototyping, a new end-to-end network scenario was implemented which includes all necessary network components also found in practice. The new topology is depicted in Figure 3.

A configurable number of remote host nodes are connected to a subnet using the internet simulation model via the ns-3 Ethernet module. The Ethernet network is further comprised of a mobile network gateway that serves the purpose of connecting the access networks of LTE and WiFi to the Ethernet network. The LTE access network includes the Evolved Packet

Core (EPC) as well as a configurable number of eNBs. The WiFi access network can be equipped with a configurable number of APs. In addition, we implemented a combined LTE and WiFi terminal node that has both an LTE UE as well as a WiFi station to accommodate real user terminals such as mobile phones. Each interface within the combined terminal can be accessed with distinct IP addresses. To further enhance the capabilities of the platform, we use an implementation of the LTE/WiFi interworking technologies LWA and LWIP [8]. This implementation allows to route traffic flows from within the LTE network over an WiFi AP that is associated to an eNB. The traffic is then received at the WiFi STA and rerouted into the LTE UE stack. With this topology, traffic splits and rerouting between LTE and WiFi at both mobile network gateway as well as LTE eNB can be investigated.

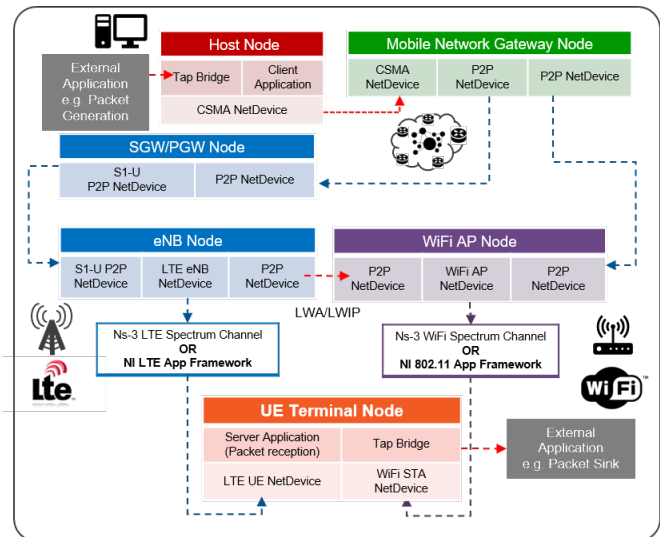


Fig. 3. LTE-WiFi Interworking topology diagram.

III. NI SDR PROTOTYPING SYSTEM

Our prototyping setup utilizes the NI USRP-2974 [9]. This is a standalone software-defined radio device that includes an Intel Core i7 CPU, a Xilinx Kintex-7 FPGA together with ADC/DAC and RF frontend. The covered frequency range spans 10 MHz - 6 GHz with an instantaneous bandwidth of 160 MHz. This bandwidth option allows for usage both in LTE with carrier bandwidths up to 20 MHz as well as 802.11a/ac with a bandwidth up to 80 MHz. The USRP-2974 device includes an embedded controller with an Intel i7 CPU comprised of 4 cores. An NI Linux Real-Time operating system is deployed to the controller which is utilized to compile and run the ns-3 code. Additionally, we use the NI LTE Application Framework [10] which is an FPGA based real-time implementation of an LTE PHY. To cover the WiFi part, we utilize the NI 802.11 Application Framework [11] which is a FPGA based real-time implementation of WiFi PHY and lower MAC. To combine and run the two radio-access technologies from a single device, we use the extension option of the USRP-2974. Through an extension port, the USRP-2974 can be connected to a legacy USRP RIO. With that both LTE and WiFi can be run on dedicated FPGAs while a single ns-3 instance is running on the Intel CPU of the embedded controller. The overall system setup is depicted in Figure 2.

The compactness of this system approach allows in a simple and easy way the joint prototyping of LTE and WiFi on a single SDR platform. The options of multi-RAT functionalities of ns-3 in conjunction with the PHY layer performance of the NI Application Frameworks offers research and prototyping possibilities on all layers. To further enhance the setup and connect the prototyping platform to real-world services and applications, a ns-3 tap bridge can be used to inject external traffic. This can, e.g., be used to stream a video through the LTE/WiFi network as well as conducting experiments for assessing throughput and delay with the whole network for measurements on different layers. An exemplary measurement of throughput for our system is discussed in Section V.

IV. GENERIC NI L1-L2 API IMPLEMENTATION

To combine the strengths of both the ns-3 network simulator as well as the NI Application Framework FPGA implementations of LTE and WiFi we developed a generic API concept to interface these components. The API is strongly influenced by the Femto Forum API [12] but as this API did not include a specific definition for the UE side for LTE, we developed an enhanced version to support both eNB and UE for LTE. We also extended the general API concept to be used for WiFi as well.

The L1-L2 API communication between MAC and PHY layer follows a common message passing paradigm of three main message types as depicted in Figure 4.

The service request (REQ) is a message designated from the higher layer to the lower layer. This message type typically contains control parameters to configure the PHY Layer. It additionally may convey also data to the lower layer but the message format is generic to allow for separated control and

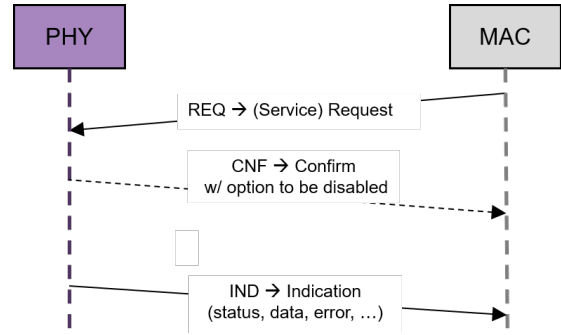


Fig. 4. Generic Message Types of NI API

data transmission. To allow a behavior with increased robustness, each request can be confirmed by a confirm message (CNF) from the lower layer. As this on one hand improves the robustness of the API it also adds additional complexity especially to the FPGA targets. Therefore, the confirm can also be disabled. For communication from lower to higher layer, the indication message (IND) is used. This message type is used for conveying status information to the higher layer as well as sending data packets from PHY to MAC.

The structure of the API messages is depicted in Figure 5.

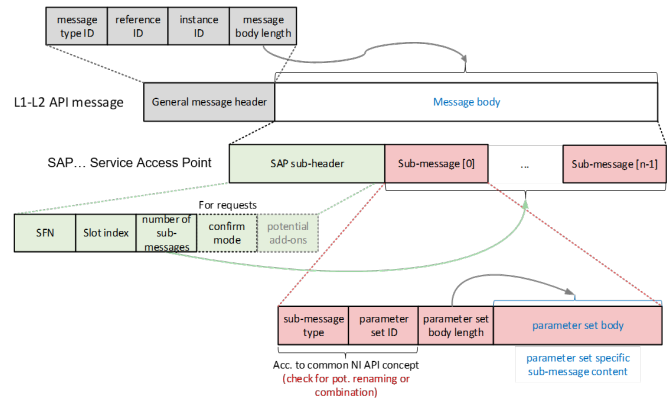


Fig. 5. Generic Message Format of NI API

An L1-L2 API message is comprised of a generic API message header. This header includes information about the message type, a reference ID, an instance ID and the message body length. While the message type identifies what standard (LTE or WiFi) and what message is encapsulated in the body, the instance ID, e.g., allows for the execution of multiple API instances on one platform. The reference ID is used to match, e.g., a confirm message that corresponds to a specific request message. In this case, reference ID in both messages should be equal. The message body itself is then comprised of the dedicated message for the Service Access Point (SAP) that depends on the standard chosen. Detailed descriptions of the message content for can be found in [13] for LTE and [14] for WiFi.

A particular difference of the functionality of both WiFi and LTE API is the timing requirements. In the WiFi ns-3

stack, packets will be handed over from the WiFi MAC High instance (see Figure 1) of ns-3 to the WiFi MAC Low instance in the 802.11 Application Framework whenever these packets arrive at the WiFi MAC High. They will be processed by the PHY layer as soon as possible on best effort. The time critical components in WiFi run on the FPGA to achieve e.g. the SIFS timing which is in the order of μs - that is the main difference and an advantage of our architecture to run only the time uncritical steps of MAC High in ns3 and the time critical Low MAC and PHY on the FPGA. The paradigm of the NI LTE API is different to WiFi in the sense that LTE itself runs with a timing grid of 1 ms which corresponds to a subframe or TTI. This timing grid is dictated by the FPGA implementation of the PHY layer and communicated to the higher layer by a timing indication message that is sent from the PHY every 1ms. The ns-3 LTE stack has to ensure that all processing dedicated to the TTI for which the timing indication was sent has to be finished within 1 ms. This has influence on the computational burden of ns-3 on the CPU as well as the transport mechanism used for the NI L1-L2 API messages. As all time critical components of WiFi run on the FPGA, we can use UDP socket connections as the WiFi API message transport mechanism. The API messages are encapsulated into UDP packets. That allows for flexible connections of different ns-3 WiFi stack instances with underlying Low MAC and PHY layer implementations. On the other hand for LTE, due the aforementioned timing constraints, a UDP socket connection is too jitter prone. Measurements of message transmission over UDP have shown jitter spikes up to 8 ms which violates the timing requirements of 1 ms. Therefore, we use the more deterministic transport method of Linux named pipes which behaves like a FIFO with negligible latency and jitter.

With the help of a generic API definition, suitably tailored API message definitions for LTE and WiFi and an API message transport mechanism, we are able to interface the ns-3 LTE-WiFi interworking topology described in Section II to the NI Application Frameworks to enable end-to-end prototyping of networks with multiple radio access technologies involving real PHY layer implementations for simulations over real radio channels.

V. END-TO-END THROUGHPUT MEASUREMENTS

To assess the performance of the ns-3 LTE-WiFi interworking platform with attached NI LTE and 802.11 Application Frameworks we conducted measurements for end-to-end throughput. We measure the throughput for WiFi and LTE by using the aforementioned LTE-WiFi interworking topology shown in Figure 3. The setup is comprised of a remote host node that acts as a client which sends UDP packets with payload data. The corresponding server that represents the packet sink for these measurements resides in the combined WiFi-LTE UE node. The UDP packets which are sent from the client in the remote host node are injected into ns-3 with the help of a tap bridge. This tap bridge appears to the Linux host as a normal Ethernet interface with an IP address. An external arbitrarily configurable UDP packet

generation application is used to generate traffic. Similarly on the server side of the transmission, the packets are output from the UE node through a tap bridge to an external server sink application that calculates the end-to-end throughput. The connections for the tap bridges can be seen in Figure 3. To assess whether the whole system is capable of conveying an envisaged throughput, we alter packet sizes as well as packet send intervals in the traffic generation of the client application. The theoretical end-to-end throughput is then calculated by dividing the packet size in bits by the packet interval. The throughput measurements for a traffic flow over WiFi and over LTE is shown in Figure 6 and Figure 7 respectively.

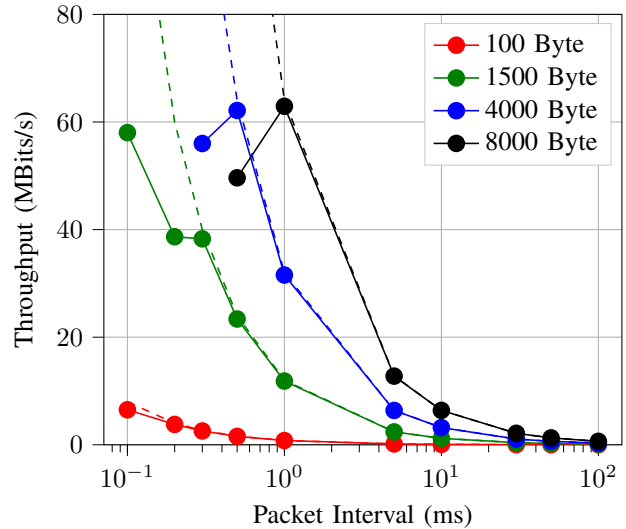


Fig. 6. End-to-End Throughput Measurements for WiFi. Solid lines show measured throughput. Dashed lines represent theoretical achievable throughput.

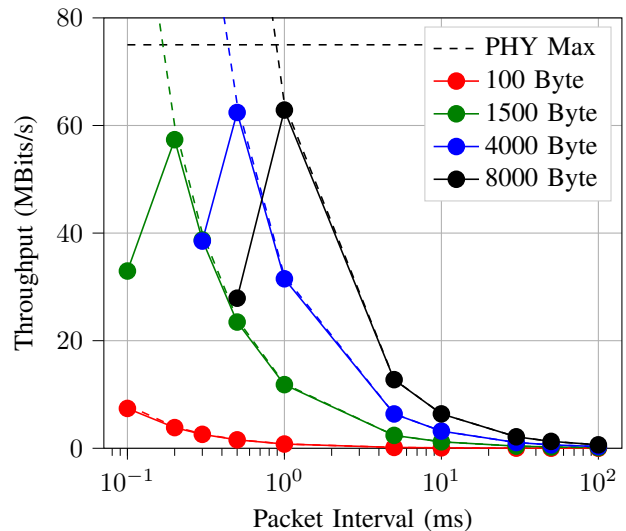


Fig. 7. End-to-End Throughput Measurements for LTE. Solid lines show measured throughput. Dashed lines represent theoretical achievable throughput. The maximum PHY layer throughput of 75 MBit/s is plotted for reference.

In general, for WiFi the throughput for all considered packet sizes follows the theoretical curve depending on the packet transmission interval. For small packet sizes such as 100 Bytes, the packet interval can be reduced down to 0.1 ms. This represents a use case where packets of small size need to be conveyed with low latency. The prototyping platform is capable of conveying the theoretical end-to-end throughput with only negligible deviation for small packet transmission intervals. For increasing packet size, the packet transmission interval cannot arbitrarily lowered to yield higher throughputs. The system saturates at around 65 MBit/s. If one lowers the packet transmission interval further, packet losses start to occur which in turn also lower the throughput in a non-deterministic way. For LTE, the behavior is similar and the achievable end-to-end throughputs approach the theoretical limit of 75 MBit/s for an LTE configuration with 20 MHz bandwidth and highest modulation and coding scheme. A small gap between theoretical and achieved throughput can still be observed but is due to the header overhead that is added through the different layers of the full LTE stack implementation. The throughput curves for WiFi and LTE look similar as expected for an end-to-end throughput measurement where the PHY and higher layers of the respective standard are only abstractions for data transfer.

The measurements show, that the prototyping platform is capable of achieving close to theoretical limits in terms of end-to-end throughput without any restrictions for computationally demanding PHY layer configurations. The evaluations have been done for LTE and WiFi separately. Initial investigations regarding LTE-WiFi interworking show the expected throughput enhancements seen in theory although a proper assessment is subject of future research.

VI. CONCLUSION

With the connection of NI's SDR platform to the higher layer ns-3 network simulator, various deployment scenarios are possible that allow for experiments which have not been possible so far. We have given an overview of the ns-3 LTE-WiFi interworking topology in ns-3 and how it can be used to cover many different transmission scenarios. A integral part of our implementation is the development of a general API concept to interface PHY layer real-time FPGA implementations such as the NI LTE and 802.11 Application Framework toward the higher layers of ns-3. We additionally showed results on the end-to-end throughput of our prototyping system which underline the computational advantages of an FPGA PHY implementation. The source code is publicly available on Github [15]. In the future we plan to complete thorough simulations for LTE-WiFi interworking as well as to extend the measurements to assess other metrics such as jitter and delay of the end-to-end system. Further, we plan to also integrate a 5G link in the multi-RAT platform that allows experimentation scenarios using 5G - LTE - WiFi interworking techniques. The entire system will be available throughout ORCA project and its testbeds. It will facilitate the prototyping in a real-time end-to-end testbed and will create better understanding

of the general requirements of RAT interworking including all aspects of a wireless network.

ACKNOWLEDGMENT

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 732174 (ORCA).

REFERENCES

- [1] *3GPP Technial Specification 36.300*, V15.4.0, Dec 2018
- [2] "<https://www.orca-project.eu/>" *Orchestration and Reconfiguration Control Architecture (ORCA)*, March 2019
- [3] D. Ibarra, N. Desai and I. Demirkol, "Software-Based Implementation of LTE/Wi-Fi Aggregation and Its Impact on Higher Layer Protocols," 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, 2018, pp. 1-6.
- [4] "<https://www.openairinterface.org/>" *OpenAirInterface*, August 2019
- [5] "<http://www.nsnam.org/>" *Network simulator ns-3*, March 2019
- [6] "<https://www.nsnam.org/docs/models/html/wifi.html>" *ns-3 Wi-Fi Module Documentation*, March 2019
- [7] "<https://www.nsnam.org/docs/models/html/lte.html>" *ns-3 LTE Module Documentation*, March 2019
- [8] M. Afaqui, C. Cano, V. Kotsch, C. Felber, W. Nitzold, "Implementation of the 3GPP LTE-WLAN Interworking Protocols in NS-3," arXiv 1901.03683, 2019.
- [9] "<http://www.ni.com/white-paper/52119/en/>" *Overview of the NI USRP RIO Software Defined Radio*, March 2019
- [10] "<http://www.ni.com/white-paper/53286/en/>" *LabVIEW Communications LTE Application Framework 2.0 and 2.0.1*, Dec 2016
- [11] "<http://www.ni.com/white-paper/53279/en/>" *LabVIEW Communications 802.11 Application Framework 2.0 and 2.0.1*, Dec 2016
- [12] "http://scf.io/en/documents/082_-_nFAPi_and_FAPi_specifications.php" *Small Cell Forum, FAPI and nFAPi specifications*, May 2017
- [13] "https://download.ni.com/evaluation/sdr/LTE_Application_Framework_Manual_ver_2.5.pdf" *LabVIEW Communications LTE Application Framework 2.5 Manual*, May 2019
- [14] "https://download.ni.com/evaluation/labview/appframework/802.11AFW_Manual_ver_2.5.pdf" *LabVIEW Communications 802.11 Application Framework 2.5 Manual*, May 2019
- [15] "<https://github.com/ni/NI-ns3-ApplicationExample>" *National Instruments ns-3 Application Example*, May 2019