Overview
Next-generation wireless networks (5G) cope with significant traffic increase because of high-quality video transmission and cloud-based applications. This creates the need for a revolutionary change in architecture rather than a series of local and incremental technology updates. A dense heterogeneous deployment of small cells such as pico/femto cells in addition to high-power macro cells is a potential solution. In this area, a significant amount of research relies on simulations at PHY, MAC, and higher layers, but it is still necessary to validate the algorithms for next-generation systems in a real-time testbed. The ever-increasing complexity in all layers of current and future generations of cellular wireless systems, however, has made an end-to-end demonstration of the network limited to industrial research labs or large academic institutions. As a solution to tame the dense deployment of wireless networks, we propose an NI PXI platform based on NI LabVIEW software in which an LTE-like SISO OFDM PHY layer is integrated with an open-source protocol stack to prototype PHY/MAC cross layer algorithms within a CROWD (Connectivity management for energEty Optimised Wireless Dense networks) software-defined networking (SDN) framework.

Table of Contents
1. Introduction
2. CROWD Architecture
3. Introduction to Open-Source NS-3 LENA LTE Stack
4. Preliminary Results
5. Conclusion
6. Authors
7. Acknowledgement
8. References

1. Introduction
Mobile data traffic demand is growing exponentially and the trend is expected to continue, especially with the deployment of 4G networks. To cope with the rapid explosion of traffic demand, mobile network operators have started to push for denser, heterogeneous deployments. Current technology needs to steer toward efficiency to avoid unsustainable energy consumption and network performance collapse due to interference and signaling overhead. Interference due to uncoordinated resource sharing techniques represents a key limiting factor in the design of dense wireless networks, where resources are limited because of either the costs for licensed bands or the proliferation of hot spots in license-exempt bands. This situation calls for deploying network controllers with either a local or regional scope, with the aim of orchestrating the access to wireless and backhaul resources of the various network elements. In the FP7 CROWD project [1], [13], we show how an SDN-based approach can be used to design the next generation of dense wireless mobile networks. This approach provides flexibility and reconfigurability while offering an energy-efficient network infrastructure for both the radio access and the backhaul.

To meet the interference challenge in such dense deployments, a significant amount of research has relied on simulations at PHY, MAC, and higher layers. However, these algorithms need to be validated in lab deployments to evaluate the performance of the complete system in a realistic setting. The ever-increasing complexity in all layers of current and future generations of cellular wireless systems has made an end-to-end demonstration of the LTE network limited to industrial research labs or large academic institutions. There are several commercial solutions from Mymo Wireless [2], Amarisoft [3], and so on, but, for an academic institution, they are expensive to deploy. There is also an open-source implementation of the LTE eNB/UE stack within the Open Air Interface project [4]; however, it uses custom RF hardware for integration with baseband and is still in development.

The main motivation in our project is to build a cost-effective testbed that can interface easily with open-source simulators like the NS-3 LENA LTE stack [5], so that we can create a unified platform for both simulating and prototyping. Conventionally, researchers conduct simulations using tools like NS-3, and, when good results are achieved, they need to rewrite the algorithm in a different development environment of the prototyping platform. This creates a barrier in prototyping small cell LTE interference mitigation algorithms and, considering that the LTE standard as a whole is complex, usually designing new algorithms in the PHY/MAC layer negatively affects higher layers or vice versa. This motivated us to build a prototyping platform using the NS-3 LTE LENA stack and interface it with a realistic subset of the physical layer using LabVIEW, so that researchers can focus on the performance of algorithmic aspects rather than the implementation details of the algorithm within the LTE standard in a realistic testbed. Nevertheless, we tried to ensure that essential LTE physical layer features were standard-compliant or the behavior was at least close to LTE philosophy to ensure that we could draw meaningful conclusions from the testbed experiments.

To interface the NS-3 LTE LENA stack with a realistic physical layer and be able to run in real time, we needed to implement specific physical layer control/data channels (PD-CCH/PDSCH) in FPGA/DSP or a similar high-performance computing platform. However, writing physical layer code in Verilog/VHDL can be daunting and interfacing the baseband with an RF front end has its own set of challenges. Therefore, we chose LabVIEW as the software defined radio (SDR) platform. The LabVIEW platform is PXI-based and is a powerful and feature-rich rapid prototyping tool [6] for research in real-time wireless communication systems.
This SDR platform provides a heterogeneous environment, which includes a multicore Windows/Linux PC and real-time OS (RTOS) running on high-performance general-purpose processors (GPPs) such as Intel processors and NI FlexRIO FPGA modules containing Xilinx Virtex-5 and Kintex-7 FPGAs. It also provides a set of RF, digital-to-analog converter (DAC), and analog-to-digital converter (ADC) modules that can meet the bandwidth and signal quality requirements of 5G systems. A traditional SDR prototyping engineer faces several challenges that arise from the use of different software design flows to address different components of the system (that is, RF, baseband, and protocol stack). In addition, the components may lack a common abstraction layer. This can result in complications and delays during system development and integration. LabVIEW addresses these challenges by providing a common development environment for all the heterogeneous elements in the NI SDR system (that is, the GPP, RTOS, FPGA, converters, and RF components) with tight hardware and software integration and a good abstraction layer [7]. The environment is also compatible with other software design tools and languages such as VHDL, ANSI C/C++, and so on. This integrated design environment is the primary reason we chose this SDR platform for prototyping. With it, we could quickly reach an initial working version of our demonstration system and rapidly iterate on that design. LabVIEW significantly eases the physical layer design. By using the LabVIEW FPGA Module with the LabVIEW and PXI platform, we implemented necessary physical layer control/data (PDCCH/PDSCH) in a matter of months and had a preliminary proof-of-concept demo interfacing an NS-3 LENA LTE stack with a LabVIEW physical layer.

In this paper, we show preliminary integration results on how an LTE-based PHY layer is integrated with the open-source NS-3 LTE LENA stack [5]. We also propose an initial concept on how this testbed can be used to demonstrate an enhanced Inter-Cell Interference Coordination (eICIC) algorithm, for example almost blank subframe (ABSF) [8], which we will fine-tune using the SDN-based approach in the later stage of the project. However, the implementation of this testbed is generic enough to prototype a small-cell dense LTE network within the lab environment and demonstrate the performance of eICIC algorithms.

2. CROWD Architecture

The proposed CROWD architecture [1] uses the heterogeneity of dense wireless deployments, both in terms of radio condition and nonhomogeneous technologies. It offers tools to orchestrate the network elements to mitigate intrasystem interference, improve performance of channel-opportunistic transmission/reception techniques, and reduce energy consumption. An extremely dense and heterogeneous network deployment comprises two domains of physical network elements: backhaul and radio access network (RAN). The latter is expected to become increasingly heterogeneous not only in terms of technologies (for example, 3G, LTE, and WiFi) and cell ranges (macro-/pico-/femto-cells), but also at density levels (from macro-cell base-station (BS) coverage in under populated areas to several tens or hundreds of potentially reachable BS in hot spots). Such heterogeneity also creates high traffic variability over time because of statistical multiplexing, mobility of users, and variable-rate applications. For optimal performance, reconfiguration of the network element is required at different time intervals from very fast (few 10s of milliseconds) to relatively long (few hours), affecting the design of backhaul and RAN components. To tackle the complex problem of reconfiguration, we propose to follow an SDN-based approach for managing network elements, as shown in Figure 1. Network optimization in the proposed architecture is assigned to a set of controllers, which are virtual entities deployed dynamically over the physical devices. These controllers are technology-agnostic and vendor-independent, which allows full exploitation of the diversity of deployment/equipment characteristics. They expose a northbound interface, which is an open API to the control applications. We define control applications as the algorithm that actually performs the optimization of network elements, for example ABSF. The northbound interface does not need be concerned with either the details of the DAQ from the network or the

![Figure 1. Network Control Architecture](image-url)
enforcement of decisions. Instead, the southbound interface is responsible for managing the interaction between controllers and network elements.

We propose two types of controllers in the network (see Figure 1): the CROWD regional controller (CRC), which is a logical centralized entity that executes long-term optimizations, and the CROWD local controller (CLC), which runs short-term optimizations. The CRC requires only aggregate data from the network, and is in charge of the dynamic deployment and life cycle management of the CLC. The CLC requires data from the network at a more granular time scale. For this reason, CLC covers only a limited number of base stations [1]. The CLC can be hosted by a backhaul/RAN node itself, for example, a macro-cell BS, so as to keep the optimization intelligence close to the network. On the other hand, the CRC is likely to run on dedicated hardware in the network operator data center. Such an SDN-based architecture provides the freedom to run many control applications that can fine-tune the network operation with different optimization criteria, for example, capacity, energy efficiency, and so on. The CROWD vision provides a common set of functions as part of the southbound interface, which the control applications can use, for example LTE access selection [9] and LTE interference mitigation [10] to configure network elements of a dense deployment.

![Diagram of CROWD architecture](image)

**Figure 2A. Hardware/Software Architecture**

![Diagram of protocol stack architecture](image)

**Figure 2B. Protocol Stack Architecture**

**Testbed Architecture**

Figure 2 shows the general overview of the testbed architecture. The functions of the MAC and higher layer protocols (including CLC) run on a Linux computer. The protocol stack communicates with the PHY layer running on the NI PXI system over Ethernet using an L1-L2 API that is based on a small cell forum API [11]. We have implemented the complex high-throughput baseband signal processing for an “LTE-like” OFDM transceiver for the eNB and UE in LabVIEW FPGA using several NI FlexRIO FPGA modules because of the high-throughput requirements. We use the NI 5791 adapter module for NI FlexRIO as the RF transceiver.
This module has continuous frequency coverage from 200 MHz to 4.4 GHz and 100 MHz of instantaneous bandwidth on both TX and RX chains. It features a single-stage, direct conversion architecture, which provides high bandwidth in the small form factor of an NI FlexRIO adapter module.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subcarrier Spacing (f)</td>
<td>15 kHz</td>
</tr>
<tr>
<td>FFT Size (N)</td>
<td>2048</td>
</tr>
<tr>
<td>Cyclic Prefix (CP) length (N&lt;sub&gt;g&lt;/sub&gt;)</td>
<td>512 samples</td>
</tr>
<tr>
<td>Sampling Frequency (F&lt;sub&gt;s&lt;/sub&gt;)</td>
<td>30.72 MS/s</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>1.4, 3, 5, 10, 15, 20 MHz</td>
</tr>
<tr>
<td>Number of used subcarriers</td>
<td>72,180,300,600,900,1200</td>
</tr>
<tr>
<td>Pilots/Reference symbols (RS) spacing</td>
<td>Uniform (6 subcarriers)</td>
</tr>
</tbody>
</table>

Table 1. LTE-Like OFDM System Parameters

Introduction to LabVIEW-Based LTE-Like PHY

The current PHY implementation has only one antenna port (that is, SISO) supported per node with FDD operation. We have chosen to implement only the downlink transmitter/receiver and plan to show the performance of our algorithms in downlink direction. However, we use the same PHY layer in uplink. Thus, resulting in symmetric uplink based on OFDM. This greatly simplifies the PHY layer design and allows easy inter-connection to the MAC layer of protocol stack. However, in future we plan to implement SC-FDMA based uplink transport channel. The PHY layer implementation is real-time and accepts real-time configuration from MAC layer of protocol stack every TTI(1ms). We have designed the PHY modules to loosely follow 3GPP specifications, and hence referred to as an “LTE-like” system because our testbed is intended for research instead of commercial development. We describe main LTE OFDMA downlink system parameters in Table I. However, we omitted some components and procedures of a commercial LTE transceiver (for example, random access and broadcast channel) because they fall outside of the scope and requirements of our testbed. Only essential data and control channel functions are implemented.

We used the Xilinx CORE Generator library for the channel coding, FFT/IFFT, and filter blocks and developed custom algorithms in LabVIEW for all the other blocks. On the transmitter side (see Figure 3), physical downlink shared channel (PDSCH) and physical downlink control channel (PDCCH) transport blocks (TB) are transferred from the MAC layer and processed by each subsystem block as they are synchronously streamed through the system. Handshaking and synchronization logic between each subsystem coordinate each module’s operations on the stream of data. The fields of the downlink control information (DCI), including parameters specifying the modulation and coding scheme (MCS) and resource block (RB) mapping, are generated by the MAC layer and provided to the respective blocks for controlling the data channel processing. When the PDCCH and PDSCH data are appropriately encoded, scrambled, and modulated, they are fed to the resource element (RE) mapper to be multiplexed with RSs and primary and secondary synchronization sequences (PSS/SSS), which are stored in static lookup tables on the FPGA. Currently, the RB pattern is fixed and supports only one user. However, multiuser and dynamic resource allocation will be included in later versions. OFDM symbols are generated as shown in Figure 3 and converted to analog for transmission over the air by the NI 5791 RF transceiver.
Figure 4 shows the high-level block diagram of our OFDM receiver implementation. The NI 5791 RF transceiver receives the analog signal and converts it to digital samples for processing by the FPGA. This is followed by time synchronization based on the LTE PSS/SSS. The cyclic prefix (CP) is then removed, OFDM symbols are segmented out, and the carrier frequency offset (CFO) compensation module corrects for CFO impairments. Fast Fourier transform (FFT) is then performed on the samples and reference symbols are extracted for channel estimation and equalization. The equalized modulation symbols for the data and control channels are then fed to a separate decoder FPGA, which is connected to the receiver FPGA using a peer-to-peer (P2P) stream over the PCI Express backplane of the PXI chassis. Figure 5 shows the implementation of the downlink channel decoder. The first stage of the decoding process is to de-multiplex the symbols belonging to the PDSCH and PDCCH. The downlink control information (DCI) is then decoded from the PDCCH control channel elements (CCEs) and passed to the SCH turbo decoder module to decode the PDSCH data, which are finally sent to the MAC using the small cell forum API [11].

Figure 5. LTE-DL PDCCH and PDSCH Decoder FPGA Block Diagram

3. Introduction to Open-Source NS-3 LENA LTE Stack

We adopted the NS-3 LTE LENA simulator [5] as the framework for implementing the upper-layer LTE stack for our testbed platform. We used the protocols and procedures provided in the LTE LENA model library, which are generally compliant with 3GPP standards. Fairly complete implementations of the data-plane MAC, RLC, and PDCP protocols are provided along with simplified versions of the control-plane RRC and S1-AP protocols. Some core network protocols such as GTP-U/GTP-C and the S1 interface are partially implemented as well. Though used primarily by researchers as a discrete-event network simulator, NS-3 can also be configured to function as a real-time network emulator and interfaced with external hardware. By creating instances of LTE nodes on top of the NS-3 real-time scheduler, we can effectively emulate the eNB and UE stack. Also, by using the provided message passing interface (MPI) support, we can exploit parallelism to boost the performance of NS-3. By running each node instance in a separate thread or process, we can conceivably emulate multiple eNB, UE, and core network nodes all on the same multicore host, as represented in Figure 2. We have also modified the MAC-PHY interface of NS-3 stack to enable real-time support and interface it easily with NI/PXI Platform.

ABSF as an Example CLC Application

A major drawback of the LTE multicellular system is inter-cell interference. A promising mechanism to cope with this is ABSF [12], which, as shown in Figure 6, mitigates the inter-cell interference by assigning resources such that some base stations produce blank subframes, thus preventing their activity when the interference exceeds a threshold. Several centralized and distributed solutions have been already suggested in the literature to exploit the ABSF mechanism. Our proposal is designed to be implemented in the CLC, which is in charge of acquiring the user channel conditions and computing an optimal base station scheduling pattern in the available subframes using the SDN framework.
Here, we provide an overview of the proposed ABSF algorithm called base-stations blanking (BSB) [8], which has been designed for a particular scenario: content distribution over the whole network, such as road traffic or map updates. Specifically, the algorithm takes care of base station scheduling during the content injection phase. The base stations delegate a few mobile users to carry and spread content updates to the other users, through the device-to-device (D2D) interface, during the content dissemination phase. The content dissemination phase is deemed outside the scope of the algorithm, given that in a dense scenario, the content injection phase becomes the most critical part. The algorithm exploits the ABSF technique to minimize the time required by the base stations to achieve high spectral efficiency when performing packet transmissions during the injection phase.

The BSB algorithm provides a valid ABSF pattern by guaranteeing a minimum SINR for any user that might be scheduled in the system. The algorithm is implemented as a part of the control application LTE eICIC running on the CLC. The CLC filters per-user data and provides global per-eNB statistics to the control application. The CLC queries the downlink channel state information of any user through its serving LTE eNB. Based on the collected user channel state information, the application computes the appropriate ABSF patterns for each eNB, and sends them back to the CLC, which dispatches the patterns to eNBs via the southbound interface running on top of the standard X2 interface.

4. Preliminary Results

Here, we illustrate the initial results of our progress regarding the testbed integration and the simulation-based evaluation of ABSF algorithms since work began with the NI PXI and LabVIEW platform. Figure 7 shows the setup of our prototype, which contains an SISO OFDM downlink (eNB + UE). Our testbed includes five basic elements:

1. Laptop running LabVIEW for developing the SISO OFDM/LTE transmitter/receiver and also deploying the PHY layer code to the PXI systems
2. Laptop running the NS-3 eNB protocol stack in Linux
3. PXI system running the eNB transmitter baseband PHY on Virtex-5/Kintex-7 FPGAs with an NI 5791 front-end module functioning as the transmitter DAC and RF upconverter
4. PXI system running eNB receiver baseband PHY on Virtex-5/Kintex-7 FPGAs with NI 5791 used as the downconverter and ADC
5. Laptop running the NS-3 UE protocol stack in Linux

In the future, we will have several more PXI systems to emulate multiple eNBs/UEs in a dense wireless lab network setting to demonstrate the performance of PHY/MAC cross layer interference mitigation algorithms like ABSF within the CROWD framework.

Testbed Integration

We currently have a working, validated implementation of the LTE PDSCH channel running on the testbed. Figure 8 shows the LabVIEW virtual instrument running the transmitter and receiver with QPSK modulation and a fixed transport block size. We have also adapted the NS-3 LENA LTE protocol stack so that it can be interfaced with the SISO OFDM PHY.

The encoding of NS-3 MAC messages for over-the-air transmission is proprietary and already completed. The current adaptation of the LENA stack supports either placing SISO OFDM PHY in a loop-back mode with eNB and UE running within a single instance of NS-3 or eNB and UE running in two separate instances of NS-3.
Figure 8. Snapshot of TX/RX VI Running SISO OFDM LTE PHY

Figure 9. Compare the subframe use per base station between the standard scheduling approach without eICIC mechanism and the CLC approach using the BSB algorithm.

ABSF Simulations

We now illustrate some initial simulation results of the proposed LTE eICIC application which affect the performance of D2D-assisted content update distribution. Let us assume that the contents are split into several chunks. Thus, LTE eNBs adapt their transmission based on any single chunk. In Figure 9, we compare the performance of the proposed control application against the standard scheduling approach. The base station activities are depicted in the graph as a function of the chunks delivered in the network. The base stations remain active for about 10 percent of the total number of subframes, showing an efficiency gain up to 10 times against the standard scheduling approach without any kind of MAC enhancements. Note that the control application uses the first chunks to reach a steady-state for the next subframes. When network changes occur, the control application adjusts the ABSF patterns accordingly.

5. Conclusion

In this paper, we showed how the NI PXI platform based on LabVIEW can be used for advanced wireless LTE prototyping to demonstrate end-to-end link. Because the design of the testbed is based on an open-source protocol stack and modular hardware from the NI PXI platform, it is much more cost-effective and flexible compared to other commercial LTE testbed solutions, especially for demonstrating research ideas for next-generation wireless systems. The graphical system design based on LabVIEW and its integrated environment were key reasons we achieved significant results and made good progress with testbed integration. We also showed preliminary results for testbed implementation of LTE PHY layer and its integration with NS-3. The next step is to complete the implementation of PDCCH and also the implementation of small cell forum API, to allow integration of LTE PHY layer with NS-3 in real time. In the end, we aim to demonstrate the performance of our proposed ABSF algorithms in a lab setting using multiple cells emulated on several PXI systems.

6. Authors

Rohit Gupta, Bjorn Bachmann and Thomas Vogel
National Instruments
7. Acknowledgement

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 318115 (CROWD).

8. References


The registered trademark Linux® is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.