Design and Implementation of 4G LTE Components - eNodeB and UE on SDR platform using srsLTE

[1]T M Kavyashree, [2]L. Swarna Jyothi, [3]Umashankar Shetty C

[1,2]Dept of ECE, RRCE Bengaluru, India
[3]Manager-Services Tenet Technetronics, Bengaluru, India

Abstract-Wireless Communication is one of the important domains of research in communications. Mobile communication is growing towards faster generation like 4G LTE-Fourth-Generation Long Term Evaluation. 4G LTE aims at providing a comprehensive IP solution wherein voice, data and streamed multimedia can be provided n to the users simultaneously on an "Anytime, Anywhere" basis at higher data rates than previous generations. The three main components of 4G systems are eNodeB, UE-User Equipment and EPC-Evolved Packet Core. This paper discusses the concepts and importance of SDR platform. Open source software - srsLTE is briefly discussed. Soft eNodeB and UE are implemented using hardware USRP. This work mainly aims at transmitting data from a soft eNodeB to UE using srsLTE software.

Keywords-LTE, SDR, UE, eNodeB, srsLTE, USRP.

I. INTRODUCTION

In the fast growing communication world, experiments and researches are always blooming. Thus Test beds are an important platform for experimental research and prototype development. They enable the researchers to test, validate and assess the performance of new technologies for wireless communication [1]. For 4G LTE, a test bed usually includes one or many User Equipments (UEs), one base station (eNodeB) and an Evolved Packet Core (EPC). However EPC is of minimal use. These components are disturbed by commercial off-the-shelf (COTS) hardware solutions. They are typically expensive, but the performance is excellent and the functionality is extensively validated.

To solve the problems associated with the current technology, researchers require adding new features to the available standards or modifying some of its hardware parts. For example, for the 5G research [2], several groups are exploring how new waveforms such as GFDM (Gaussian Frequency Division Multiplexing) can be fit in the current LTE resource grid and physical layer procedures. Another example is IoT (Internet of Things), where power and cost constraints may require simplified waveforms and protocols to be introduced. Another research problem requires instrumentation [3], where the entire network stack from PHY to IP needs to measure the impact metrics. Such metrics includes the channel Doppler spread, the interference, the number of HARQ retransmissions, and the number of turbo decoder iterations or the power headroom. Such modifications, customizations or instrumentation using the COTS hardware are extremely difficult and expensive.

Software-Defined Radio (SDR) [9] is a popular platform for implementing radio equipment on software basis. This can be done using low-cost general purpose computers and radio frontends. Within few years, it's gaining popularity as an ideal tool to build close-to-reality test beds for experimental researches. The main advantage of SDR is that if the researchers have access to the SDR application code, as it is usually open source, the test bed can be easily modified and instrumentation of the stack becomes simpler. In terms of performance, the open source SDR test beds provide a good flexibility. This flexibility and openness can be much more valuable in solving many research problems.

The available open source LTE SDR software available today are Eurecom's Open Air Interface (OAI) and openLTE [7]. OAI provides a standard-compliant implementation of a subset of Release 10 LTE for UE, eNB, MME, HSS, SGw and PGw on standard Linux-based computing PC (Intel x86 PC architectures). The software can be used in conjunction with standard RF laboratory equipments that are available in labs of National Instruments or Ettus USRP and PXIe platforms. openLTE runs with the Ettus Research B2x0 USRP and provides eNB, MME and HSS functionalities. openLTE code is well organized, documented and easy to customize or modify. However, openLTE is still incomplete and many features are still unstable or under development. Also, it does not provide an UE implementation. OAI on the other hand is comparatively very complete and provides very good performance. But, the code structure is complex and difficult for a user external to the project to modify or customize.

II. SOFTWARE DEFINED RADIO

Over the past decade, software-defined radios (SDRs) [8] have an increasingly prevalent aspect of wireless communication systems. SDR differs from traditional hardware radios where in radio protocols are implemented using static electrical circuit. SDRs implement significant aspects of physical radio protocol using software programs running on a host processor. Because they use software to implement most of the radio functionality, SDRs are much more easily modified, edited, and upgraded than their hardware-defined counterparts. Consequently, researchers and developers have been developing previously hardware-defined radio systems within software.

In the past few decades, the field of wireless communications has been developing and advancing at a rapid pace. Nearly all new electronic devices implement some sort of wireless communications, be it in the form of Wi-Fi, Bluetooth, or cellular technologies like CDMA or LTE. Each of these different radio systems has its own specific protocols. Consequently, these different radio systems had to be implemented using hardware configurations.

Hardware radios use physical components which are not easily modified. Consequently, this static nature gives hardware radios several limitations. First, needing different hardware setups for each radio technology can use significant amounts of space, especially if a particular setup needs several different radio technologies. Second, implementing separate hardware protocols becomes expensive to systems needing to use many different radio standards [6]. Cellular phone technology provides a key example of this phenomenon. In cellular phone technology, entire nations and regions have attempted to standardize the radio protocol; however, cell phones still need to support old standards still in use and alternate standards in different regions so a single phone can operate in many locations. Current hardware limitations cause cell phones to have separate physical systems for each communication standard which increases both the size and cost of cell phones. Third, hardware radios are not easily updated when new technology is developed. Radio technology and protocols are constantly evolving to become faster and more advanced. Thus, a protocol used today could be obsolete in just a few years. Under hardware-based radio schemes, systems would be unusable whenever a new protocol is developed. Because of the limitations inherent in static hardware radio systems, a different kind of radio system has been developed within the past few years.

To solve the hardware problem, engineers decided to implement parts of the radio using software rather than hardware. Using software rather than hardware to implement some stages of a radio system enables a radio to be more easily configured, modified [4], and developed for multiple systems. This new form of radio implementation came to be known as software defined radio (SDR) or software radio. The goal of SDRs is to implement fully functional radios in one system that previously needed multiple systems.

III. 4G LTE

Mobile systems focus on seamlessly integrating the existing wireless technologies including GSM, wireless LAN, and Bluetooth. 4G [7] systems supports comprehensive and personalized services, providing stable system performance and quality services. 4G is a Mobile multimedia, anytime anywhere, Global mobility support, integrated wireless solution, and customized personal service network system. The acronym Long Term Evaluation (LTE) is 4G (Fourth Generation) wireless communication standards which are developed by 3rd Generation Partnership Project (3GPP) in and around 2010. 4G is used broadly to include several types of broadband wireless access communication systems along with cellular telephone systems.

A 4G cellular system must have target peak data rates of up to approximately 100 Mbit/s for high mobility such as mobile access and up to approximately 1 Gbit/s for low mobility such as nomadic/local wireless access, according to the International Telecommunication Union [ITU] requirements. Scalable bandwidths up to at least 40 MHz should be provided. A 4G system is expected to provide a comprehensive and secure all-IP based solution where facilities such as IP telephony, ultra-broadband Internet access, gaming services and High Definition Television (HDTV) streamed multimedia may be provided to users.

The infrastructure and the terminals of 4G will have almost all the standards from 2G to 4G implemented. The infrastructure for 4G will be only packet-based (all-IP). But there is suggestion to have an open Internet platform.

IV. srsLTE : AN OPEN SOURCE LIBRARY

srsLTE is an open source library for the PHY layer of LTE Release 8. It is designed for maximum modularity and code reuse with minimal inter module or external dependencies. The code is written in ANSI C and makes extensive use of Single Instruction Multiple Data (SIMD) operations, when available, for maximum performance. In terms of hardware, the library deals with buffers of samples in system memory thus being able to work with any RF front-end. It currently provides interfaces to the Universal Hardware Driver (UHD), giving support to the Ettus USRP family of devices.

The aim of the library is to provide the tools to build LTE-based applications such as a complete eNodeB or UE, an LTE sniffer or a network performance analyser. The current features provided by the library are:

- LTE Release 8 compliant in FDD configuration;
- Supported bandwidths: 1.4, 3, 5, 10 and 20 MHz;
- Transmission mode 1 (single antenna) and 2 (transmit diversity);
- Cell search and synchronization procedure for the UE;
- All DL channels/signals are supported for UE and eNodeB side: PSS, SSS, PBCH, PCFICH, PHICH, PDCCH, and PDSCH;
- All UL channels/signals are supported for UE side: PRACH, PUSCH, PUCCH, and SRS;
- Highly optimized turbo decoder available in Intel SSE4.1/AVX (+100 Mbps) and standard C (+25 Mbps); and
- MATLAB and OCTAVE MEX library generation for many components.

V. srsUE : SOFT UE IMPLEMENTATION

srsUE is a software radio LTE UE covering all layers of the network stack from PHY to IP layer. It is written in C++ and builds upon the srsLTE library which provides the PHY layer processing. For some security functions and RRC/NAS message parsing, it uses some functions from the openLTE project. Running on an Intel Core i5 or i7 processor, srsUE achieves more than 60 Mbps downlink with a 20 MHz bandwidth in SISO configuration. Apart from the features listed above for the srsLTE library, srsUE provides the following additional features:

- supporting MAC, RLC, PDCP, RRC, NAS and GW layers;
- Soft USIM supporting Milenage and XOR authentication;
- Detailed log system with per-layer log levels and hex dumps;
- MAC layer Wireshark packet capture;
- Command-line trace metrics;
- Detailed input configuration file; and
- Virtual network interface (i.e. tun/tap device) created upon network attach.

A configuration file is provided to set parameters such as the downlink carrier frequency and log or packet capture options, making the software easy to use. The UE starts setting the USRP sampling rate to 1.96 MHz, in order to capture the synchronization PBCH signals. Once synchronization and MIB decoding is successful, it reconfigures the sampling rate to the appropriate sampling rate for the LTE signal bandwidth.

Next, the UE attempts an attachment by sending a PRACH sequence and if the correct response is received, it continues the connection setup procedure. Following successful network attachment, a new virtual network interface is created in the system and the user can then establish IP sessions with the eNodeB [5] network.

This instrumentation and the Wireshark capture capabilities make srsUE an ideal tool for many applications, including cross-layer performance analysis, education and prototype development. srsUE opens the insights of the LTE stack to the user: the real-time traces, logs and packet captures can be used to correlate user experience in video streaming or web surfing with the signal quality or any other metric inside the stack, for example. srsUE classes are organized by layers, one class per stack layer. Each class provides a separate clean interface to any other class that makes use of it, which is used for message passing between layers of the stack.

VI. TOOLS USED

Hardware requirements:

- USRP B200/210
- Antennas
- 2 PCs/Laptops: minimum of 4GB internal RAM
- PCle /1Gigabit /10Gigabit Ethernet interface cards.
- USB 3.0 cable

Software requirements:

- Operating system: ubuntu 12.04 amd64
- USRP hardware driver (UHD)
- srsGUI
- srsLTE
- srsUE
- Coding language: C++/python.

VII. METHODOLOGY

The proposed method on implementing a 4G LTE system, aims at designing the eNodeB (evolved NodeB) and UE (User Equipment) on SDR platform. The software used will be srsLTE, which is free and open-source software. To provide the RF front-end support to the system, Universal Software Radio Peripheral (USRP) – Bus series is used. Thus this proposed system provides a software platform to transmit and receive information in the form of packets at 4G speed and environment.

SDR eNodeB consists of a USRP B210 or X300 as a RF front-end, connected to a core i5 CPU via an USB 3.0 cable. Host PC with SDR eNodeB is known as soft eNodeB. Soft eNodeB accepts a configuration file that specifies eNodeB information and also how this soft eNodeB will be performing the uplink and downlink transmission with the User Equipment (UE).

srsLTE also has a separate source code that performs the function of User Equipment in 4G system. This soft UE can also be implemented on another PC with linux ubuntu 12.04 OS. USRP B200/210 acts as the RF front-end for soft UE. Thus running eNodeB and UE on different PC will create a 4G environment where in we can transmit random data packets and video from eNodeB to UE using USRP as RF front end.

Once both eNodeB and UE are successfully running, the eNodeB will start transmitting the random data in packet format, and then the UE will be able to detect the cell to which it is attached. That is, the PC acting as UE will now display the attached cell ID, details of MIB (Master Information Block) and SIB (System Information Block), number antennas used, number of Physical Resource Blocks that are used in transmission and Signal to Noise Ratio (SNR) on the monitor screen. Fig.1. shows the setup of the proposed method.

Fig.1. Setup of Proposed Method

VIII. RESULTS

At the transmitter end, using different modulation coding scheme value (MCS), eNodeB chooses various modulation techniques like BPSK, QPSK, 16QAM and 64QAM. At the receiver end, UE decodes the SIB and MIB information and searches the cell in particular frequency. Once the cell is identified, UE virtually attaches to the eNB and receives the information transmitted by it.

By repeated experimentation it is seen that using 64QAM the reception is good and the constellation diagram also resembles 64QAM. UE on reception displays the various parameters like SNR (signal to noise ratio), BLER (Block error rate), CFO (Carrier Frequency Offset) and SFN (System Frame Number). Also using srsGUI, UE plots equalized symbol graph for received symbols and control information, PSS cross correlation value, channel response magnitude and argument values. For various modulation techniques the output parameters values are as follows:

TABLE1: Output of UE for different modulation

Modulation type	**SNR (Db)**	**BLER (%)**	**CFO (KHz)**	**SFN**
64QAM	-5.6	40.06	+0.62	632
16QAM	-5.8	53.94	-2.34	156
BPSK	-6.9	100	-0.60	849

From the above table since the BLER for 64QAM is lesser, this implies that it's the better modulation techniques compared to others. Thus a satisfied Downlink communication can be established using srsLTE.

IX. CONCLUSION

This work presents an introduction to the wireless communication system. SDR is a boon for the emerging wireless communication technology. It presents an open-source, modular and fully compliant with LTE Release 8 platform that allows for LTE extension and experimentation. It also describes the methodology to implement srsLTE library as well as the srsUE and srsGUI. It also explains the merits of srsLTE software. This paper thus proposes a new method to implement 4G eNodeB and UE.

REFERENCES

[1] Ismael Gomez-Miguelez, Andres Garcia-Saavedra, Paul D. Sutton, Pablo Serrano, Cristina Cano, Douglas J. Leith, “srsLTE: An Open-Source Platform for LTE Evolution and Experimentation” Cornell University, 1062.04629v1, , pp: 1-7, 15 Feb 2016.

[2] Navid Nikaein, Raymond Knopp, Florian Kaltenberger, Lionel Gauthier, Christian Bonnet, Dominique Nussbaum, and Riadh Ghadda, “Open Air Interface 4G: an open LTE network in a PC”. Eurecom, volume 4. Pp: 1-3, 2015.

[3] George D. Sworo, Cem Sahin, Kapil R. Dandekar, Moshe Kam, “Towards Integrating Pattern Reconfigurable Antennas in WiMAX/LTE Radios”. Wireless and Microwave Technology Conference (WAMICON), IEEE. Pp: 1-5, 2015.

[4] Zhiping Chen, Jun Wu, Tongji University Shanghai, China, “LTE Physical Layer Implementation Based on GPP Multi-core Parallel Processing and USRP Platform”. IEEE. pp: 197 – 201, 2014

[5] Muhammad Amar Zulfiqar, Ahmad Talal Riaz, Muhammad Bilal, Ajreen Qammar, “Paper Development of L1 Components for eNodeB of LTE Advanced”. International Journal of Technology and Research (IJTNR). Pp: 119 -123, 2014

[6] Sara M. Hassan, A. Zekry, M. A. Bayomy G. Gomah, “Software Defined Radio Implementation of LTE Transmitter Physical Layer”. International Journal of Computer Applications (IJCA), Volume 74– No.8. pp: 41-46, 2013.

[7] Ngoc-Duy Nguyen, Raymond Knopp, Navid Nikaein, Christian Bonnet, “Implementation and Validation of Multimedia Broadcast Multicast Service for LTE/LTE-Advanced in OpenAirInterface Platform”. Department of Mobile Communications, EURECOM Sophia Antipolis, France, IEEE International Workshop on Performance and Management of Wireless and Mobile Networks. Pp: 70-76, 2013

[8] Sruthi M B, Abirami M, Akhil Manikkoth, Gandhiraj R, Soman K P, “ Low cost digital transceiver design for Software Defined Radio using RTL-SDR”. IEEE. Pp: 852-855, 2013.

[9] Mehul R. Naik1, C. H. Vithalani, “The Software-Defined Radio Is Now A Reality”. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering (IJAREEIE). Pp: 2923-2927, 2013