

Performance Study of LTE Experimental Testbed using OpenAirInterface

Chun Yeow Yeoh, Mohammad Harris Mokhtar, Abdul Aziz Abdul Rahman, Ahmad Kamsani Samingan

Signal Processing Lab

Telekom Research & Development Sdn. Bhd.

Cyberjaya, Selangor, Malaysia

yeohchunyeow@ieee.org, harris@tmrnd.com.my, abd_aziz@tmrnd.com.my, drkamsani@tmrnd.com.my

Abstract— The next generation, 5G wireless access, is going to support a wide range of new applications and use cases, with the capabilities including very high achievable data rate, very low latency, ultra-high reliability and the possibility to handle extreme device densities. To avoid the costly deployment, operation and maintenance of future mobile network, Radio Access Network (RAN) virtualization or Cloud RAN is the answer to the problem. The idea is to move the baseband processing to the data center and run the RAN L1, L2 and L3 protocol layers using the commodity hardware, such as high-performance general purpose processors. An open source software-based LTE implementation, such as OpenAirInterface (OAI), is definitely accelerating the RAN cloudization and also realizing the possibility of low cost LTE network deployment in the future. In this paper, we describe the OAI LTE implementation emphasizing on the user plane data flow. We have successfully emulated over-the-air transmission for 1 UE and 1 eNB LTE network supporting both FDD Band 5 and TDD Band 38. We have also performed a thorough profiling of OAI, in terms of execution time, on the user plane data flow. Our results could be served as the reference for future optimization by open source community.

Keywords— LTE, Open Source, Software Defined Radio, Testbed, Profiling.

I. INTRODUCTION

Long-Term Evolution (LTE), commonly marketed as 4G LTE, has gradually deployed worldwide since its first introduction in Scandinavia back in year 2010. Until the year of 2015, 124 countries have deployed the LTE service and 18 countries have scheduled for LTE service [1]. Decoupling the network functions from the underlying hardware, known as Software Defined Networking (SDN) and Network Functions Virtualization (NFV), has played a critical role in the next generation of telecommunication, providing flexibility and more responsive central control of network traffic service providers. The SDN/NFV exercises have mainly taken place at the core network equipments. Cellular system is still slow in moving towards this direction and thus locks to expensive HW/SW platforms. However, if we are able to build and support an open cellular ecosystem, we could one day leverage the commodity hardware or general purpose processor for open LTE system for future 5G. One of the open cellular systems is OpenBTS [2], a 2.5G GSM system that is able to support

Software Defined Radio (SDR) platforms from various vendors. Endaga [3] has started to commercialize this solution for rural connectivity with affordable charges. Another programmable wireless prototyping platform, WARP [4] is also available with real-time FPGA implementation of 802.11 supporting both 2.4GHz and 5GHz radio frontend which could be useful for studying the coexistence of Licensed Assisted Access (LAA) and 802.11. Another fully software-based LTE solution running in commodity PC is closed-source Amari LTE 100 [5] from Amarisoft. Besides these, Software Radio Systems Limited has also launched its high performance LTE library for SDR for physical (PHY) layer, named as srsLTE [6].

Lastly, the OAI wireless technology platform is the first open source software-based implementation of the LTE system spawning the full protocol stack of 3GPP standard both in Evolved Universal Terrestrial Radio Access (E-UTRAN) and Evolved Packet Core (EPC) [7] [8]. The transceiver functionality is realized via a software radio front end connected to a host computer for processing. The supported SDR platforms include the B210 from Ettus Research, ExpressMIMO2 from Eurecom and etc. The OAI software is written in standard C and preferable to run in low latency Linux kernel. We have setup a LTE experimental testbed using OAI and targeting the platform as possibility of low-cost 4G deployment in mind. In this paper, we first demonstrate that OAI is feasible for us to run both FDD Band 5 and TDD Band 38 LTE network comprising User Equipment (UE), eNodeB (eNB) and EPC. We also conduct the performance study of the user plane data flow by using code profiling technique to better understand the required code optimization for faster code execution and less memory occupation.

The remainder of this paper is organized as follows. Section 2 consists of the discussion of OAI emphasizing on the IP packets at the user plane from top to bottom layers of the LTE protocol stack. The scheduling of both LTE TDD/FDD frame structure is also briefly discussed. Section 3 describes the experimental setup of our LTE testbed using OAI for non-3GPP compliant and 3PP compliant scenarios. The UE tracing procedures and the method used for measuring the execution time in the OAI software components are also presented. Section 4 discussed the results obtained from the code profiling. Finally, section 5 provides concluding remarks.

II. OPENAIRINTERFACE SUPPORTING LTE FDD/TDD

OpenAirInterface (OAI), as an open source experimentation and prototyping platform for LTE, comprises the entire LTE protocol stack, including standard-compliant implementations of the 3GPP LTE access stratum for both eNB and UE and a subset of the 3GPP LTE evolved packet core protocols. OAI consists of *oaisim* to simulate a complete LTE network, and the *oaisim* is running the protocol stacks that will be used for emulation as well, except the PHY and the radio channels are fully emulated known as PHY abstraction model [9].

The OAI eNB application consists of two main Portable Operating System Interface (POSIX) threads, *eNB_thread_rx* and *eNB_thread_tx* that run using the earliest deadline first scheduling [10] supported in low latency Linux kernel. The scheduling attributes are set to 10ms of *period* equivalent to the 10ms radio frame, 1ms of *deadline* equivalent to 1ms subframe and the *runtime* calculated based on the maximum Downlink (DL) and Uplink (UL) supported Modulation and Coding Scheme (MCS), Resource Blocks (RBs), protocol processing time and etc. In the other hand, the OAI UE application consists of three main POSIX threads, *UE_thread_tx*, *UE_thread_rx* and *UE_thread_synch*. The additional POSIX thread is used for cell synchronization, including detecting Primary and Secondary Synchronization Signal (PSS/SSS) as well as reading the Master Information Block (MIB). Another POSIX thread, *eNB_thread* or *UE_thread*, is spawned to read from and write to the PHY hardware via wrapper function calls, *trx_read_func* and *trx_write_func* respectively.

In OAI software implementation for eNB, the *eNB_dlsch_ulsch_scheduler* is the function that handles the UL and DL scheduling of the PHY radio frame in unit of 1ms Transmission Time Interval (TTI) either using the TDD or FDD frame structure. We are using FDD Band 5 for coverage purpose due to its lower frequency band and TDD Band 38 for capacity enhancement or hotspot deployment due to its higher frequency band. The frame structures for both LTE TDD and FDD are illustrated in [11]. We are not going to describe further on this since detailed description has been documented at [12]. Instead, we describe how the user plane data flow or IP data flow is being transferred from EPC to the OAI eNB application before sending out over-the-air using the SDR platform and the other way round.

For non-3GPP compliant OAI eNB or UE application, *nasmesh* is the kernel module designed for interaction between the IP layer application and OAI as illustrated in Figure 1. Non-3GPP compliant in our case means that the OAI eNB is not connected to either commercial EPC or OAI EPC. Meanwhile, *rb_tool* is the utility to add or tear down the radio access bearer with differentiated services. The network device registered to the kernel is known as *oai*. Referring to Figure 2, when the user first executes the network utilities, such as ping, the first function call is the *nas_hard_start_xmit* function. Subsequently, a series of DL function calls in the kernel space are executed and pushing the IP data to the OAI eNB or UE application. The *recvmsg* function in the *pdcp_fifo_read_input_sdu* function in the user space captures the IP data. The IP data is then sent out to the PDCP and RLC layers via the *pdcp_data_req* function and then processed by

the DL scheduler in the user space. Later on, the PHY hardware is invoked to send out the data in the kernel space. LTE data received by the PHY hardware in the kernel space was processed by the MAC, RLC and PDCP layers in the user space; *sendmsg* in the *pdcp_fifo_flush_sdu* function then sends the data to *nasmesh* kernel module. This invokes a series of UL function calls in the kernel space and later the *netif_rx* function in the *nas_COMMON_receive* function pushes the data up to the upper layer which is the IP layer.

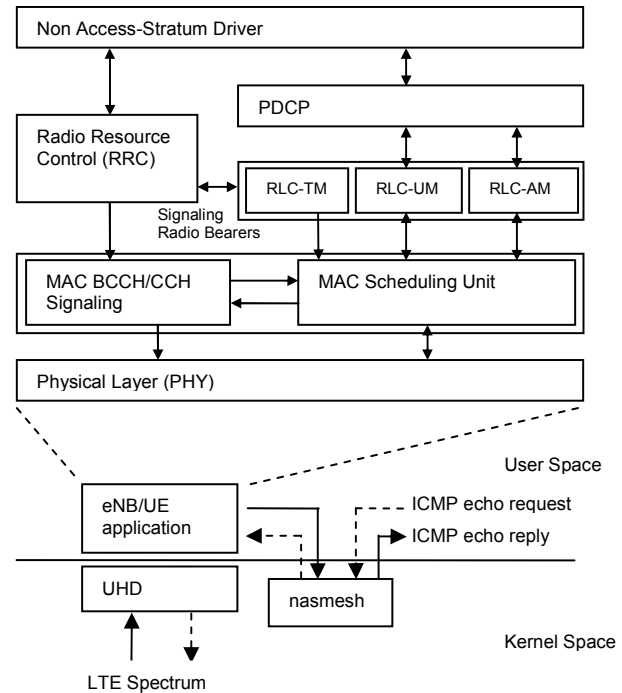


Fig. 1. Non-3GPP Compliant User Plane Data Flow in OAI

For 3GPP compliant OAI eNB application, a GPRS Tunneling Protocol (GTP) for user plane is required. As shown in the grey colour box of Figure 3, the *xt_GTPUSP* kernel loadable module is inserted in the Packet Data Network Gateway (PGW) or the EPC to add and remove the GTP User Plane (GTP-U) header after matching specific rules, such as destination and source IP addresses. The incoming GTP-U data from PGW triggers the *udp_eNB_receiver* function and also a series of GTP function calls, including the *gtpv1u_eNB_process_stack_req* function. Lastly, the *pdcp_data_req* function is triggered for LTE Downlink Shared Channel (DLSCH) processing. For LTE Uplink Shared Channel (ULSCH) processing, once the PDCP data is received and processed by PHY, MAC and RLC layers, the *gtpv1u_new_data_req* function in the *pdcp_fifo_flush_sdu* function in OAI eNB invokes a series of function calls in the user space and lastly the *gtpv1u_eNB_send_udp_msg* function to send out the GTP data to PGW. The UDP connection is setup between eNB and PGW using port number 2152 and this port number is defined using the EPC and eNB configuration files.

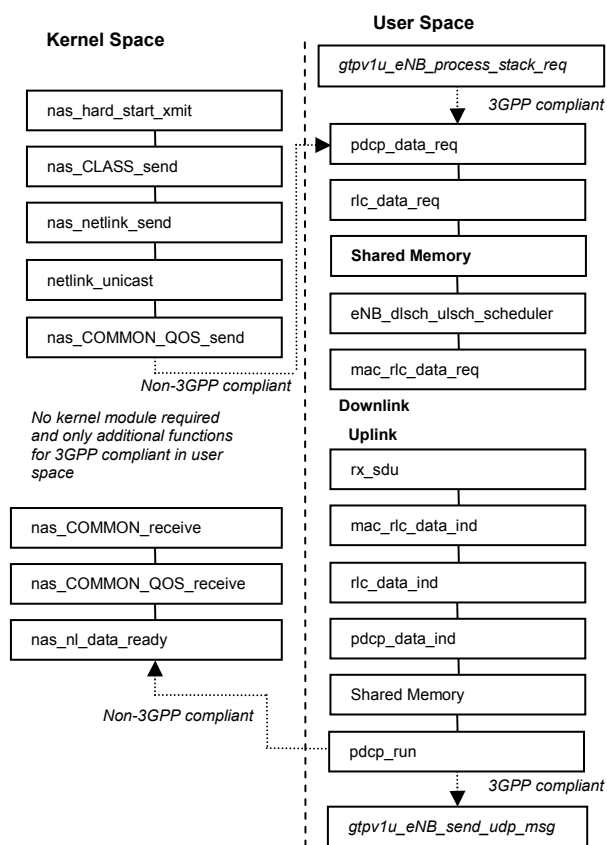


Fig. 2. Call Graph for User Plane Data Functions in OAI eNB

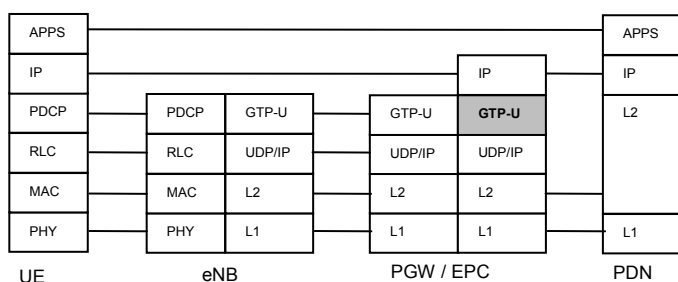


Fig. 3. 3GPP Compliant User Plane Data Flow in OAI

III. LTE EXPERIMENTAL TESTBED WITH UE TRACING AND PROFILING METHOD

As illustrated in Figure 4, the LTE experimental testbed consists of one unit of OAI UE, two units of OAI eNBs and one unit of EPC/Home Subscriber Server (HSS). All of them run in the separate Linux-based Intel x86-64 machines comprising 4 cores and 2 threads in each machine with Intel i7 processor core at 3.6GHz. The OAI software version used by our LTE experimental testbed is the modified SVN version of r7832 for the OAI UE and the SVN version of r7890 for the OAI eNB. The patches for OAI UE can be found at [13] [14]. At first, the OAI UE is connected with two units of OAI eNBs using the wideband power splitter/combiner to avoid interference from nearby commercial eNBs. The OAI EPC/HSS is connected with two units of OAI eNBs using

Ethernet interface via a L2 switch. The complete setup of the non-3GPP compliant testbed can be found at [13]. Meanwhile, 3GPP compliant testbed with EPC/HSS support can be found at [14]. The two tutorials [13][14] are conducted using over-the-air transmission with appropriate duplexer and antennas attached to both UE and eNB. The tutorials have not included the extra unit of eNB that we have added to our LTE experimental testbed. In our LTE experimental testbed, both OAI eNBs are having different Physical Cell Identity (PCI) configured using *Nid_cell* in the eNB configuration files. The reason of having an extra unit of eNB is that one is running FDD Band 5 and another unit is running TDD band 38.

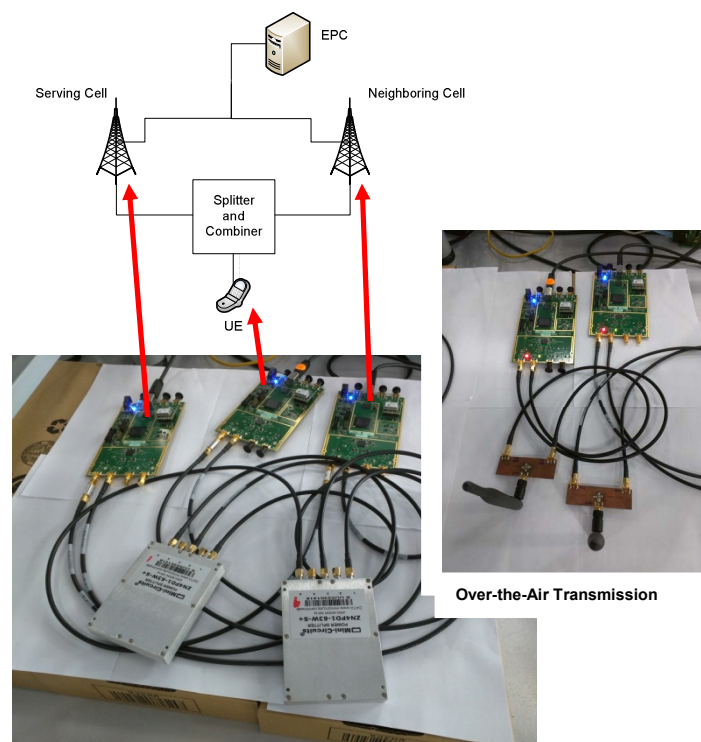


Fig. 4. LTE Experimental Testbed utilizing OAI

A. Monitor the OAI UE State Transition

One of the methods that we use to trace whether the OAI UE is successfully attached to the OAI eNB is by tracking the UE Radio Resource Control (RRC) state transition. The following RRC states are observed in the OAI software: idle → System Information (SI) received → connected → reconfigured. Once the OAI UE is initialized, it moves to the idle state. After successfully decoded the LTE system information SIB1/SIB2/SIB3, it goes to the SI received state. Then, the UE state moves to connected after the Signaling Radio Bearer (SRB) 1 on Dedicated Channel (DCCH) has been setup. The UE enters into the reconfigured state when it has successfully decoded the RRC Connection Reconfiguration message and generated the RRC Connection Reconfiguration Complete message to the eNB. After reaching reconfigured state, we can start the DL and UL data transfer between eNB and UE using the dedicated radio bearer. This is applied to both 3GPP compliant and non-3GPP compliant scenarios. But

for 3GPP compliant scenario, we can also track that the status of EPS Session Management (ESM) is moving from Bearer Context Inactive to Bearer Context Active once the UE is attached to the eNB.

TABLE I. FUNCTION NAMES IN OAI FOR MEASURING EXECUTION TIME

LTE Protocol and Network Stack	User Plane Data	
	Tx Function	Rx Function
PHY	phy_eNB_dlsch_encoding phy_eNB_dlsch_scrambling phy_eNB_dlsch_modulation	phy_eNB_ulsch_decoding
MAC	mac_dlsch_preprocessor mac_schedule_dlsch mac_fill_dlsch_dci	macface_rx_sdu
RLC	mac_rlc_data_req rlc_data_req	mac_rlc_data_ind
PDCP	pdcp_data_req	pdcp_data_ind
GTP-U	gtpv1u_process_udp_req	gtpv1u_process_tunnel_data_req
UDP/IP/L2/L1	N/A	udp_enb_task
SDR – UHD	trx_write	trx_read

B. Code Profiling using GTKWave Analyzer

To do measurement on processing delay or execution time for LTE protocol and networking stack, we use the GTKWave, a waveform viewer which reads VCD file generated by the OAI software. The GTKWave can be installed in Linux machine and run using the command *gtkwave*. Table II is the function names related to the LTE protocol and network stack defined by the OAI software. An entry point and an exit point are inserted for the function call under the profiling to estimate the execution time using the *clock_gettime* system call. For non-3GPP compliant scenario, the execution time for *nasmesh* kernel module is less than 1 microsecond and thus negligible based on our investigation. So, we have emphasized our study on the execution time required by the function calls in the user space. We have used 6 different PDCP payload sizes for our measurement and performing the profiling on both FDD Band 5 and TDD Band 38. The emulation parameters can be summarized at Table II.

TABLE II. EMULATION PARAMETERS USED FOR PERFORMANCE EVALUATION

Parameters	1 st eNB's value	2 nd eNB's value
Duplexing Mode	FDD	TDD
Frequency Band	Band 5	Band 38
UL/DL Configuration	N/A	Configuration 3 (6DL, 3UL, 1 Special Subframe)
# UE	1	1
Mobility	Static	Static
# Available RBs	25	25
PDCP Payload Sizes	84, 140, 252, 476, 924, 1500	84, 140, 252, 476, 924, 1500
RLC Mode	Unacknowledged Mode (UM)	Unacknowledged Mode (UM)

IV. MEASUREMENT RESULTS AND DISCUSSION

As illustrated in Figure 5, the CPU spends most of its time on PHY channel processing no matter it is UL or DL and no matter it is FDD or TDD. At the same time, the execution time required for PHY DL channel processing is directly proportional to the PDCP payload size as shown in Figure 6. More execution time is required for DL if the PDCP payload is getting larger. However, the execution time for PHY UL channel processing has not affected even though the PDCP payload is increased. We also notice that the MAC processing for DL and UL are almost similar even though the PDCP payload size is getting larger. This is due to the main task of the MAC processing is to assign the transmission resources or RBs to UEs. Since there is only 1 OAI UE to be scheduled in our emulation, all the available RBs will be allocated to this UE and no additional processing required for sorting out the UE based on priority.

We have further investigated the DLSCH encoding by inserting additional signal dumps in the source code for GTKWave analysis and running the emulation using PDCP payload size of 1500 bytes. Based on the results, we found out that the most time consuming module is the turbo encoding which constitutes 74% of CPU execution time and followed by 17% from the rate matching module in the *phy_eNB_dlsch_encoding* function. PDCP Service Data Unit (SDU) can scale up maximum 8188 bytes as defined by [15]. Thus, optimization on turbo encoding is necessary to deal with the large PDCP payload. We would like to point out that the OAI software is already implemented with SIMD-optimized for Streaming SIMD Extensions 4 (SSE4) in integer arithmetic. Also, the modulation mapper or the *phy_eNB_dlsch_modulation* function requires up to 26% of CPU execution time from the total of 64.99 % of PHY DL channel processing as shown in Figure 5. So it is also worth to optimize this as well.

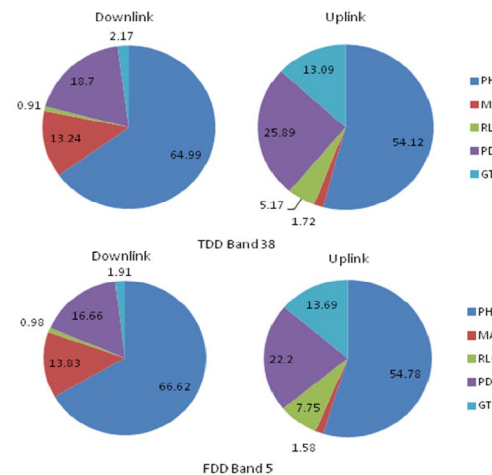


Fig. 5. Percentage of CPU Execution Time for UL/DL in Each OAI eNB's Components comprising 1 UE and 1 eNB

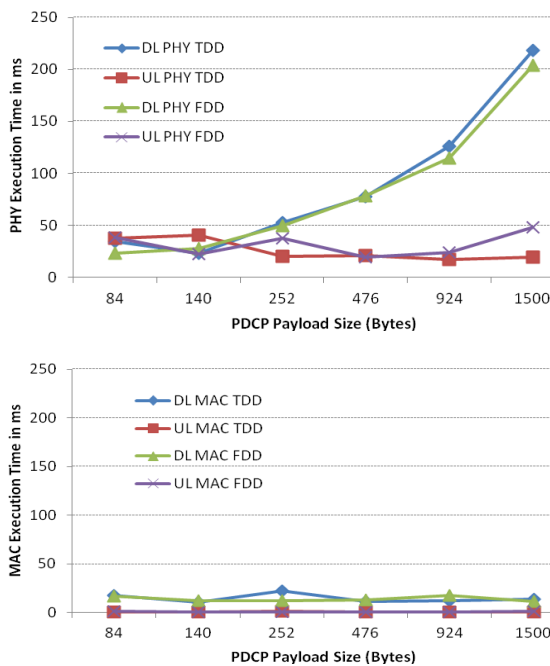


Fig. 6. PHY and MAC Execution Time for Downlink and Uplink for both FDD and TDD

Next, we investigate the DL PDCP layer processing that takes around 18% of CPU execution time as shown in Figure 5. We found out that there are 4 memory copying using *get_free_mem_block*, *memset*, *memcpy* and *free_mem_block* happened from one layer to another; first from Service SDU or GTP to PDCP, from PDCP to RLC, from RLC to MAC, and finally from MAC to PHY. We have optimized the data copying between the SDU and PDCP by allocating a shared memory region for both SDU and PDCP and simply passing the pointer from one layer to another. As shown in Figure 7, by having this simple optimization, we have able to reduce the processing time of roughly 1.5ms dealing with maximum PDCP payload amounting 8188 bytes.

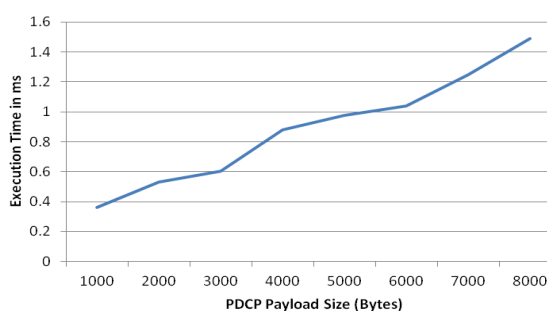


Fig. 7. Execution Time of *memcpy* based on PDCP Payload Size

V. CONCLUSIONS

We have successfully emulated the LTE FDD Band 5 and TDD Band 38 using OAI comprising 1 UE and 1 eNB using OAI. For 3GPP compliant scenario, the eNB is connected to the EPC consisting of MME, HSS, SGW and PGW. The user

plane data flow in the OAI software is clearly described and explained. Besides, we have conducted the code profiling to identify the CPU time consuming functions. It happens to lie in the DL PHY channel processing or more precisely the turbo encoder. We also proposed a simple optimization method to reduce the number of data copying in LTE protocol stack and managed to save up to 1.5ms for maximum PDCP SDU size. In future, we would like to study the PHY channel processing using Advanced Vector Extension 2 (AVX2) optimization and also to optimize the multi-thread parallel processing in LTE protocol stack.

ACKNOWLEDGMENT

We would like to thanks Raymond Knopp, Rohit Gupta and etc. from Eurecom for their support and guidance, especially on setting up the LTE FDD Band 5 emulation using OAI.

REFERENCES

- [1] S. Westwood, "The State of LTE," OpenSignal Global State of LTE Report, March 2015.
- [2] OpenBTS, September 2015. [Online] Available: <http://openbts.org>
- [3] Endaga Cellular Network, September 2015. [Online] Available: <https://www.endaga.com>
- [4] WARPLab, September 2015. [Online] Available: <http://warpproject.org/trac/wiki/WARPLab>
- [5] Amari LTE 100/Amari OTS 100 Software 4G Network on PC, September 2015. [Online] Available: <http://amarisoft.com/?p=amarilte>
- [6] Open Source 3GPP LTE Library, September 2015. [Online] Available: <https://github.com/srsLTE>
- [7] The OpenAirInterface Platform, September 2015. [Online] Available: <https://twiki.eurecom.fr/twiki/bin/view/OpenAirInterface/WebHome>
- [8] F. Kaltenberger, R. Knopp, N. Nikaein, D. Nussbaum, L. Gauthier, C. Bonnet, "OpenAirInterface: Open-source Software Radio Solution for 5G," European Conference on Networks and Communications (EUCNC), Paris, France, July 2015.
- [9] I. Latif, F. Kaltenberger, N. Nikaein, R. Knopp, "Large scale system evaluations using PHY abstraction for LTE with OpenAirInterface," 6th International ICST Conference on Simulation Tools and Techniques, Cannes, France, March 5-7, 2013.
- [10] Overview of Scheduling APIs in Linux, September 2015. [Online] Available: <http://man7.org/linux/man-pages/man7/sched.7.html>
- [11] 3GPP TS 36.211 v12.6.0, "Physical Channels and Modulation," 2014.
- [12] A. Virdis, N. Iardella, G. Stea, D. Sabella, "Performance analysis of OpenAirInterface system emulation," 3rd International Conference on Future Internet of Things and Cloud, 2015.
- [13] Tutorial on "How to Connect OAI eNB (USRP B210) with OAI UE (USRP B210) without S1 Interface", September 2015. [Online] Available: <https://twiki.eurecom.fr/twiki/bin/view/OpenAirInterface/HowToConnectOAIENBWithOAIUEWithoutS1Interface>

- [14] Tutorial on “How to Connect OAI EPC/HSS, OAI eNB (USRP B210) with OAI UE (USRP B210) using S1 interface”, Septemebr 2015. [Online] Available:
<https://twiki.eurecom.fr/twiki/bin/view/OpenAirInterface/HowToConnectOAIENBWithOAIUEWithS1Interface>
- [15] 3GPP TS 36.323 v12.4.0, “Packet Data Convergence Protocol (PDCP) Specification,” 2015.



Chun Yeow Yeoh (M'99–SM'01) born in Georgetown, Penang. He received the B.Eng. (Hons.) (First Class) degree in Electrical Electronic and M.Sc degree in Electrical Engineering from the Universiti Teknologi Malaysia, Johor, Malaysia, in 2001 and 2003. His master thesis is TCP/IP stack implementation in Embedded System.

He is currently Senior Research with TM Research & Development Sdn. Bhd., Malaysia. He has previously worked with Cozybit. Inc. in San Francisco Bay Area as staff engineer on 802.11s mesh networking and has contributed to Linux kernel. His research interests include the MAC layer of wireless communication technologies, embedded system, software-hardware co-design and Linux system.

Chun Yeow is a registered member of Board of Engineer Malaysia. He is also a member of 5G sub-WG under Malaysian Technical Standard Forum Berhad (MTSFB), a standard writing organization recognized by Malaysian law. He was treasurer of IEEE ComSoc/VT Malaysia Chapter and currently holding the position as ExComm. He has received the best paper award in IEEE ICIT, Shenzhen, China on 2007.



Mohammad Harris Mokhtar born in Kuala Lumpur, Malaysia. He received the B.E degree in Electronics Engineering from the Universiti Teknologi Malaysia, Skudai in 2002.

In 2002, he joined Mimos Berhad as Researcher. Since 2003, he joined TM Research & Development Sdn. Bhd., Malaysia. He is currently served as Associate Senior

Researcher. His research interests include Layer 2 protocols for WiFi, 3G and LTE system.



Abdul Aziz Abdul Rahman (M'09-SM'11) born in Penang, Malaysia. He received the B.Eng. degree in Electrical Electronic and M.Sc degree in Electrical Engineering from the Universiti Teknologi Malaysia, Johor, Malaysia, in 2002 and 2004, respectively and Ph.D. degree in Science and Technology from the Universiti Sains Islam Malaysia., in 2013. His Ph.D. thesis was on multi-class scheduling technique.

He is currently a Senior Researcher with TM Research & Development Sdn. Bhd., Malaysia. His research interests include physical layer algorithms for multiple-input multiple-output system, Long Term Evolution (LTE), scheduling technique and 5G system.

Ir. Dr. Abdul Rahman is a professional engineer registered with Board of Engineer Malaysia (BEM). He is also a member of Institution of Engineers, Malaysia (IEM)



Ahmad Kamsani Samingan born in Johor, Malaysia. He received the B.Eng. (Hons.) (First Class) degree and the Ph.D. degree in electronic engineering from the University of Southampton, Southampton, U.K., in 1999 and 2004, respectively. His Ph.D. thesis was on minimum bit error rate multiuser detection technique for DS-CDMA system.

He is currently an Associate Principal Researcher with TM Research & Development Sdn. Bhd., Malaysia. His research interests include physical layer algorithms for multiple-input multiple-output system, large scale antenna system, Long Term Evolution (LTE) and 5G system.

Dr. Samingan is a registered member of Board of Engineer Malaysia. He is also a member of 5G sub-WG under Malaysian Technical Standard Forum Berhad (MTSFB), a standard writing organization recognized by Malaysian law.