An Ethernet-Based Fronthaul Implementation with MAC/PHY Split LTE Processing

G. S. Birring, P. Assimakopoulos, *Member, IEEE* and N. J. Gomes, *Senior member, IEEE*
Communications Research Group, University of Kent, UK
(G.S.Birring@kent.ac.uk)

***Abstract*— A testbed implementation for an Ethernet fronthaul transporting signals arising from a long-term evolution (LTE) functional subdivision ("split") at the media-access control (MAC)/physical layer (PHY) interface is presented. Based on open LTE base station software, the testbed demonstrates significant data rate reductions compared to current fronthaul implementations that rely on In-phase and Quadrature radio sample transportation and data rates that scale with cell load. All generated traffic flows are clearly distinguishable using appropriate packet headers. A selection of test cases and their corresponding results are presented to demonstrate the operation of the fronthaul and the performance of individual flows in terms of data rates and overheads.**

***Keywords*— Functional split; LTE; 5G; Ethernet; Fronthaul, Cloud RAN**

I. Introduction

The centralized or cloud radio access network (C-RAN) will address many of the challenges faced by 4G and future 5G mobile network implementations. In a C-RAN, the traditional base station (BS) is split into a baseband processing unit (BBU), which is located centrally, and a remote radio head (RRH), located remotely. Depending on the functionalities they contain, these entities can be further generalized to a digital unit (DU) and a remote unit (RU), respectively. Currently, DU and RU communicate over a fronthaul through the transportation of digitized baseband waveforms in the form of In-phase and Quadrature (I/Q) samples, with the RU mainly performing analog and RF functions (with some limited digital signal processing) and the DU performing the bulk of the processing, including all protocol layer functionalities of LTE. This centralization of the main processing means that RUs can be of reduced processing complexity and low-power, with obvious cost benefits for operators. Current fronthaul transport is based on the Common Public Radio Interface (CPRI) [1].

As well as the benefits that a C-RAN brings in terms of management and operational cost reductions, and possibly capital expenditure reductions (depending on fiber availability) [2], the centralized architecture can facilitate functions which are now part of LTE extensions (> rel.10) and will be the norm in 5G. These include interference coordination and cooperative transmission schemes (CoMP) [3].

However, there are important challenges faced by the current C-RAN fronthaul transport when considering 4G and near-future 5G rollouts, with the most important being the data rate requirements that will be imposed [3]. Implementation costs will also be a function of vendor equipment interoperability, lack of which will have detrimental effects on the ability to achieve structural and operational convergence. In order to resolve this issue, the use of Ethernet in the fronthaul has been proposed [3-6]. Ethernet in the fronthaul has gained interest lately in the form of different standardization efforts such as the IEEE 1914 Next Generation Fronthaul Interface [7, 8] and the IEEE 802.1CM Time-Sensitive Networking for Fronthaul [9] working groups. Furthermore, the data rate limitation can be overcome by moving away from the current fully centralized approach to the implementation of different functional subdivisions (or "splits") [10-12]. This approach can bring a number of benefits in addition to significant data rate reductions, such as statistical multiplexing gains and optimized resource pooling at the DU, albeit at the cost of increasing the complexity in the RUs (this increase may not be so significant, as they currently contain processing capabilities that are underutilized). This fronthaul can further benefit operators by employing software defined networking (SDN) and network function virtualization (NFV) orchestration, both of which become more tractable by the use of Ethernet.

However, there remain a number of challenges that need to be addressed regarding both the use of Ethernet and the implementation of different functional splits, and a number of design issues need to be considered.

In this paper, we present an Ethernet fronthaul testbed that includes the full LTE protocol stack and transports LTE MAC/PHY split data flows. The LTE functionality runs in a software emulation environment based on the open source OpenAirInterface (OAI) software libraries (*see* the OpenAirInterface software alliance [13]). The testbed is flexible and can run with different options of emulated, simulated or real hardware implementations. For example, it can include the evolved packet core (EPC) functionality, hardware-based radio frequency (RF) processing and real commercial 4G phones. Alternatively, it can include emulated user equipment (UE) and simulated air interfaces. A selection of test cases and their respective results are presented to demonstrate the operation of the fronthaul. Testbeds such as that presented here will be of fundamental importance for next-generation mobile network fronthaul experimentation.

Section II begins with a high-level system description, summarizing the features of different split options. It then

This work was carried out within the framework of the European Union's Horizon 2020 research and innovation programme under grant agreement No 644526 (iCIRRUS project) and EPSRCs "Towards an Intelligent Information Infrastructure (TI3)" programme (NIRVANA project). Philippos Assimakopoulos acknowledges the funding by the NIRVANA project.

provides an overview of the implemented split in terms of processing and networking modules. Section III presents a more detailed description of the testbed set-up and the different functional entities that make up the system. A selection of experimental results from the testbed are then presented in Section IV, and the paper is concluded in Section V.

II. HIGH-LEVEL SYSTEM DESCRIPTION

A. The different split options

LTE functional subdivisions or "splits" have been considered as a means of meeting fronthaul data rate requirements for next generation mobile networks and as such have attracted the interest of standards bodies including both 3GPP [14] and IEEE groups [7]. A number of split points have been identified each with its own advantages and disadvantages; a good summary of these options and their features can be found within [12] and [14]. Factors such as data rate and latency requirements, ease of migration/deployment and ability to accommodate advanced joint signal processing techniques, play an important role in determining a suitable split point for a given use-case. A number of possible split options are shown in Fig. 1(a). In general, split points further away from the antenna and towards the mobile core, offer the highest reductions in data rates. Based on Fig. 1(a), starting from the radio side and moving towards the core, a number of interesting interface points can be identified. The first occurs at the point where the different LTE channels are demarcated (Split II) which occurs at the resource mapper (RM). At this stage, the aggregate data rate begins to depend on the cell load and thus can lead to statistical multiplexing gains. The second interesting interface occurs at the antenna-processing block (layer and port mapper) due to the transition from per-antenna flows to per-user flows (Split I). At this stage, large reductions in data rate are obtained as the data rate ceases to depend (proportionally) on the number of antennas. In general, frequency domain splits (e.g. the pre-IFFT split III) transporting frequency domain samples (instead of the sampled time waveforms) offer reductions in data rates due to reduced sample widths (bits per sample) and lack of sampling redundancy (time domain oversampling). Data rate reductions need to be considered together with other factors, such as whether a split point can accommodate advanced processing schemes (e.g. CoMP), the pooling and virtualization gains it can offer and the resulting latency constraints. Splits closer to the radio side are more capable in providing advanced features and can offer the highest virtualization/pooling gains.

The MAC/PHY split offers a good balance between these factors but at the expense of strict latency constraints. The Ethernet fronthaul will need to be adequately provisioned in order to meet such constraints. Additionally, Ethernet features such as prioritized scheduling [9], [15, 16], may offer means for guaranteeing timely delivery of packets to/from the RU.

B. Overview of the MAC/PHY split implementation

The implemented MAC/PHY split is shown in Fig. 1(a). The split interface resides between the MAC layer processing, and the error-correction block. The resulting processing module subdivision is shown in Fig. 1(b). The LTE eNodeB protocol stack, up to and including the MAC layer, runs within the digital unit (DU) and generates MAC layer protocol data units (PDUs) or transport blocks (*TBs*). The PDUs are encapsulated into Ethernet packets, are sent over the Ethernet network and are received by a remote aggregator unit (RAU) which de-packetizes the PDUs and performs all the physical layer processing (forward error correction (FEC), quadrature amplitude modulation (QAM), antenna processing, mapping of resources to resource blocks and inverse-fast Fourier transformation (IFFT)). The resulting IQ radio samples (sampled and quantized) are sent to the remote radio head (RRH) for RF processing. The RAU and RRH together then form a remote unit (RU).

The networking entity subdivision and the distribution of emulated, simulated and real-hardware entities are shown in Fig. 1(c). The EPC runs in a separate processing node that is connected through GbE (gigabit Ethernet) to the node containing the DU functionality. The DU generates the MAC PDUs which are encapsulated into Ethernet packets and sent over a physical GbE link to the RAU. The RAU then performs the PHY layer processing and sends the resulting IQ radio samples to the RRH for RF processing and transmission over the air interface. The EPC, DU, RU and UE entities are software emulations. The interface between the DU and RU is a real Ethernet hardware implementation while, for the work in this paper, RF processing and air interface are simulated.

III. TESTBED DESCRIPTION

A. The software emulation enviroment

The OpenAirInterface platform provides an open-source software-emulation environment of the complete LTE protocol stack (EPC, eNodeB, and UE) [13]. The platform consists of two main repositories of source code: 'openairinterface5G' (OAI5G) and 'openair-cn'. The OAI5G repository contains the source code used to compile the executable(s) that will run to emulate the LTE protocol stack for the radio access network (RAN) i.e. the base station (eNodeB) and/or UE.

The OAI5G source code can be compiled to produce an executable that will run to emulate one of the following: (a) A base station with radio functionality provisioned through a hardware radio transceiver (e.g. a software-defined radio); (b) A user equipment with radio functionality provisioned through a hardware radio transceiver; or (c) A number *M* of base stations and number *N* of user equipment emulated simultaneously, but with all radio signals between the emulations being simulated. Furthermore, an additional option is to compile without the EPC components and instead feed application data directly into the PDCP layer. This last optional feature of OAI can obviously run only with emulated UEs. The split implementation presented here can use any of these options but the results presented are based on option (c) without an S1 interface; that is, the radio interface is simulated with M=1 emulated base station "instances" and an arbitrary number (*N*) of emulated UE instances. The associated build configuration (that will determine which software modules are compiled) for option (c) is named 'OAISIM' (albeit modified/amended to include the MAC/PHY split modules). The 'openair-cn' repository contains the source code used to compile executables that will run to emulate the LTE protocol stack for the EPC components.

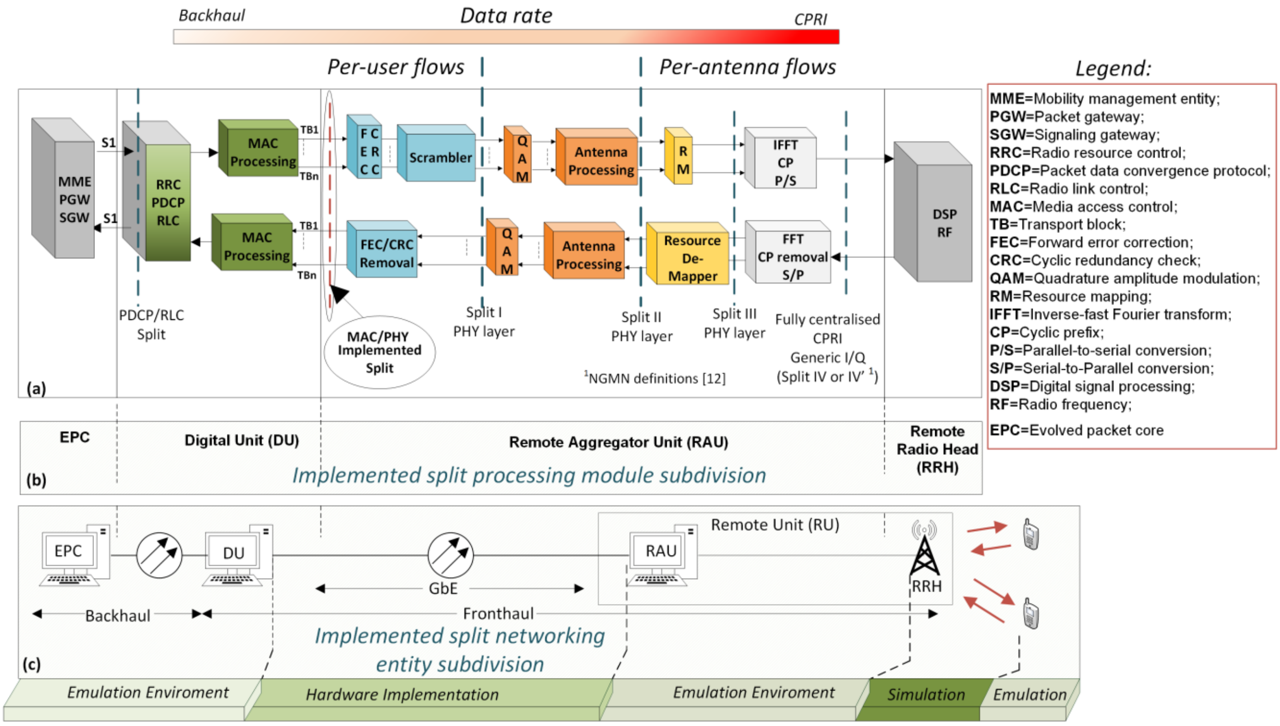

Fig. 1. (a) Different LTE functional subdivisions (functional splits) options, (b) The implemented split processing module subdivision and (c) the implemented split networking entity subdivision also showing which entities are emulated, implemented in HW and simulated.

Each EPC component is compiled as an executable that will run simultaneously alongside others. Emulated base station(s) may connect to the EPC either on the same machine (“locally”), or on another machine (“externally” via an Ethernet link).

B. The MAC/PHY split software and hardware enviroment

The hardware consists of a six-core Intel Core i7-5930K CPU based machine with 16GB RAM and two 1Gbps Ethernet NICs (dedicated to the DU and RU, respectively). The executable runs on the Ubuntu 14.04 LTS operating system with a Linux 3.19.0-59 low-latency (soft real-time) kernel.

The MAC-PHY split is implemented as an amendment to OAI5G source code. Fig. 2 illustrates the software components that constitute the new platform, and the hardware on which they operate. The purpose of the amendment is to packetize PDUs exchanged between the MAC and PHY layers of the eNodeB protocol stack, and to transmit/receive the packets efficiently and reliably over GbE (additional buffering stages are implied).

These additional functions, for both DU and RU, are made available in a new *Fronthaul Interface Library (FIL)*. The FIL encapsulates data exchanges and provides a useable abstraction (mapping-functions) to the new functionality, which is necessary for easy integration into existing OAI5G source code.

Packets are exchanged between the DU (MAC) and the RU (PHY) for each LTE subframe (or transmission time interval, *TTI*). A summary of the different packet exchanges is shown in Fig. 3. Scheduling for a subframe is triggered by the RU (through a *request* packet, namely *PKT_RU_TO_DU*). The DU responds by constructing packets encapsulating the downlink control information (DCI), through a *PKT_DCI* packet. A number of other packets are exchanged based on the information that is being scheduled for the current TTI: the random access response (RAR) through a *PKT_RAR* packet, system information (SI) through a *PKT_SI* packet, and user-plane data data for the downlink shared channel (DLSCH) through a *PKT_DLSCH* packet. Note that PKT_RU_TO_DU is used whenever there is an uplink transmission (from the RU to the DU). That is, unlike in the downlink direction, a single packet is used to aggregate all uplink transmissions. This method is used here for simplicity but separating the different uplink flows into individual packet-types is trivial and simply a matter of using the same modules that are used for packetizing in the downlink.

The PKT_DCI packet is processed at the RU before retrieval of any of the other packets is attempted. The information encapsulated within the packet notifies the RU of the type and number of allocations to expect for the current TTI.

The encapsulation format is shown in Table I. The format is common to all packets exchanged but the values assigned to specific fields are used to identify the individual traffic flows. The system has the flexibility to identify flows at varying “resolutions” by combining virtual-local area network identifiers (VLAN IDs) and packet-types. However, a simple case is to use a common VLAN ID for all flows destined to a given RU, while the individual flows within this VLAN ID are identified by packet-type.

Note that at the Ethernet level, the added headers form part of the payload. It is for the receiving Ethernet socket to identify the header boundaries. Additional fields include the subframe and system-frame number (SFN) that the received packet is intended for. These parameters, together with the packet-type field values, are used in the buffer management algorithm at the two end-points of the fronthaul network.

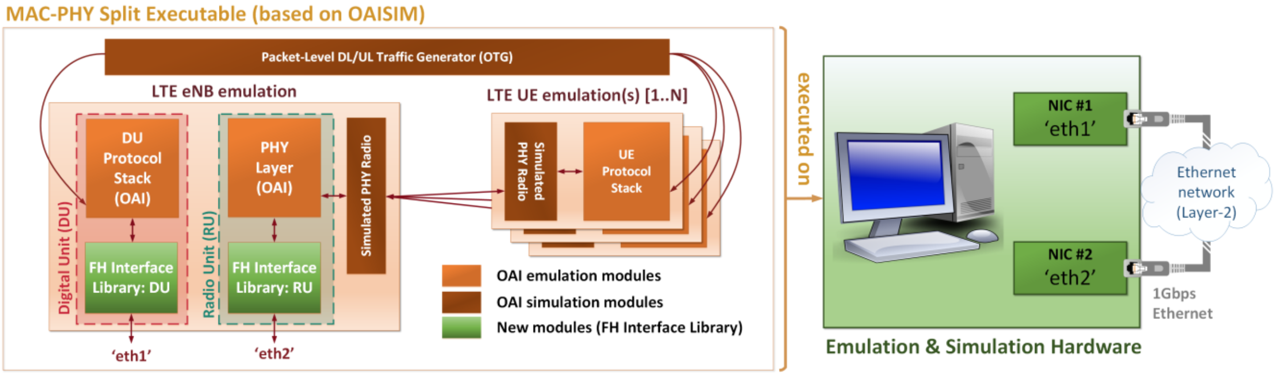

Fig. 2. Representation of the hardware and software setup used for DU and RU emulation (featuring simulated radio).

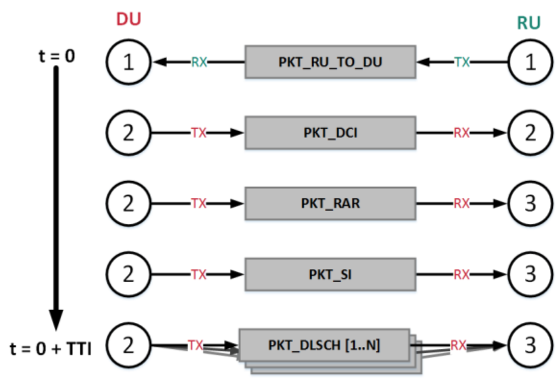

Fig. 3. Packets exchanged between the DU (MAC) and the RU (PHY) for each LTE subframe (time reference example here corresponds to the first subframe).

TABLE I. THE 28 (32)-OCTET COMMON PACKET HEADER FOR ALL PACKETS SENT/RECEIVED THROUGH THE FRONTHAUL INTERFACE. H/W: HARDWARE; SFN:SYSTEM FRAME NUMBER

Field	Size/Octets	Description
Dst MAC	6	The destination H/W address, source H/W address, and EtherType - as per IEEE 802.3. EtherType is fixed to hex '08 00' alluding to IPv4 datagram.
Src MAC	6	
VLAN ID (Optional)	4	
EtherType	2	
SFN (TX)	2	The LTE SFN and subframe the data in the packet is part of (for Tx processing).
LTE Radio Subframe (Tx)	1	
SFN (RX)	2	The LTE SFN and subframe the data in the packet is part of (for Rx processing).
LTE Radio Subframe (Rx)	1	
Packet-type	2	An unsigned 16-bit enumeration of the packet types depicted in Fig. 3.
Packet Length	2	The size of the packet in Octets, as an unsigned 16-bit integer.
Payload	*N*	Packet payload including packet-type specific data (see Table II for example, for PKT_DLSCH)
CRC	4	Cyclic redundancy check

a. Fields with italicized fonts indicate the standard Ethernet frame fields

The Ethernet payload section contains packet-type specific data (fields) in addition to the MAC PDU. An example is shown in Table II for the PKT_DLSCH packet.

The top-level software architecture of the MAC/PHY split is depicted in Fig. 4. Components (i.e. files required during compile-time and run-time) that constitute the FIL are shown, together with preexisting OAI5G components. The logging module (LOG) from OAI5G is repositioned so that it can be shared with the FIL.

TABLE II. ETHERNET FRAME PAYLOAD FIELDS FOR PKT_DLSCH

PKT_DLSCH (*Ethernet Frame Payload Section*)		
Field	**Size/Octets**	**Description**
UE index	1	Index of the UE the data in the packet is intended for
RNTI	2	UE Cell radio network temporary identifier
Length	2	Length of the payload
Payload	N-5	

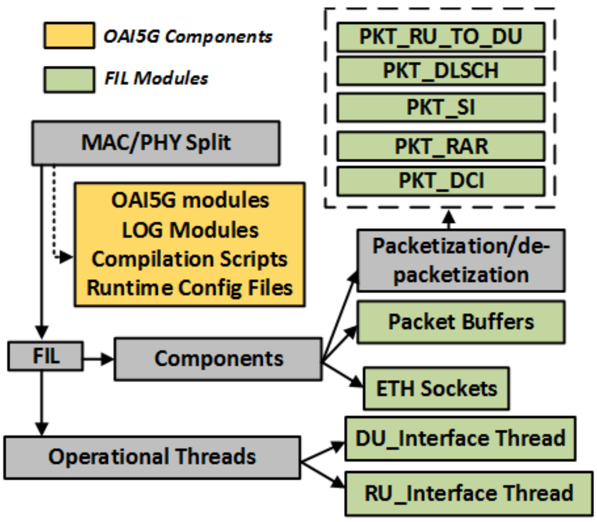

Fig. 4. Organization (modular source code architecture) of the MAC-PHY split environment.

The FIL implements and abstracts the following functions:

- Packetization/depacketization of messages exchanged between the DU and RU, including the necessary functions to append/remove data into/from the packets, and functions to send/receive packets to/from the transmit (Tx) and receive (Rx) buffers respectively.
- DU and RU raw packet sockets that exchange data with the NICs in accordance to the state of their read/write buffers.
- Tx/Rx buffers: First-in First-out (FIFO) structures designed to exchange data between the NIC and the interface thread. The Rx buffers are multidimensional structures. A number of fields from the common packet header are used by the buffer management algorithm to decide to/from which location in the buffer the packet should be pushed/pulled.

IV. Measurement Results

The results presented in this section make use of the OAI traffic generator (OTG) which has the ability to generate IP traffic and feed it directly into the PDCP layer of the eNodeB.

Fig. 5 shows a comparison of the data rates produced by the OTG (application layer) and those over the fronthaul interface, for different number of UEs. The application data rate per UE is approximately 1.2 Mbps and is for downlink flows (from the eNode to the UE). The uplink data rate is entirely due to control information (scheduling requests, HARQ acknowledgements etc.) and FIL encapsulation overheads. The 'total overhead' trace shows the increase, as a percentage, from the application data rate to the fronthaul data rate and is approximately 43% for the different number of UEs. The higher data rate over the fronthaul is a result of the encapsulation overheads added by the LTE protocol stack and the FIL. Note that the data rate over the fronthaul scales with the application data rate and number of UEs (i.e. the cell load). The equivalent data rate over the fronthaul for a 5 MHz LTE bandwidth and IQ radio transportation (CPRI-type) with a sample width of 16 bpS (bits-per-sample) would be fixed (i.e. independent of cell load) to approximately 200 Mbps.

Fig. 6 shows the results of three different tests for a single UE, with each test representing a different data rate from the traffic generator. Also shown are the percentage increases in data rate at different processing stages. The first stage is when the application data is encapsulated into the DLSCH (MAC PDU). The resulting data rate increase is 34% and is a result of the addition of LTE headers (PDCP, RLC and MAC). The second stage is when the MAC PDU has been processed by the FIL with a resulting further data rate increase of 3%. Note that both of these increases are constant for all three measurements as the TB size is fixed to approximately 1000-octets. The last stage includes all flows (i.e. all packet types) transmitted over the fronthaul. In this case, the data rate increase varies (as a percentage) from one test to the next. This variation is a result of the additional packet types (other than the DLSCH) which include SI and DCI (the RAR data rate is negligible) and the corresponding FIL encapsulation overheads for these packet types. While the amount of SI data is independent of application data rate, the amount of DCI data is not. This is shown in more detail in Fig. 7, where the different packet-types that make up the total fronthaul data rate are indicated with their equivalent contributions. The DCI contribution (in an absolute manner) increases for higher DLSCH data rates as more resource blocks are allocated to the UE and these allocations have to be indicated by the DCI.

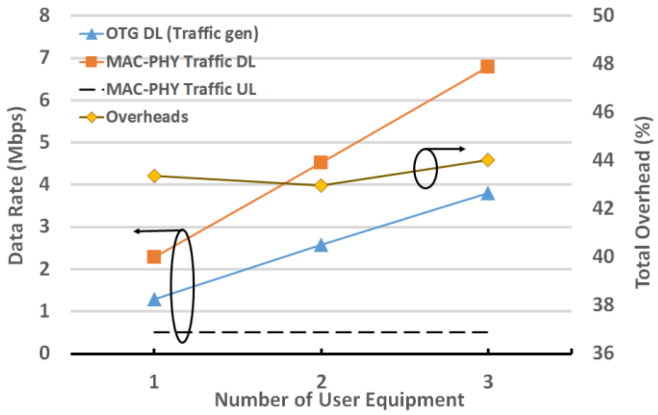

Fig. 5. Frontahul and application (OTG traffic generator) data rate measurement results for different numbers of UEs. The traffic generator is producing traffic only for the downlink direction.

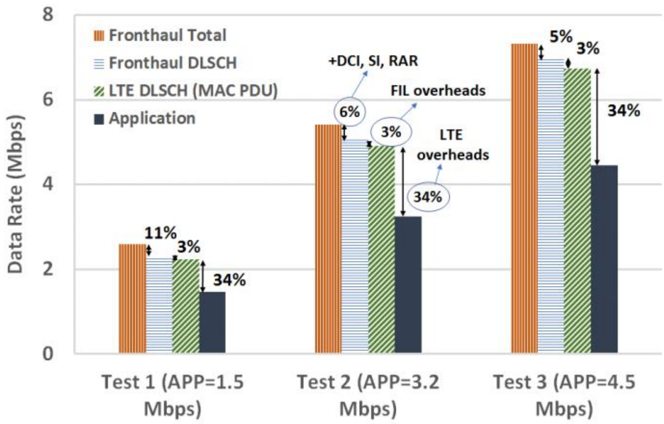

Fig. 6. Data rates and percentage increases at different points in the processing chain, for three different tests of ascending application layer data rates.

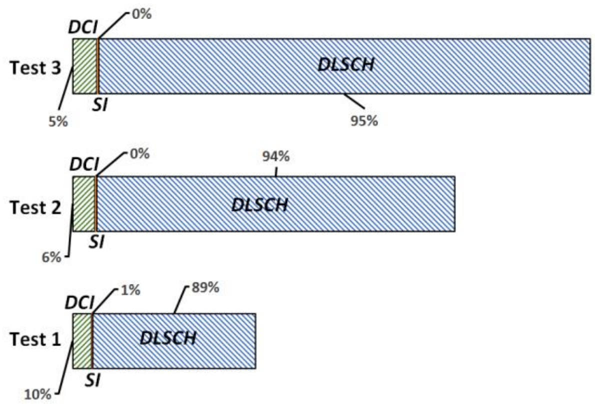

Fig. 7. Different packet-type contributions to the total fronthaul data rate. Note that the SI contribution is shown as zero due to rounding.

However, the DCI increase is much smaller that the increase in DLSCH and therefore its relative contribution to the total data rate reduces.

Fig. 8 shows the subframe processing latency of the FIL for different number of UEs. The latency is measured from the point the DCI packet for a subframe is received at the Rx buffer in the RU to the point that the last DLSCH packet for that subframe is received. The latency increases approximately linearly with number of UEs and the majority of this increase is due to the serialization delay in the Ethernet interface.

V. Conclusion

An Ethernet-based fronthaul testbed that includes the full LTE protocol stack and employs a MAC/PHY split has been presented. The LTE functionalities are software emulations based on open source software (OpenAirInterface emulation libraries). Different traffic flows are encapsulated directly into the payload section of Ethernet packets, avoiding the added overheads and processing latency that would result from using higher internetworking layers. All generated traffic flows are clearly identified by appropriate header fields. Experimental results demonstrate the operation of the fronthaul testbed in terms of data rate and the overheads added by the different processing stages, for different application data rates and number of user equipment. The different flows that are present over the fronthaul are further analyzed in terms of their contribution to the total data rate. Significant reductions in data rate are obtained compared to the centralized I/Q radio-based approach.

Ethernet in the fronthaul offers many advantages but also a number of challenges. Time-sensitive networking profiles will be fundamental for meeting latency and latency variation requirements for the latest LTE releases and envisioned 5G use cases. The testbed presented here can be employed to test these different Ethernet networking features with real functional split traffic. The testbed can potentially be used for a wide set of test cases and is by no means limited to the test cases presented in this paper.

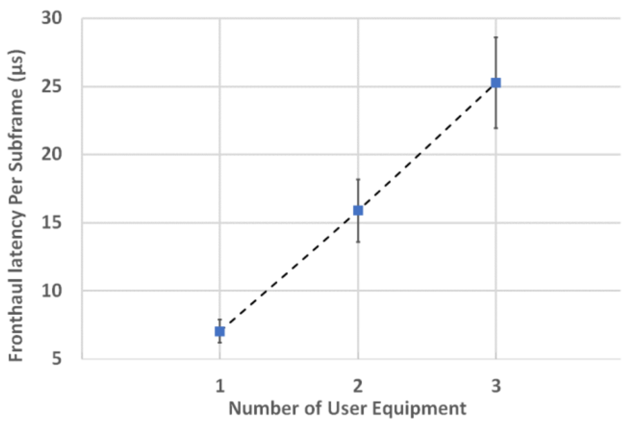

Fig. 8. Fronthaul processing latency per LTE subframe for different number of UEs.

Future improvements of the system will include separation of the DU and RU entities into different processing units, this would require decoupling of the OpenAirInterface MAC and PHY modules for standalone execution. A subsequent solution would be a computer-on-module platform featuring hardware offloading (e.g. through an FPGA).

Acknowledgment

The authors would like to thank the development teams and the members of the OpenAirInterface software alliance. Data used in this work is stored in Kent Academic Repository (https://kar.kent.ac.uk/). The authors aim to release an optimized version of the FIL source code in due course and make it available to the research community.

References

[1] CPRI (Oct. 2015), *CPRI Specification V7.0, Interface Specification* [Online]. Available: http://www.cpri.info/spec.html.

[2] China Mobile (Oct. 2011), *C-RAN: The Road Towards Green RAN (white paper)* [Online]. Available: http://labs.chinamobile.com/cran/wp-content/uploads/CRAN_white_paper6_v2_5_EN.pdf.

[3] iCIRRUS (July 2015), *D2.1: iCIRRUS intelligent C-RAN architecture* [Online]. Available: http://www.icirrus-5gnet.eu/category/deliverables/

[4] iCIRRUS (Jan. 2016), *D3.1: Verification of Ethernet as transport protocol for fronthaul / midhaul* [Online]. Available: http://www.icirrus-5gnet.eu/category/deliverables/

[5] N.J. Gomes, V. Jungnickel, P. Chanclou, J.-P. Elbers, and P. Turnbull, “A flexible, Ethernet fronthaul for 5th generation mobile and beyond (Invited),”, in *Optical fiber Commun. Conf. (OFC)*, 2016, Anaheim, CA, 2016, paper W3C.1

[6] P. Assimakopoulos, M.K. Al-Hares, and N.J. Gomes, “Switched Ethernet fronthaul architecture for cloud-radio access network”, in *OSA/IEEE J. Optical Commun. and Netw*., vol: 8, no. 12, pp. B135-B146, Dec. 2016

[7] “Next Generation Fronthaul Interface,” IEEE 1914 Working Group [Online]. Available: http://sites.ieee.org/sagroups-1914

[8] “Standard for Radio Over Ethernet Encapsulations and Mappings,” IEEE Standard P1914.3 [Online].

Available: http://sites.ieee.org/sagroups-1914/p1914-3/

[9] “Time-Sensitive Networking for Fronthaul,” IEEE Standard P802.1CM [Online]. Available: http://www.ieee802.org/1/pages/802.1cm.html

[10] U. Dötsch et al, “Quantitative analysis of split base station processing and determination of advantageous architectures for LTE,” in *Bell Labs Tech. J.*,vol. 18, no. 1, pp. 105-128, June 2013.

[11] China Mobile et al, *White Paper of NGFI, ver. 1.0 (En)* [Online]. Available: http://labs.chinamobile.com/cran/.

[12] NGMN (Mar. 2015), *A deliverable by the NGMN alliance: Further study on critical C-RAN technologies* [Online]. Available: https://www.ngmn.org/publications/technical.html

[13] “OpenAirInterface (OAI)”, OpenAirInterface Software Alliance [Online]. Available: http://www.openairinterface.org/

[14] 3GPP (Aug. 2016), *Study on New Radio Access Technology; Radio Access Architecture and Interfaces (Release 14), 3GPP TR 38.801 V0.4.0* [Online]. Available: http://www.3gpp.org/DynaReport/38-series.htm

[15] IEEE (2014, December), *802.1Q-2014-Standard for Local and metropolitan area networks—Bridges and Bridged Networks* [Online]. Available: http://standards.ieee.org/about/get/802/802.1.html

[16] “Enhancements for Scheduled Traffic," IEEE standard 802.1Qbv [Online]. Available: http://www.ieee802.org/1/pages/802.1bv.html