Cloud-based Convergence of Heterogeneous RANs in 5G Disaggregated Architectures

Nikos Makris[†⋆], Christos Zarafetas[†⋆], Pavlos Basaras[†⋆],
Thanasis Korakis[†⋆], Navid Nikaein[◇] and Leandros Tassiulas[‡]
[†]Department of Electrical and Computer Engineering, University of Thessaly, Greece
[⋆]Centre for Research and Technology Hellas, CERTH, Greece
[◇]Eurecom, Sophia Antipolis, France
[‡]Department of Electrical Engineering, Yale University, New Haven, USA
Email: {nimakris, hrzarafe, pabasara, korakis}@uth.gr, navid.nikaein@eurecom.fr, leandros.tassiulas@yale.edu

***Abstract*—Cloud-RAN based architectures are widely considered a fundamental part of 5G networks. As a consequence, in the upcoming standards for 5G RAN, disaggregating the RAN functionality between a Central Unit (CU) and multiple Distributed Units (DUs) is considered, addressing the splitting of the 5G protocol stack at the PDCP/RLC point. This split is expected to bring numerous advantages to mobile network operators, as through the isolation of the stack from the PDCP layer and upwards, the CU will be able to act as the Cloud-based convergence point among multiple heterogeneous technologies in the provisioned networks and hence able to serve multiple heterogeneous DUs. Moreover, data rate requirements for this type of split are not very demanding, thus allowing the IP-based transferring of data from the DU to CU and vice-versa. In this work, we propose, implement and evaluate a protocol for a Cloud-RAN based architecture allowing the selection and dynamic switching of different heterogeneous networks in the RAN. We rely on the open source OpenAirInterface platform and extend it to support data plane splitting of the LTE functionality, and the subsequent data injection to WiFi networks. We evaluate the platform using a real network setup, under several scenarios of network selection and different delay settings.**

***Index Terms*—CU/DU split, Midhaul, HetNets, RAN disaggregation, OpenAirInterface**

I. INTRODUCTION

Cloud-RAN is a key enabler for the 5th Generation of Mobile Networking systems (5G). The technology relies on decoupling the computational processing taking place traditionally on the base stations (BSs), and offloading part of it to Cloud instantiated VMs. The point at which the splitting of this process happens relies highly on the capacity of the fronthaul interface/technology used, between the Remote Radio Unit (RRU) and the Baseband processing Unit (BBU). Several studies have proposed different options for selecting the split point of the mobile stack (e.g. [1], [2]). Based on these, splits at the very low physical layer (e.g. I/Q samples are sent to the RRU), require constant high-rate connections between the BBU and the RRU, and hence costly fiber deployments. In such solutions, protocols such as Common Public Radio Interface (CPRI) or the newly introduced eCPRI are used for fronthauling the RRU. Nevertheless, higher layer splits can be efficiently served over traditional Ethernet/IP connections for the fronthaul interface [3]. These higher layer splits usually regard the splitting of the processing in legacy base stations either at the Layer 2 of the OSI stack, or inside the PHY layer.

Cloud-RAN technologies facilitate the creation of multi-connectivity functional architectures for 5G systems. For example, the usage of a cloud-based convergence point consisting from joint processes for either MAC or PDCP layers and upwards is introduced in [4]. This trend is also reflected in the upcoming standards for the new 5G radio interface (e.g. [5], [6]), where a CU includes the processes of the PDCP layer and upwards, able to control multiple DUs incorporating the RLC layer and downwards. The communication between the CU and the DU is taking place over the newly introduced F1 interface, utilizing the F1 Application Protocol (F1AP), even supporting DUs providing heterogeneous wireless network connections (e.g. 5G, LTE, WiFi). A single CU should be able to serve multiple DUs (one-to-many relationship), whereas each DU is served from a single CU (one-to-one relationship). The data plane traffic (payload traffic forwarded to the network UEs) is traversed over the F1-U interface, encapsulating the traffic with GPRS Tunneling Protocol (GTP) headers over UDP/IP, similar to S1-U interface, whereas the control plane (e.g. RRC signaling) is using the F1-C interface, running over SCTP/IP, as in S1-C interface. Since this decoupling of the base station functionality takes place at a higher layer, it allows for lower layer splits to be also incorporated, thus creating a multi-tier disaggregated architecture. Therefore, we refer to this interface as the midhaul interface from this point onwards.

In this paper, we build on top of our former work [7] and extend the PDCP/RLC data-plane splits with an organized protocol for managing the CU/DU communication. Using our protocol, we introduce WiFi based DUs to the network managed through the same CU instance as the rest of the network. We employ the OpenAirInterface platform [8] for developing the splits over the implementation of the LTE protocol that is provided. On top of this functionality, we explore network performance for different transport protocols and latency settings on the midhaul link. Through the utilization of the one-to-many relationship between the CU and DUs, we evaluate different policies for network selection/aggregation. We provide our experimental findings collected from a heterogeneous testbed offering all the components for our tests.

The rest of the paper is organized as follows: Section II provides a literature overview of the field. Section III presents our protocol for the intercommunication between CUs and heterogeneous DUs, and our policies for network selection. Section IV describes our experimental setup. In Section V we showcase our results and in Section VI we conclude.

II. Related Work

Cloud-RAN is a key technology for 5G mobile networks, with multiple benefits for both operators and network users, as identified in several existing works (e.g. [9], [10]). All these works focus on identifying different splits, based on a high-throughput fiber interface for the network fronthaul, and splitting the stack before/at the baseband processing level. Nevertheless, multiple splits have been further identified, particularly within the higher layers of the networking stack, for example in Layer 1 or 2, which require lower capacity for the fronthaul interface. Examples of these splits are included in [1], [2]. Packet based transferring of data over the fronthaul interface was initially introduced by China Mobile in 2015, through the Next Generation Fronthaul Interface (NGFI) [3]. In their white paper, six different splits that can be accommodated within an Ethernet based fronthaul are analyzed, along with their requirements for serving remote units with different characteristics (e.g. number of antennas, resource blocks used, etc.). Similarly, the authors in [11], measure the impact of packetization for an NGFI based fronthaul interface and validate the transferring requirements for the identified splits. In [12], they discuss the processing overhead in a Cloud-based setup for similar split architectures.

The importance of the splitting processes is also pinpointed by the current 3GPP standardization efforts for 5G. In the current efforts for the new radio interface, the PDCP/RLC split is included in [5]. Based on this study, we adopt hereafter the 3GPP terminology and refer to the cloud-based units as Central Units (CUs), consisting from PDCP layer and upwards, and the remote radio units as Distributed Units (DUs), depicting the mobile networking stack from the RLC layer and downwards (see. Fig. 1). One of the PDCP roles in the mobile networking stack is to manage and rearrange the independent RLC entities. Thus, it may be used for the subsequent management of DUs (RLC and below layers) corresponding to different technologies, enabling higher network capacity and network selection policies even on per-packet basis, as shown in [13].

Incorporation of heterogeneous technologies on the mobile networking stack has been included in the 4G protocol standards as well. Through the introduction of the Xw interface, the PDCP instance of a 4G base station shall be able to communicate with WLAN based cell deployments, towards expanding the network capacity and utilizing the unlicensed bands [14]. This process is known as LTE-WLAN Aggregation (LWA) and utilizes the LWAAP protocol for the intercommunication and signaling of the different components.

In this work, we deal with the incorporation of this type of interface in 5G Cloud-RAN deployments. In our previous work [7], we provided an experimental evaluation of the PDCP/RLC and MAC/PHY splits. Our work has been developed in OpenAirInterface [8], for supporting the (data plane) transferring of data. We concluded that the split that has the loosest requirements for the fronthaul/midhaul capacity is the PDCP/RLC. Based on our findings, and since the 3GPP standards are not yet finalized, we hereafter propose an architecture for the intercommunication of heterogeneous (LTE and WiFI) DUs with the respective CU. Based on this architecture, we propose and evaluate different policies to provide network selection for the downlink traffic (CU to DU communication) or use them for network aggregation.

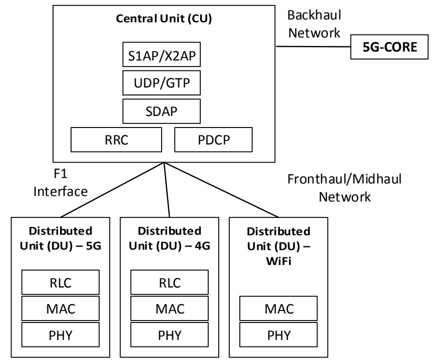


Fig. 1: 5G Data Plane architecture for the CU/DU split

III. System Architecture

In this section, we describe the system architecture and the protocol that we developed for the intercommunication between the CU and the heterogeneous DUs. We use as a reference architecture the 5G RAN architecture (see Fig. 1), and as our implementation platform the OpenAirInterface platform. Hence, our reference networking stack is 4G stack, and we analyze the processes that take place in our developed functionality for the CU, the LTE DUs and the WiFi DUs.

A. Central Unit

The CU is incorporating all the processes from PDCP layer and upwards. Thus, it provides an interface to the Core Network (e.g. EPC or 5G-Core) for transferring GTP encapsulated user data. Moreover, it is integrating the RRC procedures for signaling and controlling the operation of the different layers (PDCP, RLC, MAC). In the 5G architecture, the Service Data Adaptation Protocol (SDAP) is introduced, as a means to map traffic flows to data radio bearers. In our system design, we intercept the data plane traffic only at the Service Access Point (SAP) interface with the RLC. At that point, all the procedures of the PDCP layer have taken place, including stripping off the GTP headers, PDCP header encapsulation, numbering and data compression for the downlink (DL) traffic flow. The opposite procedures take place for the uplink (UL) traffic. Subsequently, our solution is encapsulating the PDCP PDUs with the appropriate headers to match the receiving DU(s), and forwards the traffic over the midhaul. This is the point where the network selection policies may be also running. The headers that we use are analyzed in subsection III-D.

B. LTE Distributed Unit

The DU side of the architecture deals with receiving the F1-U packets over the midhaul interface, and based on the packet header is able to invoke the respective RLC processes. For the LTE stack that we use as reference for development, our protocol intercepts the data plane specific SAPs between the layers (*pdcp_rlc_data_req* for the DL flow and *rlc_pdcp_data_ind*

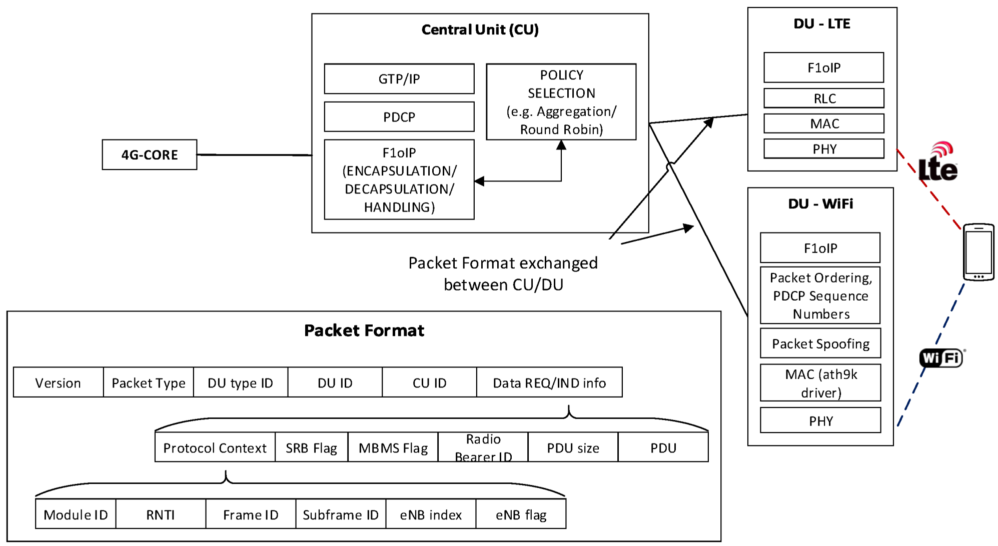


Fig. 2: CU and DU split in OAI and the transported F1-U packet format

for the UL flow) and subsequently is running the default processes provided by the stack. Therefore, if these processes need to return specific values on the further execution of the stack, like e.g. the RLC Operation Status retrieved from the data request, we handle them by packing them to new messages and sending them back to the CU.

C. WiFi Distributed Unit

As the WiFi stack significantly differs from the mobile networking stack in terms of the supported procedures, different processes need to take place upon the reception of the data request for traversing the payload to the network UE, or sending the data back to the CU. These processes include the reception of the data request transmitted from the CU, unpacking and stripping off the PDCP header, and subsequently delivering the payload to the wireless driver running on the DU device. For the UL data flow, payload traffic shall be encapsulated in the respective PDCP headers for the PDCP instance running on the CU. This includes dedicated processes for assigning new sequence numbers for the packets sent to the CU, as well as packet compression. For our implementation, and since the PDCP entity of OpenAirInterface is not supporting compression, we omit this procedure. In order to incorporate the information that does not exist in WiFi (e.g. protocol context, data bearer ID) and in order to allow the transparent handling of the packet reception at the CU side, we also introduce a lookup table at the DU side that is mapping the IP address of each receiving client to the protocol context information that we extract from incoming data requests. This process requires that the initial packet transmission happens from the CU to the DU, in order to keep this information. In the case where the end-client side uses a similar joint PDCP procedure, this process can be omitted.

D. Communication Protocol

The communication protocol that we employ is instrumenting the whole exchange process. We refer to our protocol and the procedures that take place as *F1 over IP (F1oIP)* hereafter. The CU and DU units are able to discover each other upon system startup, using a predefined capabilities and configuration file with the locations of the different modules. Upon the initial connection over the midhaul interface between the CU and the DUs, capabilities messages are exchanged with each other, stating the technology that is used by each DU. From this point, the exchange of the user-destined data taking place either on the DL or the UL channels, is being carried out through our functions in the midhaul interface. Since we need to keep both ends informed of all the values needed for carrying out any computations at each receiving end (e.g. hash tables with the network users), we piggy-back the needed information in the packets that are exchanged.

The F1oIP packet format used is depicted in Fig. 2. Each PDCP data request/indication PDU is encapsulated in a packet including fields for packet type, DU type, and addressing the DU ID and the CU ID. Different types may be supported for the same DU, as a single unit may incorporate functionality for both technologies, whereas the selection of the interface is made by the CU. Fields containing the data request or indication information (Data REQ/IND info) are used for piggy-backing the information needed for carrying out computational functions at the different network ends. This information includes the protocol context, as well as the receiving UE RNTI, and scheduling information for the transmission over the air (frame/subframe). The overall overhead posed by this header, along with the current status in the size of the respective variables that are used and exchanged for OpenAirInterface is measured to be 80 bytes long. For the case of the WiFi based DUs, this information is redundant and therefore ignored.

As an extension of the scheme we foresee the incorporation of GTP tunnels for the communication (as drafted in the standards for 5G RAN). In such a setup, each GTP TEID is mapped to the respective variables for the UE, bearer IDs, etc. and therefore the context variables can be omitted.

E. Network Selection Policies

The decoupling of the base station stack to a CU/DU functionality, and the incorporation of heterogeneous DUs in the system, creates fertile ground for the application of network selection algorithms. Based on the output of these algorithms, the CU may select the DU to which the traffic will be forwarded to, and thus select which network will be utilized. Since all the traffic is sent over the PDCP layer, this selection can be performed even on a per-packet basis. As a proof-of-concept, we developed the following policies and further evaluate them in section V:

- **LTE WiFi Aggregation**: In this mode, each data packet generated by the PDCP process is flooded to all available DUs in the system.
- **Round-Robin Scheduling**: For this policy, the CU selects to which DU to send the traffic in a Round-Robin manner.
- **Single Interface Selection**: This policy is forwarding traffic to only the selected DU.

Of course, these policies are only indicative. The system can be easily extended to host new policies for network selection, as well as gather information on the current network status and make decisions on the employed networks. This allows the implementation of traffic steering for aggregated networking topologies, e.g. selecting the transmission of time critical data over LTE/5G and using WiFi otherwise.

IV. Testbed Implementation

The described functionality has been developed in the OpenAirInterface platform and is executed over the NITOS testbed. NITOS is a heterogeneous testbed located in the premises of University of Thessaly, in Greece. It offers a very rich experimentation environment with resources spanning from commercial LTE, to WiFi and Software Defined Radio platforms that suits our experimentation needs [15].

For the development of the messaging exchange scheme, we employed Google's Protocol Buffers Library and the C language bindings [16]. By formatting the message header through the *protobuf* library, the overall header size of our communication solution, along with the piggy-backed information, is 80 bytes, that is exchanged between the CU and DU and vice-versa whenever a packet is transmitted over the network. The development of the CU/DU functionality has been written as a separate module inside the Layer 2 functionality of the OpenAirInterface code. As the transport protocol between the CU and DUs, we use an asynchronous TCP or UDP interface. The current configuration of the CU enables the utilization of different transport channels per each DU, thus allowing them to run with different settings (e.g. TCP for the LTE-DU and UDP for WiFi-DU).

The utilization of the protobuf library provides the opportunity for applications of different languages to use the same message definitions. Therefore, for the development of the WiFi DU we used a Python based agent. This agent is capable of receiving the CU messages, retrieving the payload and injecting it to the WiFi device that is configured as an Access Point. The injection is being handled by the scapy Python module [17], which provides bindings for creating packets and injecting them to a network interface.

TABLE I: Equipment parameters

Network Parameters	Values
LTE mode	FDD Band 7
LTE Frequency	2680 MHz (DL)
Antenna Mode	SISO
No RBs	50 (10 Mhz)
UE	Cat. 4 LTE, Huawei E3272
Backhaul/Midhaul RTT	$\sim$ 200 msec
Backhaul/Midhaul capacity	1Gbps Ethernet
Ethernet MTU size	1400 bytes
WiFi Clients	Atheros AR9380

The topology used for our experimentation process is given in Fig. 3. Since the current version of F1oIP is only overriding the data plane communication between the CU and the LTE DU, the production of two different binary files is not possible. However, we emulate this type of behavior by injecting delay between the network interfaces that are used for this communication between the CU and DU, equal to 0,250ms. The delay that we inject is done with the *netem* application and is equal to the mean delay that we measure over the midhaul between the CU and the WiFi DU.

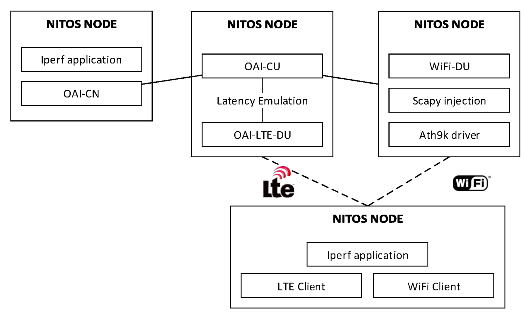


Fig. 3: Experiment mapping over the NITOS testbed

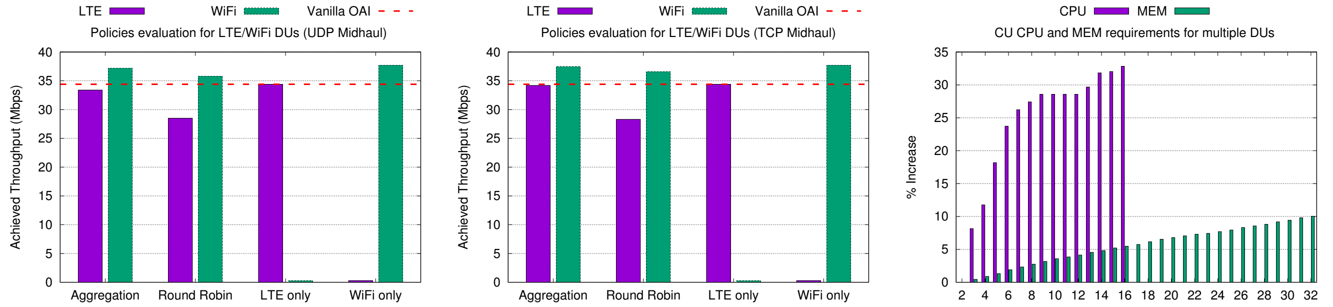
In the following section, we provide our experiment results gathered from running the platform in the testbed. Each experiment is provided with a resolution of 10 for each measurement. For generating traffic for our measurements, we use the *iperf* traffic generator, set to saturate the wireless link with UDP traffic. The LTE and WiFi DU clients are always logged to use the same Modulation and Coding Scheme over the channel, for all the experiment measurements. The configuration of all the involved testbed components is provided in Table I.

V. Experimental Results

The experimental evaluation of our scheme is organized in two subsections: 1) Initial benchmarking of the platform for the different policies for network selection, and in terms of Cloud resource consumption as the number of DUs increases, and 2) evaluation based on the delay over the midhaul.

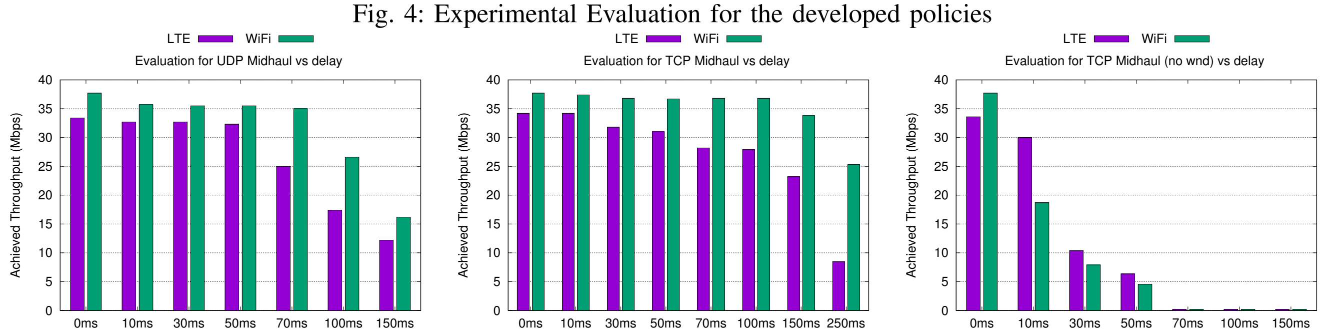
A. Policy Evaluation and Benchmarking

As a first set of experiments, we measure the performance of the network selection schemes listed in Section III-E. We measure the single-unit *vanilla* OpenAirInterface eNB to achieve 34.4Mbps goodput for the DL channel for the under-test configuration. Subsequently we measure the performance



(a) Policy Evaluation using a UDP Midhaul (b) Policy Evaluation using a TCP Midhaul (c) CPU/MEM allocation vs Number of DUs

Fig. 4: Experimental Evaluation for the developed policies



(a) Evaluation for varying delay on UDP Midhaul (b) Evaluation for varying delay on TCP Midhaul (c) Evaluation for varying delay on TCP Midhaul (no wnd)

Fig. 5: Experimental Evaluation for varying Midhaul delay

of OpenAirInterface including our additions, for either UDP or TCP based midhaul (Figures 4a and 4b respectively).

We see that for the Aggregation mode, in which the CU is forwarding traffic to all available DUs, the achieved performance for the LTE network is close to the *vanilla* setup. Likewise, the single network selection policy produces similar results. This is due to the configuration of our protocol that exchanges signaling messages between the CU and the DU only during the initial setup phase. For the Round-Robin configuration we observed slightly lower performance for both DUs, caused by the extra delay induced in the system by the respective processes that determine the DU selection.

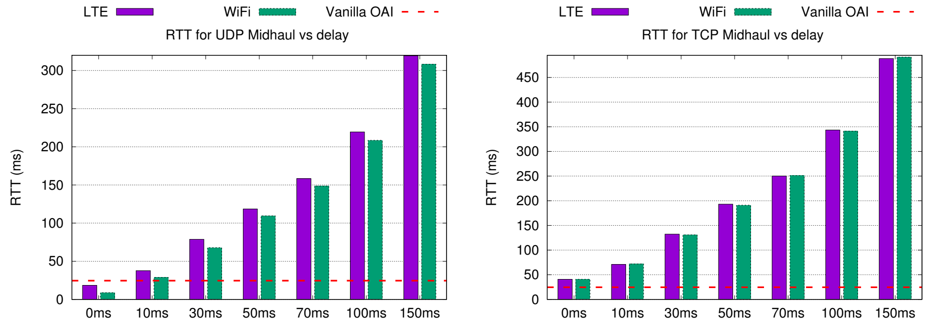
It is worth to mention here that the WiFi configuration is able to reach a maximum of 37.7 Mbps when saturating the channel. This limitation comes from the usage of the python Scapy module for injecting traffic to the WiFi interface; Scapy is opening a new socket connection for each new packet that arrives at the DU for delivering the traffic to the WiFi driver.

As a second benchmarking evaluation we measure the requirements for the CU in processing power and memory, when varying in the number of DUs deployed. Figure 4c depicts our experimental findings in terms of measured overhead for each new DU introduced to the system, compared to an initial setup with 2 DUs. We use only WiFi DUs for this type of experiment. We measure the resource requirements for up to 16 DUs in the system, as at that point we determine that the CPU of the machine running the CU software is exhausted. As illustrated, the processing resources needed to run the CU for up to 8 DUs requires approximately 25% more processing power compared to the 2 DUs scenario. For supporting the remaining set of the DUs (up to 16 DUs) we require about 34% more processing power. For the memory usage we observe a near linear increase as new DUs are added to the system. Approximately, from the CU side, each new DU consumes additionally about 30MB of memory for its efficient operation.

B. System Evaluation for varying Midhaul delay

As a second set of experiments, we measure the delivered goodput and Round Trip Time (RTT) for varying delay on the midhaul link. For the LTE case, we use the *socat* application to redirect the requests from the CU to an intermediary testbed node before delivering them to the DU. The latency on both the LTE and WiFi links is measured to be the same.

We use the *netem* application to set delay on the midhaul link. We use the aggregation policy for these experiments, as this is the policy that produced higher results in the initial benchmarking experiments in the previous section. Figures 5a and 5b show the results for either UDP based midhaul or TCP. For both cases, we see that the performance starts to drop at around 70ms of midhaul latency. Nevertheless, the respective RTT (see Fig. 6) for the same interfaces seems to be growing by the double delay and a fixed amount added by the wireless access. Based on our results, we can incur that if the midhaul interface is realized over a fiber based Ethernet link, the CU will be able to serve distributed DUs located at 500 Kms away without any decrease in the provisioned service at the end-client. Of course, in such environments, we need to further investigate on how to differentiate the paths that low latency applications take in order to minimize the impact on the user's



(a) RTT Evaluation using a UDP Midhaul

(b) RTT Evaluation using a TCP Midhaul

Fig. 6: RTT Evaluation results for different Midhaul interfaces

QoE. UDP from that point and for higher delays, starts to perform worse than TCP, which through the adaptation of the congestion window and the receiving window is able to better handle the higher latency on the midhaul link. However, if these features are deactivated (see Figure 5c), we see that TCP cannot handle even lower delays in the midhaul (e.g. 30ms). For the WiFi case, we see that it is more resilient to delay, starting to drop for delays higher than 100ms. This is caused by the fact that the bottleneck of our implementation is not the midhaul interface, but the injection module at the WiFi DU.

VI. Conclusion

In this work, we provided an experimental evaluation of a protocol enabling the Cloud-based convergence of heterogeneous networks, when operating with the CU/DU split. We detailed the communication protocol and the development process for the split operation of the OpenAirInterface networking stack. We provided proof-of-concept and performance experiments on the network selection policies that we use for the DUs, the indicative cost in processing power and memory allocation at the CU side, and the overall delivered goodput and RTT for varying delay over the midhaul.

Our results show that the proposed CU/DU split is not posing any performance limitations compared to the legacy eNB setup, but only adds-up to the overall flexibility of the provisioned network. All our developments are publicly available through the OpenAirInterface repository under the *feature-127-protocol-split* branch. In the future we foresee the incorporation of the RRC messaging as well in our communication scheme and the tailoring of the protocol according to the developments made to the 5G standards (e.g. incorporation of the F1AP protocol for the midhaul interface). We also plan the extension of the scheme to include network status messages, based on the network utilization (e.g. WiFi performance degradation due to external interference), and the subsequent management of the involved DUs from the CU point. Finally, we expect the incorporation of pricing schemes, located at the CU for the DU selection, according to selections made by the network's end-users.

Acknowledgement

This work has been funded by European Union under Grant Agreement 762057 (H2020 5G-PICTURE).

References

[1] NTT DOCOMO. "R3-161687, Draft TR 38.801 (v030) Study on New Radio Access Technology: Radio Access Architecture and Interfaces", NTT DOCOMO, INC (Rapporteur), 3GPP TSG RAN3, August 2016.
[2] Small Cell Forum. "Document 159.07.02, Small cell virtualization functional splits and use cases, Small Cell Forum", January 2016.
[3] China Mobile Research Institute. "White Paper of Next Generation Fronthaul Interface", 2015.
[4] A. Ravanshid, P. Rost, D. S. Michalopoulos, V. V. Phan, H. Bakker, D. Aziz, S. Tayade, H. D. Schotten, S. Wong, and O. Holland. Multi-connectivity functional architectures in 5g. In *IEEE International Conference on Communications Workshops (ICC)*, May 2016.
[5] 3GPP. "3GPP TS 38.401 V0.2.0 (2017-07), 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; NG-RAN; Architecture description (Release 15)", 2017.
[6] 3GPP. "3GPP TS 38.470 V0.3.0 (2017-09), 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; NG-RAN; F1 general aspects and principles (Release 15)", 2017.
[7] N. Makris, P. Basaras, T. Korakis, N. Nikaein, and L. Tassiulas. "Experimental Evaluation of Functional Splits for 5G Cloud-RANs". In *IEEE International Conference on Communications (ICC)*, 2017.
[8] N. Nikaein, M. K. Marina, S. Manickam, A. Dawson, R. Knopp, and C. Bonnet. "OpenAirInterface: A Flexible Platform for 5G Research". *SIGCOMM Comput. Commun. Rev.*, 44(5), 2014.
[9] D. Wubben, P. Rost, J. S. Bartelt, M. Lalam, V. Savin, M. Gorgoglione, A. Dekorsy, and G. Fettweis. "Benefits and impact of cloud computing on 5G signal processing: Flexible centralization through cloud-RAN". *IEEE signal processing magazine*, 31(6), 2014.
[10] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann. "Cloud RAN for mobile networks - a technology overview". *IEEE Communications surveys & tutorials*, 17(1), 2015.
[11] C. Y. Chang, R. Schiavi, N. Nikaein, T. Spyropoulos, and C. Bonnet. "Impact of packetization and functional split on C-RAN fronthaul performance". In *2016 IEEE International Conference on Communications (ICC)*.
[12] N. Nikaein. "Processing radio access network functions in the cloud: Critical issues and modeling". In *Proceedings of the 6th International Workshop on Mobile Cloud Computing and Services*. ACM, 2015.
[13] Y. Khadraoui, X. Lagrange, and A. Gravey. "Performance analysis of LTE-WiFi very tight coupling". In *13th IEEE Annual Consumer Communications Networking Conference (CCNC)*, 2016.
[14] 3GPP. "3GPP TS 36.360 V14.0.0 (2017-03), 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); LTE-WLAN Aggregation Adaptation Protocol (LWAAP) specification (Release 14)", 2017.
[15] N. Makris, C. Zarafetas, S. Kechagias, T. Korakis, I. Seskar, and L. Tassiulas. "Enabling open access to LTE network components; the NITOS testbed paradigm". In *1st IEEE Conference on Network Softwarization (NetSoft)*, April 2015.
[16] Protobuf-C. "protobuf-c–C bindings for Googles Protocol Buffers–Google Project Hosting".
[17] P. Biondi. "Scapy: explore the net with new eyes". Technical report, Technical report, EADS Corporate Research Center, http://www. secdev. org, 2005.