An adaptive functional split in 5G networks

Alberto Martínez Alba
Chair of Communication Networks
Technical University of Munich
Munich, Germany
alberto.martinez-alba@tum.de

Jorge Humberto Gómez Velásquez
Chair of Communication Networks
Technical University of Munich
Munich, Germany
jorge.gomez@tum.de

Wolfgang Kellerer
Chair of Communication Networks
Technical University of Munich
Munich, Germany
wolfgang.kellerer@tum.de

***Abstract*—The 5G radio access network (RAN) features a partially centralized architecture, in which a subset of the network functions are deployed in a centralized unit. Centralizing these functions reduces operating costs and enables coordination techniques. However, the more functions are centralized, the more capacity is needed on the fronthaul network connecting centralized and distributed units. In addition, the required fronthaul capacity also depends on the instantaneous user traffic, which varies over time. Therefore, in order to optimize its performance, the 5G RAN should be able to dynamically adapt its centralization level to the user traffic. In this paper, we present the design of an adaptive RAN that can switch between two different centralization options at runtime. We provide design objectives and challenges, as well as measurement results from a working implementation.**

***Index Terms*—5G, functional split, flexible, adaptive.**

I. Introduction

The performance objectives envisioned for 5G networks pose important challenges on all design aspects of next-generation mobile networks. In order to meet the expected user demands, three ambitious use cases are considered: enhanced mobile broadband (eMBB), ultra-reliable low latency communications (URLLC), and massive machine-type communications (mMTC) [1]. For the eMBB use case, downlink and uplink data rates are expected to be ten times higher than those offered by 4G networks. Similarly, in URLLC, the latency will be ten times lower, and in mMTC the amount of supported devices will be in the order of hundreds of thousands. Such challenging objectives necessarily entail a highly efficient utilization of the radio and computational resources available to the network. This motivates a radical change in the architecture of the next-generation radio access network (RAN) with respect to that of 4G.

In 4G, the RAN consists of a single element: the base station or eNodeB, which is deployed close to the antennas. This element individually handles the radio resources and processes all the traffic coming from or to the user equipment (UE). In contrast, for 5G it has been proposed to centralize all the processing of all base stations into a data center [2]. This has two main advantages. On the one hand, pooling all computational resources reduces the cost of 5G deployments by leveraging resource multiplexing gain. On the other hand, centralization enables easier coordination between next-generation eNodeBs (gNBs), which helps to reduce inter-cell interference. In turn, this allows for denser cells and hence higher data rates.

Nevertheless, a centralized RAN (C-RAN) has also a major drawback: it requires a low-latency, high-capacity *fronthaul* network connecting the data center and the remote locations. This implies high costs for many network operators, which cannot reuse their existing less-performing networks. To address this issue, the 3GPP opted for a partially centralized architecture, in which only a subset of the functions of the 5G processing chain are located in a centralized unit (CU) [3]. The remaining functions are deployed in a distributed unit (DU), which is collocated with the antennas. This *functional split* reduces the burden on the fronthaul network [4], but poses a new challenge: finding the optimal subset of functions to centralize for every network.

Given the advantages of C-RAN, a mobile network operator should centralize as many functions as possible. However, the more functions are centralized, the more capacity is needed on the fronthaul for the same user traffic. This is due to the presence of headers, control signals, error correction codes, etc., which are increasingly added to the user data as the split point gets lower. This is illustrated in Fig. 1, where four possible functional splits are depicted. As a consequence, the functional split that a mobile network can afford for each gNB depends on the characteristics of the user traffic and the limitations of the fronthaul network. This can be formulated as an optimization problem, which has been already tackled by previous work [5]. Nevertheless, the current literature on this problem usually focuses on selecting the functional split for the deployment phase. That is, only a priori statistics of the expected user traffic, such as the peak or the average data rate, are used to find the optimal functional split for each gNB in the RAN. Such an approach effectively ignores the variability of the user traffic, which is the main component of the fronthaul traffic for most functional splits.

If the functional split is fixed, the fronthaul may be underutilized when the user traffic is low, or congested when the user traffic is high. In the former case, a more-centralized functional split would be less costly and would allow for improved function coordination. Conversely, in the latter case, a less-centralized functional split would alleviate congestion on the fronthaul. Hence, an adaptive RAN that can change the functional split depending on the user traffic would utilize the resources more efficiently and would provide better service to the users. This is illustrated in Fig. 2, where the fronthaul traffic resulting from two different functional splits (PDCP-

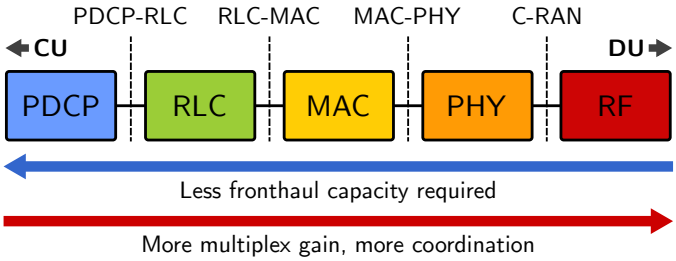

Fig. 1. Possible functional splits for the 5G RAN, where each function is a layer of the protocol stack (incomplete list). The functions to the left of each split are located in the CU, whereas those to the right are in the DU.

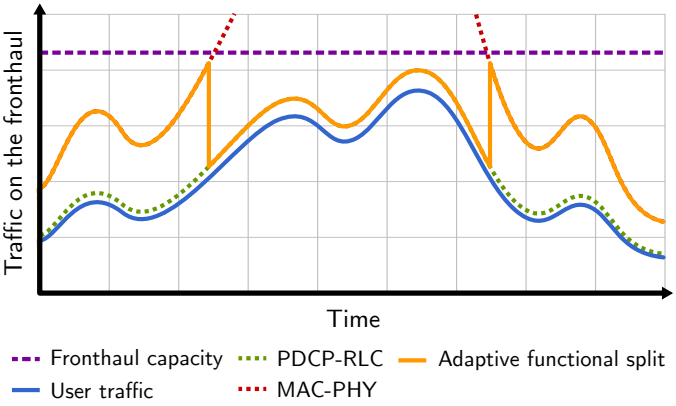

Fig. 2. Example of the fronthaul traffic produced by a 5G RAN implementing a PDCP-RLC split, a MAC-PHY split, and an adaptive functional split as a function of the user traffic.

RLC and MAC-PHY) and an adaptive RAN are compared. We can see how a static MAC-PHY split would not be possible, as it will exceed the maximum capacity of the fronthaul link. Conversely, a PDCP-RLC is feasible, but it would underutilize network resources. This may lead, for instance, to unmanaged interference between gNBs, which could be prevented with a more centralized architecture. With an adaptive RAN, the functional split can be changed to implement the most centralized architecture that is possible at any time.

Although the idea of an adaptive RAN seems promising, its actual design and implementation are not trivial. Indeed, the 5G RAN is a rather complicated system and the functional split is a recent concept. Nonetheless, the foreseen benefits motivate the research towards an adaptive functional split. In this paper, we present a flexible platform that can switch between two functional splits (MAC-PHY and PDCP-RLC) at runtime. To the best of our knowledge, this is the first work that addresses in detail the task of implementing an adaptive functional split. In summary, our contributions are mainly two. On the one hand, we present the objectives, challenges, and existing options to switch functional splits at runtime. On the other hand, we present a working implementation of an adaptive RAN that can switch between PDCP-RLC and MAC-PHY splits, from which we provide actual measurements.

The rest of this paper is organized as follows. Section II summarizes the current state of the art on the topic. Section III introduces the considered functional splits. Section IV presents the objectives and challenges of implementing an adaptive functional split. Section V shows the details of our implemented adaptive RAN. Finally, Section VI concludes the paper.

II. Related work

The introduction of functional splits in the architecture of the 5G RAN has triggered the research on their optimal definition and selection. We can find abundant work on the characteristics of diverse functional splits, such as [4], [6], [7], or [8]. Owing to their differences, the authors conclude that the employed functional split has to be adapted to the conditions that the network faces. To this end, [9] presents a high-level overview of a flexible 5G RAN, which includes the ability to support multiple functional splits, in order to adapt to the expected user traffic. Building upon this, in [10] the authors propose a RAN architecture that simultaneously supports different functional splits for each DU in the network. However, none of these works consider changing the functional split on-the-fly, but rely on a priori statistics of the network to select the optimal, static functional split.

Conversely, there are works which do tackle, to a greater or lesser extent, the change of functional splits at runtime. For instance, in FlexCRAN [11] the authors propose a framework for a partially centralized RAN that supports on-the-fly changes of the functional split, although they do not elaborate on this feature. In [12], the problem of dynamically selecting the functional split for each cell is addressed. The authors present an algorithm that allows virtual mobile operators to change the functional splits every time a new virtual network is added. Similarly, [13] proposes an architecture for a 5G RAN implementing an algorithm to dynamically select the functional split of a cell. This work places special emphasis on enabling changes at runtime. Nonetheless, it only covers the optimal selection of the functional split, regardless of the feasibility of the required changes. Finally, in [14] the authors present a pioneer platform that can switch between functional splits at runtime. However, the options are limited to low-layer, intra-physical splits, and the details about performing the switching are not explained.

In the light of these previous works, we see that there is interest in implementing a network that can adapt its functional split at runtime. Nevertheless, there is still little work coping with the challenge of realizing such an adaptation. In this paper, we contribute to overcome this challenge.

III. Functional splits

The 3GPP envisions eight different options for the functional split [15]. These correspond to the interfaces of the five layers of the protocol stack (RRC-PDCP, PDCP-RLC, RLC-MAC, MAC-PHY, and PHY-RF), plus three internal splits within the RLC, MAC, and PHY layers. However, not all options are valid for actual implementations. For instance, the RLC-MAC split is deemed almost useless, as it is more complex than PDCP-RLC but does not bring any additional benefit [15].

In this work, we focus on an adaptive 5G RAN that can switch between PDCP-RLC and MAC-PHY splits. The selection of these two splits is based on their advantages and their relative simplicity. In the following paragraphs, we briefly summarize the characteristics of each split.

A. PDCP-RLC

In this split, the PDCP function is centralized in the CU, whereas the RLC, MAC, PHY, and RF functions are located in the DU. This has three main advantages. First, it reduces the operating cost with respect to a distributed architecture, since the PDCP function is in charge of ciphering, which may be a compute-intensive task. Second, the fronthaul traffic is very similar to the user traffic, as only a small PDCP header is added to each IP packet. Thus, the fronthaul traffic is also comparable to the backhaul traffic in LTE, which enables reutilization of backhaul networks. That is, operators could reuse their former backhaul infrastructure as fronthaul network. Finally, the standardization effort required to implement this split is low, given that it has already been considered for LTE Dual Connectivity [15].

Owing to its advantages, PDCP-RLC is the split currently considered by the 3GPP Release 15 for the New Radio (NR) specifications [16]. Nonetheless, the PDCP-RLC lacks enough centralized functions to perform any kind of advanced coordination technique with other gNBs, and the DU is still required to implement the also compute-intensive MAC and PHY layers. Therefore, a more centralized split would be a better option if the fronthaul capacity allows.

B. MAC-PHY

In this split, the PDCP, RLC, and MAC functions are centralized in the CU, whereas the PHY, and RF functions are located in the DU. More centralized functions means less operating costs with respect to PDCP-RLC. Furthermore, coordination techniques such as coordinated scheduling or coordinated link adaptation are possible with a centralized MAC [17], while the fronthaul traffic is still considerably lower than those of Intra-PHY or C-RAN splits.

Due to its characteristics, the MAC-PHY split has been selected by the Small Cell Forum for their envisioned network, and standardized in the nFAPI initiative [18]. Nevertheless, the MAC-PHY split has also disadvantages. On the one hand, the presence of additional headers and control signals between MAC and PHY layers increase the fronthaul capacity and latency requirements with respect to PDCP-RLC. On the other hand, its coordination capabilities are limited with respect to C-RAN or Intra-PHY splits.

IV. Adaptive functional split

In this section, we provide an overview of the objectives and challenges that an adaptive 5G RAN faces. Furthermore, we introduce the details of our solution, implementing an adaptive functional split between PDCP-RLC and MAC-PHY.

A. Objectives and challenges

Given that a centralized architecture outperforms a distributed one in terms of cost and coordination capabilities, the objective of an adaptive 5G RAN is to make sure that each gNB in the network operates at the most centralized functional split that is supported by the fronthaul network. Since the load of the fronthaul network depends on the user traffic, which may change over time, the functional split should also be able to change at runtime. The difference between functional splits is the location of the functions of the 5G processing chain, as explained in Sec. III. As a result, switching between functional splits at runtime is equivalent to live migrate functions from the CU to the DU, or vice versa.

There are at least two main obstacles when live migrating RAN functions: increased fronthaul traffic and function downtime. The former refers to the additional information that needs to be exchanged between CU and DU during the migration, which leads to an increase in the fronthaul traffic. The latter is the time during which the functions being migrated are not available, owing to a possible need of halting these functions in order to complete the migration. These two obstacles lead to two secondary objectives. On the one hand, the overhead traffic during the migration should be minimized, as the sheer motivation of the migration may be the reduction of the traffic on the fronthaul. That is, if the user traffic increases and the centralization level has to be reduced to decrease the fronthaul load, a migration that produces high overhead would be counterproductive. On the other hand, the downtime of the migration should be minimized as well, since it may negatively impact the experience of the user. For instance, in our case, a downtime between PDCP and RLC layers translates into a delayed transmission of PDCP packet data units (PDUs), which may be a problem for low latency users. Moreover, an interruption of the communication between MAC and PHY layers cannot be higher than the scheduling interval, or else entire slots will be wasted.

B. Migration strategy

As mentioned above, any change in the functional split implies moving functions from the CU to the DU, or vice versa. In our case, the difference between MAC-PHY and PDCP-RLC is the location of the MAC and RLC functions, as depicted in Fig. 1. In the MAC-PHY split, these functions are located in the CU, whereas in PDCP-RLC split they are located in the DU. Hence, when switching from PDCP-RLC to MAC-PHY, we need to move the MAC and RLC functions from the DU to the CU, and vice versa. In order to do this, we need an underlying migration strategy that fits the characteristics of the functions and the requirements of the network.

The live migration of functions is a well-studied topic in the field of network function virtualization (NFV). Indeed, a common strategy to live migrate a function is to deploy it on a virtual machine and then migrate the virtual machine [19]. A virtualization platform is hence needed to manage the migration, since the hardware of CU and DU may be different. For instance, platforms like OpenStack, based on hypervisors

such as KVM or Xen, allow for such live migrations [20]. However, this *virtualization-based* approach conflicts directly with the two limitations presented in the previous section. To the best of our knowledge, no existing virtualization platform can offer a downtime comparable to the scheduling interval of a 5G network (1 ms or less) [21]. Moreover, live migrating a virtual machine entails copying its disk and memory to the destination, thus producing a high traffic overhead until the migration is completed.

A faster, lighter type of live migration is therefore needed to change functional splits in a 5G RAN. We propose a *replication-based* approach, in which the MAC and RLC functions are simultaneously deployed in both DU and CU. That is, at every instant there is one active set of MAC and RLC functions at either the DU or the CU, and an inactive one at the other unit. When the migration is performed, the roles are exchanged: the active functions are disabled, and the inactive functions, enabled. This approach can be considerably faster and produces much less overhead during the migration than the virtualization-based approach. Its main drawback is that it requires to have MAC and RLC processes running simultaneously on both units, even when they are not used. This creates additional memory and CPU consumption on the inactive unit, which should be taken into account as operating expenses. Furthermore, in order to provide uninterrupted service to the users, the MAC and RLC functions cannot be just turned on or off. There is a set of variables and data structures that are created, modified, and used by both functions at runtime. These have to be carefully transferred to the destination before completing the migration.

C. State transfer

In a replication-based migration, there are three basic tasks to accomplish: (i) transfer the *state* from the old set of functions to the new one, (ii) deactivate the old set of functions, and (iii) activate the new set of functions. That is, before being able to toggle the two sets of functions on and off, we must guarantee that they are in the same state. In this work, we define the state of the MAC and RLC functions as the set of parameters that are required for the correct operation of these functions and are susceptible to change at runtime.

We can classify these parameters into three types, according to their origin and updating frequency. First, we have the *external parameters*, which are used by the MAC and RLC layers but not created or modified by them. These are mostly details of the connected UEs and defined radio bearers provided by either the UEs or by higher layers of the gNB. Some examples of external parameters are logical channel IDs (LCIDs), Radio Network Temporary Identifiers (RNTIs), timer values, and channel quality indicators (CQIs). These parameters do not change frequently, as they are only created or modified when a UE connects or disconnects, or when the higher layers decide so. Second, the *internal parameters* are those created or modified by the MAC and RLC layers themselves, such as the RLC sequence number, frame and subframe numbers, and the list of active HARQ processes. In general, these parameters change every slot or subframe. Finally, we consider the *content of the buffers* as the last type of state parameters, including the RLC transmission and retransmission buffers, and the HARQ retransmission buffer. The content of these buffers varies every slot or when new data is received.

Given their different characteristics, the three types of parameters can be treated differently in order to minimize the overhead and downtime of the migration. For instance, the external parameters can be forwarded to both active and inactive RLC and MAC functions every time they are updated, e.g., when a new UE connects or disconnects. This results in a small overhead traffic following these events, but since it is performed ahead of time, it reduces the amount of data exchanged during the migration. Conversely, the internal parameters, owing to their fast updating frequency, have to be transmitted during the migration, as only the active MAC and RLC layers know the current values. The amount of overhead traffic caused by transferring these internal parameters during a migration is, however, almost negligible, as only a few bytes per UE are needed to store them. Namely, up to two bytes for the RLC sequence number, two bytes for frame and subframe numbers, and one byte for the HARQ processes.

In contrast, transferring the content of the buffers is a more challenging task. According to the 3GPP specifications for 5G [22], the RLC retransmission buffer should store from 50 to 160 RLC PDUs in order to account for the maximum expected acknowledgment delay. This implies that, if the content of this buffer is transferred in one scheduling interval (to avoid downtime), the overhead traffic produced during the migration would be between 50 and 160 times higher than the downlink data rate of the air interface. This is actually comparable to the capacity needed for full centralization. Hence, an adaptive functional split would be pointless with such a high overhead.

In order to reduce the overhead, a solution would be to put both active and inactive buffers into a common pre-defined state just before the migration, instead of transferring the content. As the packets stored in the buffer cannot be modified, this can only mean to empty the buffers before completing the migration. There are two options to achieve this: either to drop the content of the old buffers, or to redirect new packet arrivals to the new buffers while the old buffers drain normally. The former, or *hard migration*, provides the fastest migration with no overhead traffic, although it implies packet losses. The latter, or *soft migration*, prevents packet losses, but it may introduce delay while the old buffers are emptied, and produces an overhead traffic equal to the arrival rate of new user packets. In addition, a combination of both migration options, or *custom migration*, can be also defined. For this option, the old buffers are given a fixed amount of time to empty, after which all remaining content is dropped. Therefore, a custom migration provides an adjustable trade-off between latency and packet losses.

D. Proposed migration procedure

In this section, we describe our migration procedure between PDCP-RLC and MAC-PHY splits. Based on what

is explained in the previous section, we treat the synchronization of external and internal parameters differently. We synchronize the external parameters before the migration by duplicating configuration messages at two points. First, when a UE performs a successful random access procedure, the data structures defining the established radio bearers are copied to the inactive MAC and RLC functions. Second, after the core has successfully registered or updated a UE, the RRC function forwards its configuration to both active and inactive MAC and RLC functions before generating the messages *RRC Connection Setup* or *Reconfiguration*.

The internal parameters and the content of the buffers are synchronized by means of a five-step migration procedure. For the sake of brevity, we focus on a migration from MAC-PHY to PDCP-RLC with only downlink flows. However, extending this procedure to the other direction or to the uplink is straightforward. The procedure is as follows:

1) **CU request handling:** The CU receives the command for a soft, hard, or custom migration through its northbound interface. In the case of custom migration, it also receives the maximum time allocated for the draining of RLC and HARQ buffers.
2) **CU traffic steering:** A migration controller function redirects the flow of arriving PDCP PDUs to the DU. As a consequence, the RLC buffer at the CU stops receiving new arrivals and the RLC buffer at the DU starts receiving them.
3) **Buffer synchronization:** This step may consist of up to three stages, depending on the type of migration.
   3.1) **RLC draining:** Only for soft and custom migrations. The MAC scheduler continues its normal operation until the RLC buffers are empty, or until the allocated time runs out (in a custom migration).
   3.2) **MAC draining:** Only for soft and custom migrations. The MAC layer at the CU waits until the acknowledgment of the last active HARQ process is received, or until the allocated time runs out (in a custom migration). In a soft migration, this step guarantees that no packets are lost.
   3.3) **MAC and RLC reset:** Only for hard migrations, and custom migrations after the allocated time. The content of the RLC and HARQ buffers is dropped and the list of active HARQ processes is reset.
4) **MAC and RLC synchronization**: The current frame, subframe, and RLC sequence numbers are sent from the CU to the DU, and the RLC and MAC functions at the DU are updated accordingly.
5) **DU traffic steering:** The migration controller activates the MAC and RLC functions at the DU, so they can start processing the PDUs stored in the RLC buffers.

This procedure produces no downtime, as it guarantees that there are always active MAC and RLC functions during the migration. This is achieved by letting the MAC and RLC functions at the DU store the new arriving PDUs (step 2), while the functions at the CU are still processing those in the buffer when the migration starts. However, the absence of downtime does not imply that the migration does not cause additional latency to the users. Indeed, step 3.2 (MAC draining) is basically a waiting step, which may delay the processing of PDUs arriving at that time. This delay is prevented by hard migrations or limited by custom migrations, at the cost of possible packet losses (step 3.3).

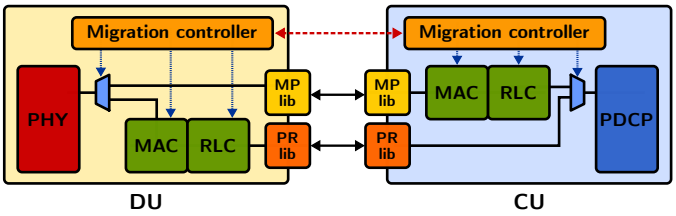

Fig. 3. Functional architecture of the two binaries supporting an adaptive functional split. The blue trapezoid symbolizes the ability of the migration controller to steer the flow of data.

V. Implementation Results

In this section, we present the details of our flexible platform implementing an adaptive 5G RAN, as well as measurement results. This platform is able to switch from PDCP-RLC to MAC-PHY, or vice versa, at runtime without interrupting ongoing user traffic. In order to do this, it follows the aforementioned procedure for replication-based migrations of the three types: hard, soft, and custom.

A. Platform description

The hardware equipment consists of four off-the-shelf Intel i7 PCs for the UE, DU, CU, and core network. The fronthaul link connecting DU and CU is a 1 Gb/s Ethernet link. The same link is used for the backhaul, which connects the CU to the core network. For the radio interface between the UE and the DU, two programmable USRP B200 are used to transmit a single-carrier 10 MHz LTE signal. Although only one UE is used in this setup, the results are applicable to any multi-UE which produces the same joint downlink rate.

The software platform is a modification of srsLTE [23]. Although srsLTE does not provide functional splits out of the box, its code is clearly organized into layers, thus facilitating the implementation of functional splits. From the original, monolithic srsLTE binary that contains all the RAN layers, two different 5G binaries are created: one containing the PHY, MAC, and RLC functions for the DU, and the other hosting the MAC, RLC, PDCP, and RRC functions for the CU (see Fig. 3). The code implementing the MAC and RLC functions can be disabled at runtime in both binaries, thus providing the software basis for switching functional splits.

In order to make these two binaries work with each other, interfaces for the PDCP-RLC and MAC-PHY splits need to be defined. These interfaces are implemented as libraries whose purpose is to convert the original function calls between layers into Ethernet packets. This is done by means of Google Protocol Buffers, which is a tool to transform C++ objects into serialized data. The two libraries are known as the *PR*

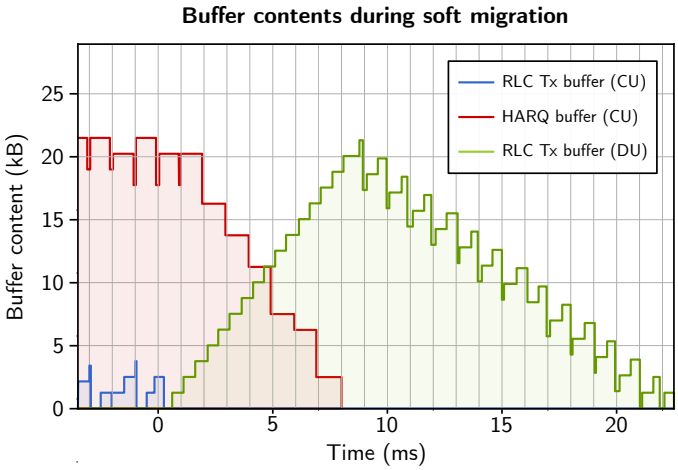

Fig. 4. Content of the RLC and HARQ buffers during a soft migration. The input traffic is a constant-rate 20 Mb/s stream, and the network downlink capacity is 31.7 Mb/s. The migration starts at $t = 0$ ms.

library for the PDCP-RLC split, and the *MP library* for the MAC-PHY split. The PR library performs a pure serialization of the objects exchanged in the srsLTE function calls, but the MP library follows the general structure of nFAPI [18]. Both libraries are logically connected in pairs through the fronthaul network. This is also shown in Fig. 3.

Apart from the RAN layers and their interfaces, two *migration controllers* are needed to orchestrate the migration in both units. These functions can activate or deactivate the MAC and RLC functions of their unit, and steer the user traffic accordingly. Furthermore, they have East-West interfaces to communicate with each other, in order to orchestrate the migration, and a northbound interface to receive the migration command from upper layers. In future work, the intelligence required to automatically trigger the migration will be added to the migration controller at the CU.

B. Measurement results

In order to show the performance of our implemented 5G RAN, we present here some interesting experimental results. In Fig. 4, we can see the evolution of the content of the RLC and HARQ buffers when a soft migration from MAC-PHY to PDCP-RLC is performed, that is, when the MAC and RLC functions are migrated from the CU to the DU. For this experiment, a constant-rate 20 Mb/s downlink stream is used as input. The service rate is 31.7 Mb/s, corresponding to the downlink data rate supported by the air interface of a UE experiencing the highest channel quality in a 10 MHz carrier. The data shown in the figure is extracted from logs, which report the status of the buffers every time they are updated (every millisecond or less). We can see that the RLC draining step lasts a small fraction of a millisecond, after which the RLC buffer at the CU is already empty. Then, the HARQ buffer takes around 8 ms to drain. During that time, the RLC buffer at the DU receives PDCP PDUs from the CU, but it is not yet ready to pass them on to the MAC layer. Shortly after the HARQ buffer at the CU is empty, the migration is

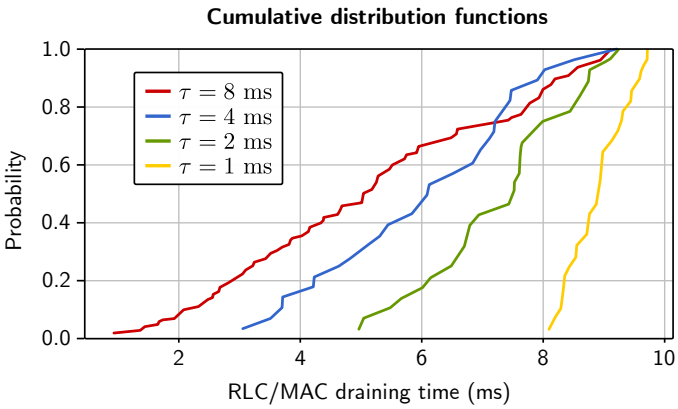

Fig. 5. Cumulative distribution functions of the RLC/MAC draining times for different inter-arrival times τ. Each curve corresponds to 100 migrations.

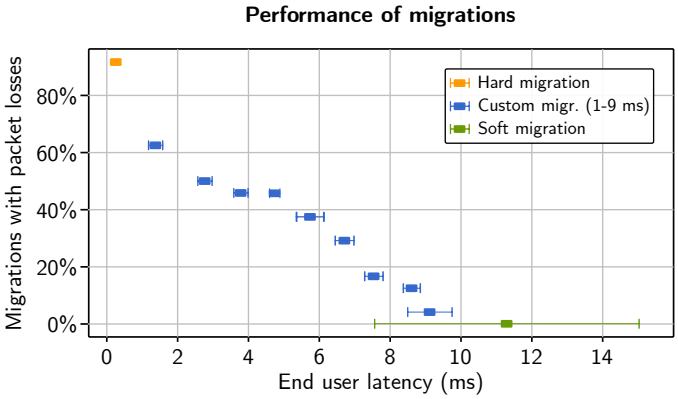

Fig. 6. Number of migrations experiencing one or more packet losses as a function of the maximum end-user latency. Nine custom migrations are shown, with allocated times ranging from 1 to 9 ms. Each point represents 20 migrations from MAC-PHY to PDCP-RLC.

finished and the RLC buffer at the DU starts to be emptied by the MAC layer. In this measurement, we can clearly see how the handover between old and new MAC and RLC functions is performed.

As mentioned in Sec. IV-D, the time it takes for the RLC and HARQ buffers to drain is important, as it impacts the end-user latency. Therefore, a specific experiment is performed to find out the distribution of this RLC/MAC draining time as a function of the inter-arrival time τ of the incoming PDCP PDUs. The results, shown in Fig. 5, allow us to conclude that the lower the inter-arrival time, the higher the average draining time. This makes sense, as the lower the inter-arrival time, the more HARQ processes will be active. Besides, we see that the maximum draining time is around 10 ms, which corresponds to 1 ms to empty the RLC buffer and 8–9 ms to empty the HARQ buffer (containing up to 8 MAC PDUs).

Finally, in Fig. 6 we show the results of an experiment that measures the trade-off between packet loses and end-user latency. Eleven types of migration are considered: hard migration, soft migration, and customs migrations with allocated time ranging from 1 to 9 ms. The RAN is dealing with a traffic of periodic packets transmitted every 0.3 ms, and interference

conditions producing a 10% packet error rate (1 out of 10 MAC PDUs has to be retransmitted). For each migration, the maximum additional end-user latency is recorded, as well as the presence of packet losses. After 20 migrations of each type, the ratio of migrations with one or more packet losses is plotted against the additional end-user latency. We observe that hard migrations barely add any delay, but are prone to suffer from packet losses. Conversely, soft migrations can guarantee that there are no packet loses, at the cost of introducing a delay of up to 15 ms. The values for packet losses and end-user latency of custom migrations lay in the middle, showing a roughly linear relationship between latency and packet losses.

VI. Conclusion

The 3GPP proposes a partially centralized architecture for the 5G RAN. However, a static implementation of this architecture is inefficient, owing to the variable behavior of user traffic. An adaptive RAN that can modify its functional split to match the traffic conditions would perform better, as it would be able to benefit from the highest possible level of centralization all the time. In this paper, we present an adaptive 5G RAN implementation that supports migrations between functional splits at runtime. We lay out the objectives, challenges, and technical options to realize it. Then, we describe our proposal to switch from MAC-PHY to PDCP-RLC without service interruption. Finally, we present measurements extracted from our implementation, which reveal that switching functional splits at runtime is indeed possible, at the cost of introducing packet losses or additional delay. Future work will investigate the conditions under which such an adaptation of the functional split should be done.

Acknowledgment

This work is part of a project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No 647158 - FlexNets). The authors alone are responsible for the content of the paper.

References

[1] N. Alliance, "5g white paper," *Next generation mobile networks, white paper*, pp. 1–125, 2015.

[2] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, "Cloud ran for mobile networksa technology overview," *IEEE Communications surveys & tutorials*, vol. 17, no. 1, pp. 405–426, 2015.

[3] 3GPP, "NG-RAN; Architecture description," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.401, 09 2018, version 15.3.0. [Online]. Available: http://www.3gpp.org/DynaReport/38401.htm

[4] U. Dötsch, M. Doll, H.-P. Mayer, F. Schaich, J. Segel, and P. Sehier, "Quantitative analysis of split base station processing and determination of advantageous architectures for lte," *Bell Labs Technical Journal*, vol. 18, no. 1, pp. 105–128, 2013.

[5] A. Garcia-Saavedra, X. Costa-Perez, D. J. Leith, and G. Iosifidis, "Fluidran: Optimized vran/mec orchestration," in *Proc. IEEE Infocom*, 2018.

[6] L. M. Larsen, A. Checko, and H. L. Christiansen, "A survey of the functional splits proposed for 5g mobile crosshaul networks," *IEEE Communications Surveys & Tutorials*, 2018.

[7] J. Bartelt, P. Rost, D. Wubben, J. Lessmann, B. Melis, and G. Fettweis, "Fronthaul and backhaul requirements of flexibly centralized radio access networks," *IEEE Wireless Communications*, vol. 22, no. 5, pp. 105–111, October 2015.

[8] L. Valcarenghi, K. Kondepu, F. Giannone, and P. Castoldi, "Requirements for 5g fronthaul," in *2016 18th International Conference on Transparent Optical Networks (ICTON)*, July 2016, pp. 1–5.

[9] A. Maeder, A. Ali, A. Bedekar, A. F. Cattoni, D. Chandramouli, S. Chandrashekar, L. Du, M. Hesse, C. Sartori, and S. Turtinen, "A scalable and flexible radio access network architecture for fifth generation mobile networks," *IEEE Communications Magazine*, vol. 54, no. 11, pp. 16–23, 2016.

[10] O. Chabbouh, S. B. Rejeb, N. Agoulmine, and Z. Choukair, "Cloud ran architecture model based upon flexible ran functionalities split for 5g networks," in *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, March 2017, pp. 184–188.

[11] C. Chang, N. Nikaein, R. Knopp, T. Spyropoulos, and S. S. Kumar, "Flexcran: A flexible functional split framework over ethernet fronthaul in cloud-ran," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–7.

[12] D. Harutyunyan and R. Riggio, "Flex5g: Flexible functional split in 5g networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 961–975, Sep. 2018.

[13] A. Marotta, D. Cassioli, K. Kondepu, C. Antonelli, and L. Valcarenghi, "Efficient management of flexible functional split through software defined 5g converged access," in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–6.

[14] Y. Alfadhli, M. Xu, S. Liu, F. Lu, P. Peng, and G. Chang, "Real-time demonstration of adaptive functional split in 5g flexible mobile fronthaul networks," in *2018 Optical Fiber Communications Conference and Exposition (OFC)*, March 2018, pp. 1–3.

[15] 3GPP, "Study on new radio access technology: Radio access architecture and interfaces," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 38.801, 03 2017, version 14.0.0.

[16] ——, "NG-RAN; F1 Application Protocol (F1AP)," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.473, 09 2018, version 15.3.0. [Online]. Available: http://www.3gpp.org/-DynaReport/38473.htm

[17] A. Martínez Alba, A. Basta, J. H. Gómez Velásquez, and W. Kellerer, "A realistic coordinated scheduling scheme for the next-generation RAN," in *2018 IEEE Global Communications Conference: Next-Generation Networking and Internet (Globecom2018 NGNI)*, Dec. 2018.

[18] Small Cell Forum, "FAPI and nFAPI specifications," Document 082.09.05, 05 2017, release 9.0.

[19] W. Cerroni and F. Callegati, "Live migration of virtual network functions in cloud-based edge networks," in *2014 IEEE International Conference on Communications (ICC)*, June 2014, pp. 2963–2968.

[20] Openstack. [Online]. Available: https://www.openstack.org/

[21] C. H. Benet, K. A. Noghani, and A. J. Kassler, "Minimizing live vm migration downtime using openflow based resiliency mechanisms," in *2016 5th IEEE International Conference on Cloud Networking (Cloudnet)*, Oct 2016, pp. 27–32.

[22] 3GPP, "NR; User Equipment (UE) radio access capabilities," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 38.306, 10 2018, version 15.3.0.

[23] I. Gomez-Miguelez, A. Garcia-Saavedra, P. D. Sutton, P. Serrano, C. Cano, and D. J. Leith, "srslte: an open-source platform for lte evolution and experimentation," in *Proceedings of the Tenth ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation, and Characterization*. ACM, 2016, pp. 25–32.