Integrating Fronthaul and Backhaul Networks: Transport Challenges and Feasibility Results

Sergio Gonzalez-Diaz, Andres Garcia-Saavedra, Antonio de la Oliva, Xavier Costa-Perez, Robert Gazda, Alain Mourad, Thomas Deiss, Josep Mangues-Bafalluy, Paola Iovanna, Stefano Stracca, Phillip Leithead

Abstract—In addition to CPRI, new functional splits have been defined in 5G creating diverse fronthaul transport bandwidth and latency requirements. These fronthaul requirements shall be fulfilled simultaneously together with the backhaul requirements by an integrated fronthaul and backhaul transport solution. In this paper, we analyze the technical challenges to achieve an integrated transport solution in 5G and propose specific solutions to address these challenges. These solutions have been implemented and verified with pre-commercial equipment. Our results confirm that an integrated fronthaul and backhaul transport dubbed *Crosshaul* can meet all the requirements of 5G fronthaul and backhaul in a cost-efficient manner.

Index Terms—5G-Crosshaul, Fronthaul, Backhaul, 5G, C-RAN

1 Introduction and Motivation

Future mobile networks (5G and beyond) are expected to deliver data rates an order of magnitude higher, fostering the evolution and deployment of new services that were not possible before. In order to increase the available data rates per area unit substantially, 5G network deployments will require a much higher capillarity, effectively increasing the density of the network. In this way, a more effective spectrum reuse will be enabled and thus, much higher data rates achieved than in current mobile systems.

A key architectural component to support 5G's increased user data rates is the *Transport Network*, which is responsible for interconnecting and feeding data to/from the Radio Access Network (RAN). The *traditional* approach employs fully-fledged radio access points[1], where all the radio protocol stack, including the baseband processing functionality, is distributed and co-located with the radio frontend, and connected to the mobile core through a *backhaul* (BH) transport network. Early *backhaul* segment technologies consisted in circuit switching nodes and evolved into a packet-based network, typically consisting in packet-switching nodes interconnected by *wires*, i.e., optical fiber and/or copper, and in whole or in part, wireless links using high capacity microwave bands. We will refer to this approach as distributed RAN or D-RAN and is illustrated in Fig. 1 (top figure).

Motivated by the success of mobile technologies, RAN designers have been seeking innovative ways of pushing the limits of RAN performance and reducing costs. As a result, Centralized/Cloud RAN (C-RAN) has emerged as the technology of choice to realize both performance and cost-effectiveness. C-RAN decomposes radio access point functionality into a small footprint, basic radio part (remote radio heads, RRHs), which includes the lower levels of the protocol stack (amplification, digital-to-analog converters, etc.) and are located at distributed radio sites (distributed units, DUs), and a pool-able base band processing part (base band unit, BBU) that centralizes the upper levels of the stack at central/cloud units (CUs). C-RAN will be widely used in 5G since it helps reduce the costs associated with the RAN and provides additional performance gains due to the ability to pool resources and the coordinated processing of signals from several cells [1]. The seminal paper [2] proposes the use of C-RAN as a mechanism to reduce the CAPEX

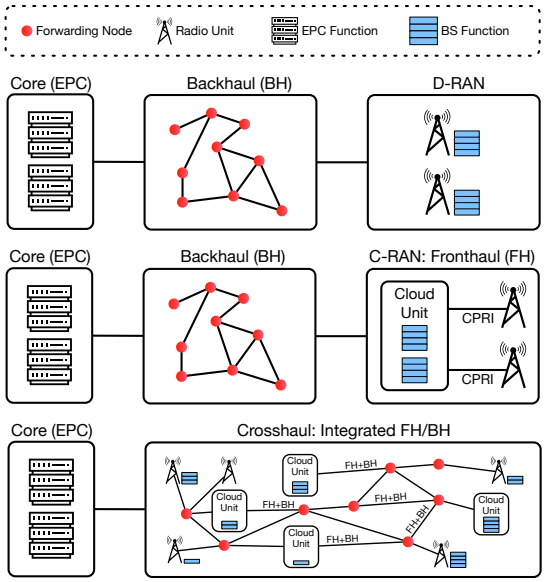


Figure 1: D-RAN, C-RAN and Crosshaul concepts.

- *S. Gonzalez-Diaz and A. de la Oliva are with Universidad Carlos III de Madrid.*
- *A. Garcia-Saavedra, and X. Costa-Perez are with NEC Laboratories Europe.*
- *R. Gazda, A. Mourad and P. Leithead are with InterDigital.*
- *T. Deiss is with Nokia.*
- *J. Mangues-Bafalluy is with CTTC.*
- *P. Iovanna and S. Stracca are with Ericsson Research.*

1. Base Transceiver Station (BTS or BS) in 2G, NodeB in 3G, enhanced nodeB (eNB) in 4G or next Generation NodeB (gNB) in 5G.

and OPEX of networks, benefiting from reduced expenses on the site antenna. C-RAN has been also proven useful to improve the performance of the air interface due to an easier synchronization across radio access points, allowing the use of Coordinated Multipoint (CoMP) techniques. Traditional 4G C-RAN, where the CU/DU functional split is deep in the physical layer, requires both a very high bandwidth and a very low latency network connection between DUs (hosting RRHs) and CUs (hosting BBUs). This network segment has traditionally been known as *fronthaul* (FH) and is illustrated in the middle plot of Fig. 1.

The deployment of the C-RAN concept is already a commercial reality in 4G. The main problem with current 4G C-RAN approaches is the technology used for the fronthaul interface, Common Public Radio Interface (CPRI), which connects the distributed units and the central units. This legacy fronthaul protocol requires a point-to-point dedicated fiber link between the BBUs (at the CU) and RRHs (at the DUs) due to the CPRI's stringent throughput and latency requirements [3]. This increases the cost of management and operation of the network, since the operator now has to face the operation of two different networks, one based on packets (the normal backhaul or transport network) and a second one using a completely different technology. This fact has triggered a change in how standardization bodies has focused on C-RAN for 5G and beyond, working on solutions based on packets that can use standard switching technologies [4] and the feasibility of multiple, alternative functional splits between CU and DUs [5], [6] that relax such tight transport constraints while retaining the cost-effectiveness and radio performance gains of pure C-RAN as much as possible.

In this way, a re-design of the fronthaul is taking place in multiple fora, with the common goal of defining a next generation fronthaul interface (NGFI) which is based on a packet-switched transport protocol that enables statistical multiplexing, infrastructure reuse and higher degrees of freedom for routing. Recently published deployment guidelines for 5G [7], consider a two-tier architecture, where the antennas are connected to a Distributed Unit through eCPRI or CPRI (reusing the base-band infrastructure from 4G for NSA), while connecting the DUs to the CUs through the so called F1 interface [8]. This interface assumes an RLC/PDCP split (higher layer functional split) as the one we present in this work. This blurs the traditional separation between *backhaul* and *fronthaul* transport networks driving their convergence towards a common packet-based transport solution which we refer to as *Crosshaul* [9]–[12] as illustrated at the bottom of Fig 1.

In this work, we present the first design and experimental evaluation of an integrated fronthaul and backhaul (Crosshaul) network. It has been built as a proof-of concept by integrating solutions from several relevant transport technology manufacturers and vendors specifically designed for the purpose of this research work. The aim of this proof-of concept is to analyze the feasibility of using a common, multi-domain transport substrate to forward fronthaul and backhaul traffic, including the novel 5G functional splits (see §2) while coexisting with legacy CPRI and backhaul traffic. Specifically, our main contributions are:

Table 1: Summary of technologies used in our Crosshaul proof of concept

Technology	Description	Type
LTE small cell	eNB 0	Commercial
PDCP/RLC DU	DU for eNB 1	Prototype
PDCP/RLC CU	CU for eNB 1	Prototype
MAC/PHY DU	DU for eNB 2	Prototype
MAC/PHY CU	CU for eNB 2	Prototype
PHY/RF DU	DU for eNB 3	Commercial
PHY/RF CU	CU for eNB 3	Commercial
Fast-Forward	XFE mmWave PHY	Prototype
EdgeLink™	XFE mmWave PHY	Prototype
WDM fiber	XFE fiber PHY	Commercial
Crosshaul Packet Forwarding Element (XPFE)	XFE SDN Ethernet Link Layer	Prototype
Crosshaul Circuit Switching Element (XCSE) remote node	XFE TDM Link Layer	Prototype
Crosshaul Circuit Switching Element (XCSE) hub node	XFE TDM Link Layer	Prototype
Evolved Packet Core	4G EPC	Commercial
LTE transceivers	User Equipments	Commercial

- We have designed a multi-layered switching node (XFE) that is capable of transporting Crosshaul traffic, i.e., traffic from heterogeneous functional splits. To this aim XFE integrates different switching technologies (packet and circuit-based) and different physical technologies (copper, fiber and radio);
- We have designed mmWave-based radios to transport flows from low-level functional splits. To the best of our knowledge, this is the first paper that uses mmWave technologies to transport low-level functional splits;
- We have designed a novel TDM framing method to accommodate the use of standard WDM fiber technology to transport flows from heterogeneous functional splits;
- We have designed a novel SDN-based packet-based switching node that is capable of rendering low-latency forwarding using off-the-shelf hardware components;
- In addition to performing an in-depth experimental validation of individual features/technologies of XFE, we have built a comprehensive testbed with all the technologies to validate the ability of XFE to transport flows from different functional splits *within the same switching infrastructure*. To the best of our knowledge, this is the first empirical validation that this is possible in practice.

Table 1 summarizes the technologies comprising our *Crosshaul proof-of-concept*, including the prototypes devised within the context of this work to make the deployment of a Crosshaul network feasible in practice.

The remainder of this paper is structured as follows. §2 introduces the most important concepts and definitions used in our Crosshaul design. §3 presents a theoretical analysis while §4 details the solutions used to integrate fronthaul and backhaul in a common transport segment. The testbed and scenario used to analyze the integrated transport network is presented in §5, with the obtained results shown in §6. §7 details the lessons learned from the experimentation, §8 revises related work and, finally, §9 draws our conclusions.

Table 2: Functional splits analysis in a LTE scenario: 1 user/TTI, 20 MHz bandwidth; Downlink: MCS (modulation and coding scheme) index 28, 2x2 MIMO, 100 Resource Blocks (RBs), 2 transport blocks of 75376 bits/subframe; Uplink: MCS 23, 1x2 SIMO, 96 RBs, 1 transport block of 48936 bits/subframe. Replicated from [10].

Split #	LTE BS Functional decomposition	DL/UL BW req. (Mbps)	Delay req. (μs)	Gains
1	RRC/PDCP	151/48	$30 \cdot 10^3$	Enables L3 functionality for multiple small cells to use the same HW Enhanced mobility across nodes w/o inter-small cell data forwarding/signaling Reduced mobility-related signaling to the mobile core segment No X2 endpoints between small cells and macro eNBs Control plane and user plane separation
1	PDCP/RLC	151/48	$30 \cdot 10^3$	Enables L3 and some L2 functionality to use the same HW.
2	RLC/MAC	151/48	$6 \cdot 10^3$	Resource sharing benefits for both storage and processor utilization.
3	High/Low MAC	151/49	$6 \cdot 10^3$	Synchronized coordination and control of multiple cells Coordination across cells enables CA, CoMP, eICIC or cross carrier scheduling
4	MAC/PHY	152/49	250	Enhancements to CoMP with RU frame alignment and centralized HARQ.
5	PHY Split I	173/452	250	More opportunities to disable parts of the CU at quiet times to save power
6	PHY Split II	933/903	250	Central L1 CU can be scaled based on average utilisation across all cells
7	PHY Split III	1966/1966	250	Smaller CU results in less processing resource and power saving
8	PHY Split IV	2457.6/2457.6	250	Enhancements to joint reception CoMP with uplink PHY level combining

2 Fronthaul & Backhaul Integration: Crosshaul

The concept of C-RAN has already been deployed in 4G networks, providing cost reductions and performance improvements [1]. The 4G C-RAN implementation is the lowest of the possible functional split in the LTE protocol stack. This split partitions the RF layer at the DU, known as remote radio head (RRH), which samples the air interface and sends this information to the CU implementing the base band processing, also known as the base band unit (BBU). The protocol used for the transmission of the RF samples from the air interface is known as the Common Public Radio Interface (CPRI) [13], [14]. CPRI utilizes a serial transmission, instead of packetized, typically requiring a point-to-point, high bandwidth and low delay dedicated fiber link between the RRH and BBU.

In addition, CPRI suffers of two key problems, jeopardizing its possible application to 5G and beyond. The first problem is known as bandwidth explosion. With CPRI, there is a direct relation between the bandwidth needed to transport the air interface samples with the radio channel bandwidth and number of antennas. A single 20MHz LTE channel providing 75 Mbps to the end user (with no MIMO) requires of 1228.8 Mbps transport capacity. This value is independent of the actual user traffic in the channel, i.e., an empty channel will also require of 1228.8 Mbps of transport, and linearly dependent with the number of antennas used for MIMO (2x2 MIMO requires 2457.6 Mbps for the same scenario), which renders this model unfeasible for massive MIMO deployments expected in 5G. The second issue is the delay required to transport CPRI data. Current CPRI specifications requires 100 μs one-way delay between the RRH and BBU[2], which prevents any possible processing within the transport or limits the number of transport hops between the RRH and BBU.

In short, the *conventional* Cloud RAN model based on CPRI to support the centralization of baseband processing is not fully suitable for 5G for the following reasons:

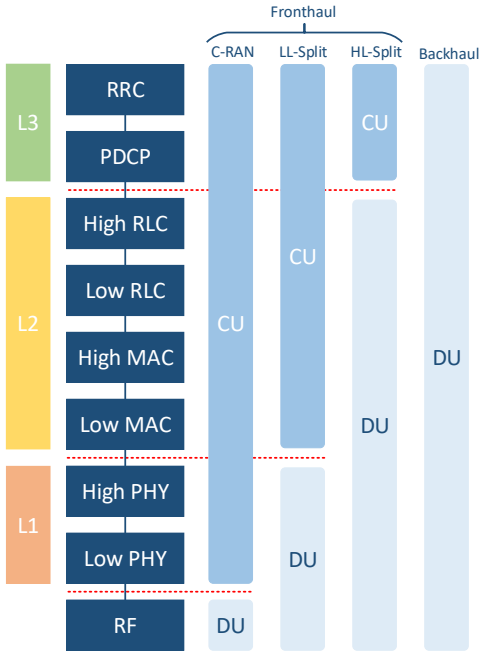


Figure 2: Traffic profiles

- Transport capacity requirements for fronthaul dramatically increase in 5G, due to larger bandwidth and increased number of antenna elements.
- Support of Massive MIMO also implies latency requirements that preclude centralized processing.

Given that C-RAN is a mechanism intended to reduce deployment costs while providing performance gains, 3GPP has analyzed different options for the functional split between centralized and distributed units, in terms of transport requirements (e.g. capacity, latency) and supported features (e.g. CoMP, beamforming). Seven options (in addition to CPRI) were considered in [6]. Currently three of the possible splits were down-selected for implementation. Fig. 2 shows the LTE protocol stack overlaid with the three functional split options, while Table 2 provides a summary of their features [15]:

2. This number comes from the LTE HARQ timer (1 ms) plus the time required to process the CPRI signal, leaving a mere 100 μs transport delay budget.

- Option 2 (PDCP/high RLC) as the Higher Layer (HL) split point (called the F1 Interface).
- Option 6 for MAC/PHY split as one of the candidates for the Lower Layer (LL) split. Option 6 has been defined by the Small Cell Forum as nFAPI (network Functional Application Platform Interface) [16], oriented towards the support of small cells baseband processing virtualization.
- Option 7 for intra-PHY split as the other contending candidate for the lower layer split.

Although currently there is ongoing discussion on the exact lower layer split to be used in 3GPP 5G networks, there is a clear need for protocols to carry such mix of backhaul/fronthaul (that is, crosshaul) data. This data will be packetized and *fronthaul and backhaul traffic will coexist in the same transport network*. This need has been already acknowledged by the industry that is working on new protocols such as eCPRI [4] and Next Generation Fronthaul Interface (NGFI, IEEE 1914 [17]). Both of these protocols allow the use of Ethernet as transport technology. IEEE 802 is also working on extensions to improve the transport of the fronthaul traffic at the IEEE 802.1 Time Sensitive Networking (TSN) [18] working group, which has its counterpart at the IP level in the DetNet Working Group [19] of IETF. §8 makes a proper review of standardization work on this topic. **Designing and integrating a crosshaul forwarding element able to transport such heterogeneous crosshaul traffic is precisely the goal of this work.**

3 Theoretical Analysis

In this section we describe an analytical framework for the optimized design of a crosshaul network based on our previous work presented in [11], [12]. The theoretical analysis is introduced for the sake of completeness and to better motivate the design of our XFE and crosshaul network. The interested reader can find more details of this theoretical analysis and related ones in the literature revision we present in §8.

3.1 System Model

We model the operation of a radio access point as a chain of functions that process signals and data traffic to and from users sequentially, as depicted in Fig. 2. Some of these functions, such as L3 functions, may run into virtual machines (VMs) at either CUs or DUs; while others, such as L1 functions, may require specific hardware. We consider a crosshaul network with a set $\mathcal{N} = \{1, \dots, N\}$ DUs/RRHs and $\mathcal{B} = \{b_1, \dots, b_N\}$ CUs. We assume DUs and CUs are paired accordingly and let b_n denote the CU associated with DU n. We model the crosshaul network connecting DUs to CUs as a graph $G := (\mathcal{I}, \mathcal{E})$, where $\mathcal{I}$ is the superset of routers or switches (XFEs in our setup), RUs and CUs. Each edge in $(i,j) \in \mathcal{E}$ is characterized by its bit-rate capacity $c_{i,j}$ and the latency $d_{i,j}$ it introduces. Moreover, we let $p_{n,k} := \{(c_n, i_1), (i_1, i_2), \dots, (i_L, n) : (i_i, i_j) \in \mathcal{E}\}$ denote a network path k connecting DU n and its corresponding CU, characterized by a path latency equal to $d_{p_{n,k}}$, and $\mathcal{P}_n = \bigcup_k p_{n,k}$ denote the set of all possible paths between DU n and its CU.

With no loss in generality, we focus on the downlink and dimension the system so we are able to route all the network load incurred by all the eNBs in the system at maximum capacity (other strategies can be applied). In this way, we need to route a set of N flows between each DU-CU pair and the bit-rate load $R^{(n)}$ of each flows flow $n \in \mathcal{N}$ is determined by the functional split chosen for eNB n, as presented in Table 2. Moreover, we need to respect certain path latency constraints that are also determined by each choice of functional split. As a result, we can model our crosshaul routing problem with the optimization problem presented in the sequel.

3.2 Functional splits

Let $\mathcal{F}_n = \{1, 2, \dots, F_n\}$ collect all possible functional split choices ($F = 8$ in Fig. 2) supported by eNB n. Let $x_f^{(n)}$ determine whether functional split option (see Fig. 2) is chosen from eNB n ($x_f^{(n)} = 1$) or not ($x_f^{(n)} = 0$). Evidently, only one option for each eNB must be selected, i.e.,

$$\sum_{f \in \mathcal{F}_n} x_f^{(n)} = 1, \qquad \forall n \in \mathcal{N} \tag{1}$$

3.3 Routing decisions

Let $r_p^{(n)}$ denote the amount of traffic allowed to flow through path $p \in \mathcal{P}_n$, serving the pair DU-CU n. Of course, routing decisions must satisfy network link capacity constraints:

$$\sum_{n \in \mathcal{N}} \sum_{p \in \mathcal{P}_n} r_p^{(n)} I_p^{i,j} \leq c_{i,j}, \qquad \forall\, (i,j) \in \mathcal{E} \tag{2}$$

where $I_p^{i,j} = 1$ if path p includes link (i,j) or $I_p^{i,j} = 0$ otherwise.

Moreover, we shall guarantee that the whole flow from the CU-RU n pair is transported across all possible paths between them:

$$\sum_{p \in \mathcal{P}_n} r_p^{(n)} = \sum_{f \in \mathcal{F}_n} \hat{R}_f x_f^{(n)}, \qquad \forall\, n \in \mathcal{N} \tag{3}$$

where $\hat{R}_f$ is the amount of flow emanating from functional split f (see Table 2), i.e., $R^{(n)} = \sum_{f \in \mathcal{F}_n} \hat{R}_f x_f^{(n)}$.

Finally, our routing choices shall also satisfy the latency constraints of the selected functional split. To accommodate this constraint, we can simply force zero flow bit-rate across paths that exceed the required latency. In this way, variables $x_f^{(n)}$ also determine which paths from $\mathcal{P}$ are eligible to route flow for each eNB. As in [12], we partition each set $\mathcal{P}_n$ into F overlapping clusters of paths $\mathcal{P}_n^f$ that *violate* the latency constraint of each split $f \in \mathcal{F}_n$. In this way, for instance, if $x_F = 1$, then $\mathcal{F}_n^F$ includes all paths with latency exceeding 250 ms. Note that, as latency requirements become more demanding as the amount of function centralization (f) grows, $\mathcal{F}_n^1 \subseteq \mathcal{F}_n^2 \subseteq \cdots \subseteq \mathcal{F}_n^F \subseteq \mathcal{F}_n$. Hence, in general, we shall comply with

$$\sum_{p \in \mathcal{P}_n^f} r_p^{(n)} \leq M \left(1 - x_f^{(n)}\right), \qquad \forall f \in \mathcal{F}_n, \forall n \in \mathcal{N} \tag{4}$$

where $M \gg 0$ is a large number, a method commonly known as Big-M method [20].

3.4 Problem Formulation

Putting all the above together, our problem can be summarized as the following Mixed-Integer Linear Problem (MILP):

Problem 1 (Crosshaul Dimensioning Problem)**.**

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{r}} \quad & \sum_{n\in\mathcal{N}} \sum_{f\in\mathcal{F}_n} c_f x_f^{(n)} \\ \text{s.t.} \quad & (1),(2),(3),(4) \\ & r_p^{(n)} \geq 0 \qquad \forall p\in\mathcal{P}_n, \forall n\in\mathcal{N} \\ & x_f^{(n)} \geq 0 \qquad \forall f\in\mathcal{F}_n, \forall n\in\mathcal{N} \end{aligned}$$

where $\boldsymbol{r} := (r_p^{(n)} \mid p \in \mathcal{P}_n, n \in \mathcal{N})$ and $\boldsymbol{x} := (x_f^{(n)} \mid f \in \mathcal{F}_n, n \in \mathcal{N})$ are decision vectors collecting all decision variables. Moreover, c_f is a cost value associated to functional split f. Typically, we would assign lower costs to splits that centralize more eNB functions, i.e., $c_1 > c_2 > \cdots > c_F$. This mixed-integer lineal problem (MILP) can be easily proven to be NP-hard by reduction to a knapsack problem and, to solve it, standard techniques can be applied such as the Benders decomposition method used in [11]. Moreover, different flavors of the above problem can be devised, e.g., to optimize computing cost as in [11], placement of edge computing functions as in [12] or to support unsplittable flows as in [10]. We focus on the simplest optimization problem of them all to simplify the motivation of our Crosshaul Forwarding Element design.

3.5 Analysis

We now use the above formulation to analyze a real network topology of a real operator and study the highest amount of centralization achievable upon different network conditions. To this aim, we use a topology based on "Italy topology" in [11], descriptively represented in the graph in Fig. 3, which consists in an urban network with 200 DUs/RRHs, 1 CU and over 1300 aggregation or switching nodes (XFEs in our paper) in a large city in Italy. To assess our optimization problem in general settings, we set up the network links with different capacities and latencies, spanning from 20 Gb/s to 500 Gb/s and 0 to 100 μs, respectively, and solve the optimization problem described in Problem 1 using the approach proposed in [11] (mildly modified for our specific problem) for each topology parametrization. Furthermore, we use the same computing costs (c_f in Problem 1) as in [10].

The results presented in Fig. 4 show the *centralization degree*, defined as the ratio of eNB functions that are centralized in the CU, with a color mapping and a gradient palette (e.g., C-RAN configuration for all eNBs in the system yield a centralization degree equal to 1 and is represented in yellow; pure D-RAN for all eNBs where all the functions are distributed at the DUs result in a centralization degree equal to 0 and is represented in dark blue). Evidently, the highest amount of centralization is achieved for very-high capacity very-low latency links. More interesting is the fact that *small changes in network conditions (link capacity, link delay) induce changes in the maximum amount of centralization permitted by the system*. For instance, a change in link delay from 16 to 20 μs causes the system to lose 20% of centralization degree,

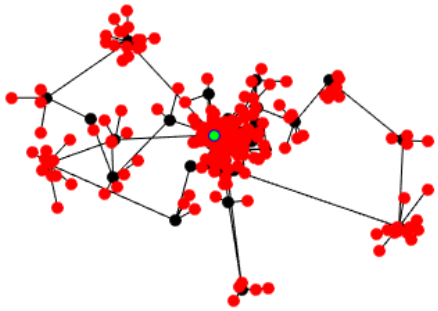


Figure 3: Representation of "Italian topology" [11]. Red dots are RUs, white dots are aggregation points (switches, routers) and the green dot is the CU.

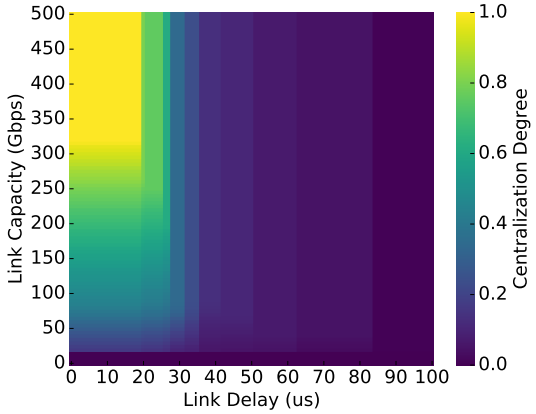


Figure 4: Maximum centralization degree for the topology shown in Fig. 3 parametrized with a variety of link capacities and link delays.

forcing 12.5% of eNBs to lower their functional split from C-RAN (split 8 in Table 2) to split 2. Capacity changes render more gracious functional split changes: 100% of eNBs are allowed a C-RAN configuration (centralization degree equal to 1) when the link capacities are equal to 320 Gb/s, 95% (and centralization degree 0.93) with 300 Gb/s and 50% (and centralization degree 0.46) with 100 Gb/s (given negligible link latency).

In light of our results and those in the related literature [10]–[12], it becomes evident that **there is not a one-size-fits-all Crosshaul configuration, which motivates the need for a switching technology design able to accommodate flows from heterogeneous functional splits in a flexible manner—precisely the goal of this paper.**

4 Crosshaul Forwarding Element design

The transport capacity required by CPRI is overly high for LTE networks; and it dramatically increases with 5G networks, where higher bandwidth and massive MIMO are envisioned, requiring a capacity of several tens or even hundreds of gigabits per second. As an example, an 8x8 MIMO antenna covering four sectors produces 32 antenna carriers, which translate into around 160 Gb/s for 100 MHz bandwidth channels [13].

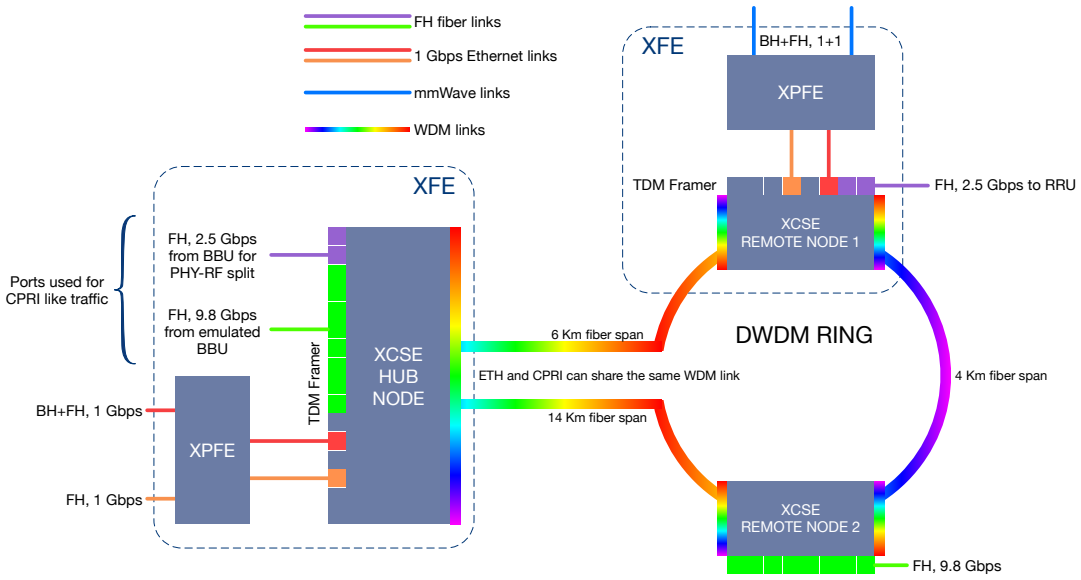


Figure 5: Crosshaul Forwarding Element (XFE)

Multiple academics, industrial companies and standardization bodies have been working on the packetization of fronthaul traffic, allowing the encapsulation and fostering the integration of fronthaul and backhaul segments in a unified transport substrate, the Crosshaul. We review this (mostly theoretical) work in §8). Different physical technologies (e.g., copper, fiber, mmWave, etc.) can be used to develop this common substrate depending on the traffic requirements, increasing flexibility, scalability and efficiency, while reducing the cost of 5G transport networks.

In order to realize this unified transport for backhaul and fronthaul, **we have designed and developed the Crosshaul Forwarding Element (XFE), which is a multi-layered switching node formed by packet and circuit switch layers controlled by software and a variety of physical (PHY) technologies, namely, fiber, copper and wireless.** The packet switch component of the XFE is called Crosshaul Packet Forwarding Element (XPFE), while the circuit switch component is called Crosshaul Circuit Switching Element (XCSE). The XFE enables a unified transport network by combining the flexibility of Wavelength Division Multiplexing (WDM) with the benefits of packet switching over both wire (copper) and wireless (mmWave), integrating different traffic types with diverse transport requirements. Fig. 5 illustrates a small network comprised of two fully-fledged XFEs (comprised of one XPFE and one XCSE) and one standalone XCSE connected with a fiber ring. Furthermore, to cope with the different requirements on multi-tenancy and isolation, the XFE design leverages on IEEE 802.11ah Provider Backbone Bridging (PBB) [21] framing as the encapsulation choice of the network-to-network interface (NNI) (details follow).

In the following, we explain the different physical (§4.1) and link (§4.2) technologies that comprise an XFE.

4.1 Physical technologies

4.1.1 Optical Fiber

The stringent requirements of some functional splits (e.g. CPRI) require the use of fiber optics in order to achieve the minimum delay and maximum capacity possible. Single Mode Fiber (SMF) is the main physical technology used by XCSE and it is comprised of the following components:

- WS-WDM-PON: it is transparent WDM transport solution compatible with SDN-based control plane implementation, it is used to connect core and aggregation network with access devices and client premises (DUs, small cells, distribution point units, optical terminations, etc);
- WS-WDM-OLT: this is based on a L2 Ethernet OpenvSwitch (NXP LS2088ARDB evaluation board) and the optical modules needed to adapt signals from metro network to the WDM-PON;
- WS-WDM-ONUs: tuneable 10G WDM clients; they are the optical WDM termination that represents an optical distribution point unit for allocating the available resources among devices and client premises.

4.1.2 Copper

In order to provide a cost-efficient solution in the presence of less stringent delay and bandwidth requirements, we integrate 1-Gbps copper PHYs. To this aim, we use DPDK (Data Plane Development Kit), a set of libraries to accelerate packet processing workloads, which allow us to forward packets with minimal CPU usage footprint. The use of DPDK in our XPFEs allows us to meet the requirements of the MAC-PHY split in high-load scenarios.

4.1.3 mmWave

Microwave communication technology has indeed been successfully integrated in backhaul networks in 3G and 4G mobile systems, i.e., to transport backhaul traffic only. Our work strives to integrate multiple technologies that are able to transport both backhaul and fronthaul traffic in an integrated manner. In this way, due to the limited amount of bandwidth available for microwave links, this technology can only be used for the less demanding flows. To address this issue, XPFE also integrates Millimeter Wave (mmWave) PHYs specifically tailored for scenarios where deployment

of fiber optics is overly expensive or simply unfeasible. In particular, we design two different mmWave-based PHYs operating on 60-70 GHz frequency range, EdgeLink™ and Fast Forward.

EdgeLink™ (EL). This is an integrated mmWave SDN-controlled mesh transport system with a centrally controlled mesh software platform that resides on top of a WiGig MAC/PHY solution. The system utilizes a WiGig/802.11ad baseband (Peraso PRS4601 60 GHz baseband IC) that operates at 60 GHz (PRS 1126 60GHz Radio IC) with a high-gain antenna for long range transmission, up to 1 Km.

The mmWave node integrates a 10 GigE access port for Point of Attachment (PoA) of various devices, such as C-RAN HL fronthaul radio units, LTE eNB small cells, and 802.11 access points which require transporting of traffic with low latency and high throughput. EL is also a low-cost alternative to fiber and microwave solutions. The EL system consists of at least one gateway (GW) node that associates with a set of non-gateway nodes (NGW) and is the connection point to a broader transport network, such as the proof of concept system showed in this paper. Each node consists of one Processor Unit and up to 2 Antenna Units (38dBi high gain antenna for long range up to 1 Km operating in the 57-66 GHz frequency band). Therefore, each node will have a maximum of 2 sectors. The Processor Unit contains a high-performance computing, evaluation and development platform (NXP 1043A) that includes a 64-bit quad-core ARM processor, high speed ports that include a 10GigE Ethernet and 2 super-speed USB 3.0 type A ports.

Each EL sector can be configured dynamically as either an Access Point or as a station by a centralized SDN OpenFlow Controller. The EL mesh software in the nodes communicate with the SDN controller to manage the nodes and sectors to realize key EL mesh features that include network discovery, neighbor selection and transport slice service level management that can support multi-tenant service level agreements (SLA) and priorities. The EL mesh software solution incorporates in-band signaling to support configuration of the mesh network, traffic flows and alternate paths. The SDN enabled mesh topology ensures fault tolerance and high-availability. The mesh software platform includes a rate adaptation system, this system adapts the rate dynamically based on the packet loss using modulation and coding scheme (MCS) indexes from 1 to 9.

Fast Forward (FF). This solution is specifically designed for extremely low latency communication, such as the LL fronthaul profiles (see Fig. 2). The FF platform comprises a commercial off-the-shelf field programmable gate array (COTS FPGA) carrier board mated to a mezzanine card with high-speed analog to digital converters (ADCs) and digital to analog converters (DACs). These interfaces operate in the frequency of 70 GHz with a 1 GHz channel bandwidth and 2 transceivers and antennas. The baseband processing implemented in the FPGA is based on the 802.11ad single carrier physical layer (SC PHY). However, medium access is schedule-based, i.e., not contention-based like conventional WiFi. This is a time division duplex (TDD) system with 1-ms timeslots and 100-ms time synchronization intervals. The baseband operates at 625 Msymbols/sec and supports modulation and coding scheme (MCS) indexes 2-12 which utilizes binary phase shift keying (BPSK), quadrature phase

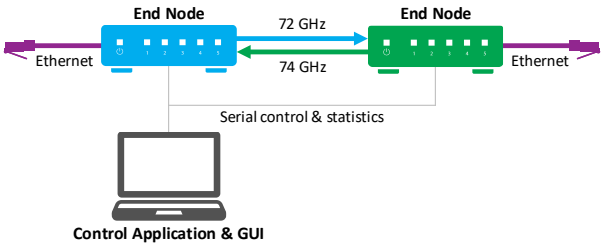


Figure 6: Fast Forward system configuration

shift keying (QPSK), and 16-level quadrature amplitude modulation (16-QAM) modulations with 1/2, 5/8, 3/4, and 13/16 code rates. The baseband supports a 1500-byte maximum transmission unit (MTU). The antenna is an 8x8 patch providing at least 20 dBi gain over the operational frequency range. This allows for a maximum link distance of 15-95 meters depending on the MCS selection.

Each FF platform contains 2 separately controlled sectors which allows configuration as either an end or a relay node. The simplest system configuration, as used in the proof of concept scenario described in this paper, consists of 2 end nodes and no relay nodes. One or more relay nodes may be added to the link for either sector to extend the distance at the cost of additional latency per hop.

The FF configuration used in our proof of concept system provides the highest throughput possible while meeting the latency budget allocated to the mmWave link for the C-RAN LL (MAC-PHY) split, i.e., maximum $250\mu s$. Two platforms configured as end nodes were utilized. Each sector was set up to send data in only 1 direction, one sector for uplink and the other for downlink, to minimize link latency variation. This frequency division duplex (FDD) system configuration provides the lowest possible latency and symmetric throughput, needed for the LL FH. A diagram of the system configuration is shown in Fig. 6. Packets were sent over the normal pathway allowing the highest throughput with some tolerance for large bursts of data.

4.2 Link layer technologies

4.2.1 Crosshaul Packet Forwarding Element (XPFE)

The XPFE is built as an OpenFlow software switch based on a modified version of the Lagopus[3] vSwitch. The XPFEs are based on a packet processing pipeline defined by SDN controllers depending on the configuration and requirements of the network. The specific pipeline implemented in the XPFEs enables encapsulation, decapsulation, and forwarding of PBB frames, taking the design presented in [22] as a basis. To manage the traffic correctly, ports are divided logically into two groups:

- UNI (User Network Interface): ports connecting end-hosts. UNI ports encapsulate and decapsulate PBB frames.
- NNI (Network Network Interface): ports connecting different XPFEs. NNI ports forward PBB frames.

The baseline pipeline design is depicted in Fig. 7. In this design, we have a total of $3 + 7|T| + 3$ Flow Tables (tables, from now on), where T is a set containing all coexisting

3. http://www.lagopus.org/

Table 3: Pipeline tables summary

Table	Function
0	Separate frames by ingress port (NNI/UNI)
1	Identify the tenant
2	Remove PBB header and record tunnel id in metadata
<t,0>	Determine the priority based on VLAN PCP and record in metadata
<t,1>	Determine the service based on VLAN VID
<t,2>	Access Control List (ACL) to control the ingress traffic
<t,3>	Apply tenant policies, e.g. limit ingress traffic to a specific bandwidth
<t,4>	Determine if the frame has to be forwarded (decapsulation), encapsulated or is multicast
<t,5>	Forward frame to all UNI ports In case the destination address is a multicast address
<t,6>	Add PBB header to the frame
252	Set the PCP field in the outer VLAN based on the priority encoded in metadata
253	Enqueue the frame
254	Determine the forwarding port
-	Send the frame

tenants. The first 3 tables (indexed 0, 1 and 2 in Fig. 7) correspond to ingress tables that are common across tenants. Similarly, the last 3 tables (labeled 252 to 254) are common egress tables. In between, each tenant has 7 tables that are labeled <*t,x*> where $x = [0, 6]$ representing tables for flow processing. The task of each table is summarized in Table 3.

In this pipeline, packets can follow three different paths, encapsulate, decapsulate and forward traffic:

- Encapsulation: the tenant which the packet belongs to is identified in table 1 based on source MAC addresses. Its priority is selected in <*t,0*>, and stored in metadata. The packet must be included in one of the services provided by the tenant, which is determined based on the customer VLAN and the ingress port in table <*t,1*>. Access control and policing is performed in tables <*t,2*> and <*t,3*>. Then, packets are encapsulated in table <*t,6*> and the PCP (Priority Code Point) field is set in the outer VLAN header based on the priority encoded in metadata in table 252. Finally queue and out port are selected in tables 253 and 254.
- Decapsulation: upon arrival, the PBB header of the packet is removed in table 2 and some metadata such as the *tunnel id* is stored for future use. The egress port decision is taken in table <*t,4*>, while in case of multicast, the packets are sent to table <*t,5*>.
- Forwarding: to speed up the process of forwarding, we use simple lookup of the VLAN ID and MAC address in table 254.

After the pipeline was implemented, and considering latency is of critical importance in this application, a detailed analysis of the latency of the pipeline was conducted. Measured operations included:

- Match: We evaluated the impact of having matching a flow within a table. Results show an average added delay of $0.008\mu s$ per match.
- Set field: We evaluated the impact of rewriting a field in the packet header with the set field action. Results show an average delay of $0.6\mu s$ without packet loss with this operation.
- Push and pop PBB: We measured the delay and packet loss impact of the PBB encapsulation and decapsulation procedures. The results show an average extra delay of $0.5\mu s$ when these operations are performed for a flow.
- Go to table: We evaluated the impact of traversing 1 to 10 tables by including *go to table* operations. Results show an added average delay of $0.4\mu s$, without packet loss, when a maximum of 4 tables are traversed. For flows going through 5 or more tables we experienced packet loss and average delay of one flow, minimum across flows, of $152.3\mu s$ for input rates higher than 2 mega-packets per second (Mpps).
- Write metadata: This action is used to store information that follows the packets between tables. We measured the delay and packet loss impact of reading a value from the packet header and storing it in one of the metadata registers of the switch. Results show maximum average delays across all flows of $1\mu s$ without packet loss for low input rates. For input rates higher than 2 Mpps, packet loss occurs, suffering an average delay across all flows of $193.96\mu s$.

In light of the above results, a redesign of the pipeline in order to reduce latency is needed (particularly in the presence of the most stringent functional splits). To this aim, we first make the observation that an extended use of "Go to table" and "Write metadata" actions has a severe impact on the packet processing delay. To reduce the impact on the delay of these two actions, we redesigned the pipeline so the aggregate of all operations is carried out in only two tables. In this way, we minimize the number of tables traversed by a packet when being processed. In addition, all usage of the "Write metadata" action has been removed. To store information between tables, we use the "set field" action over the *tunnel id* metadata, which incurs in low extra delay (see above). Fig. 7, shows the resulting pipeline after its redesign, showing two colored regions, one in green for ingress processing (tenant separation, access control, decapsulation, etc.), and a second one (in yellow) for output processing (encapsulation, port and queue selection, etc.). Similarly to the baseline pipeline design, the operation of the redesigned version can be decomposed into three paths:

- Encapsulation: In the first table (green), packets are matched by ingress port and destination MAC address and then tagged according to the tenant policies and priorities using the *tunnel id* metadata. The second table (yellow) pushes the PBB header and selects the egress port matching by ingress port and *tunnel id*.
- Decapsulation: In the first table (green), packets are matched by ingress port, destination MAC address and VLAN tag and the outer VLAN tag is recorded as a *tunnel id* metadata and the PBB header is removed. The egress port is determined in the second table (yellow) matching by the destination MAC address and *tunnel id*.
- Forwarding: In the first table (green), packets are matched by ingress port and destination MAC address and tagged using the *tunnel id* metadata. In

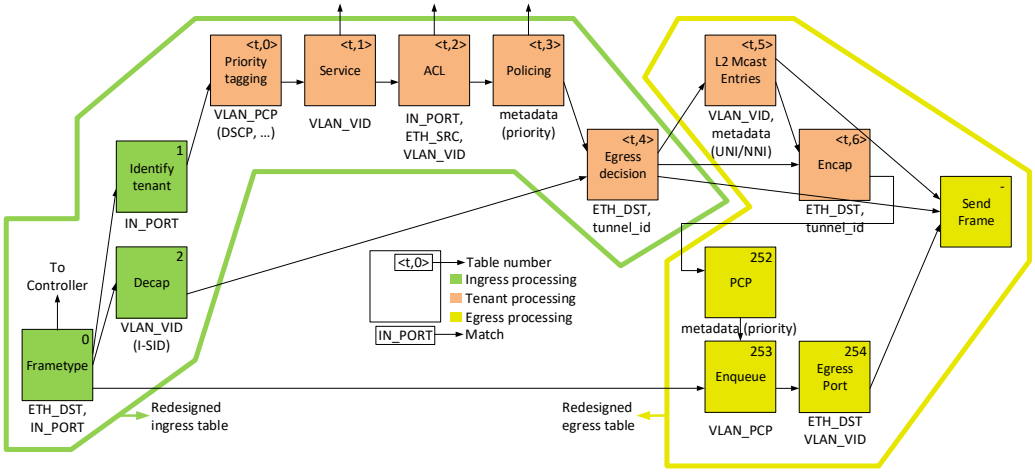


Figure 7: Initial and redesigned pipeline

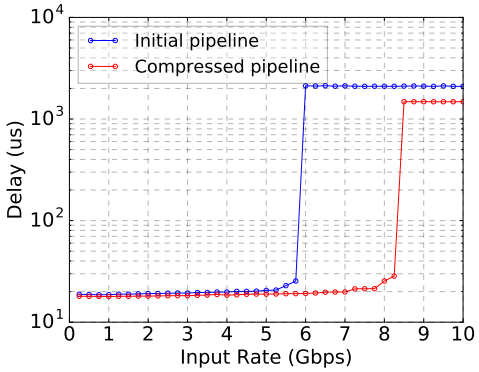


Figure 8: Pipeline delay comparison

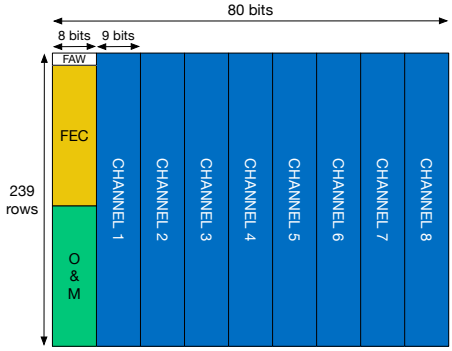


Figure 9: XCSE frame format

the second table (yellow), the egress port is selected based on the destination MAC address and *tunnel id*.

The compressed pipeline has reduced the number of entries per flow from 12 to 2 in the worst case, but conversely, the number of entries per table has increased since all the tenants' flows have been integrated into these two tables. Fig. 8 shows the comparison of the delay measured when using both pipelines in a 2-hop scenario using 10 Gb/s links. The maximum throughput achieved without packet loss increases from 5.75 Gb/s to 8.25 Gb/s with the compressed pipeline. The redesigned pipeline also shows a reduced delay, improving performance; however, at the cost of reducing scalability. In OpenFlow the number of flow entries per table is limited. In the redesigned pipeline, several entries are needed per flow, heavily impacting the number of flows supported. More details and performance metrics about the XPFEs can be found on [23].

4.2.2 Crosshaul Circuit Switching Element (XCSE)

The XCSE enables the use of optic fiber to transmit a mix of fronthaul and backhaul traffic by implementing a novel Time Division Multiplexing (TDM) framing method on top of standard WDM technology. Such TDM method assigns fixed capacity circuits (by choosing the needed slots) for each traffic type, enabling packet and CPRI data to share the same wavelength.

On top of the physical layer (see 4.1), a novel TDM scheduling mechanism, consisting on the following functions, has been implemented:

- Time division framing that allows to mix heterogeneous client traffic (CPRI, Ethernet) on the same wavelength with minimum latency and providing transport synchronization information compliant with CPRI requirements.
- TDM based switching, configurable through SDN techniques, that enables dynamic network reconfiguration, optimizing the aggregation of traffic in the different wavelengths.

The XCSE has been designed considering the specific requirements of CPRI traffic in terms of low latency, high bandwidth and asymmetry control. As CPRI links are synchronous and constant bitrate, the core of the XCSE system is a Time Division switch that allows to implement low latency (a few tens of nanoseconds) and deterministic pass-through time. This switch has been implemented in a Field Programmable Gate Array (FPGA) circuit that includes also the blocks needed for DWDM transmission (Forward Error Correction and serializer/deserializer).

The frame used within the DWDM ring has a period of 1.94 μs, with a bitrate of 9.8304 Gbps and a per channel payload rate (excluding FEC) of 1.10592 Gbps. CPRI signals are encoded using an 8B/9B scheme obtaining a rate saving of 10% with reference to the gross bandwidth. Ethernet words

are re-coded with a 17B/16B format and rate adaption is done with the insertion of bit stuffing. The frame format is shown in Fig. 9. A frame is composed by 239 x 80 = 19120 bits, with a column 8-bits wide dedicated to overhead (Frame Alignment Word (FAW), FEC, O&M signaling) and 8 payload columns (each 9-bits wide) defined "channels". A single channel can map the content of a GBE link or of a 1.2288 CPRI link. Eight channels can map a whole CPRI Option 7 link (9.8304 Gbps) adding FEC and O&M signaling without increasing the bitrate—this is possible thanks to reduced line coding overhead, as described below.

The bit rate of 9.8304 Gbps was chosen to have a line rate directly related to CPRI client bit rate—it is four times the widely used 2.4576 Gbps (CPRI option 3) used to connect a Baseband Unit (BBU) with a Remote Radio Unit (RRU). This simplifies the realization of clock recovery system, that are needed to fulfill the tight frequency synchronization requirements of CPRI (2 parts-per-billion). The frame period was chosen to allow the implementation of a robust FEC (Reed-Solomon 255/239, the same adopted e.g. in ITU G.709-OTN) while keeping the added latency within a value acceptable for CPRI clients. The overall latency introduced end-to-end by framing, FEC coding and decoding, parallel-to-serial and serial-to-parallel conversion, is below 4 μs. The line coding, needed to guarantee adequate balancing and transitions for clock and data recovery, is based on statistical approach (scrambling) like in OTN. The FPGA family used in the proof-of-concept is Xilinx Virtex7 that includes GTX transceivers capable of up to 10 Gbps link rate. A single FPGA was used in each of the XCSE nodes.

The flexibility provided by the XCSE allows the physical integration of traffic with diverse requirements, including legacy CPRI fronthaul and backhaul packets, in the same fiber deployment of the operator. This functionality is extended with the XPFE, which can be implemented on top of the XCSE to build a complete XFE or be deployed as a standalone packet switch without the XCSE functionality as shown in Fig. 5.

5 Proof-of-Concept System Design

We deploy the scenario depicted in Fig. 10, including three transport *islands* (namely, XPFE, mmWave and XCSE) and one EPC. They are serving four eNBs comprised of three DU/CU pairs (eNB 1-3) and a fully-fledged LTE small cell (eNB 0). DU/CU 1 (eNB 1) has a RLC-PDCP (HL) split, DU/CU 2 (eNB 2) has a PHY-MAC (LL) split and DU/CU 3 (eNB 3) has a PHY-RF split (C-RAN). The testbed is presented in Fig. 11 and a list of the technologies used in this deployment is summarized in Table 1.

5.1 Radio Access Points (eNBs), User Equipments (UEs) and Mobile Core

eNB 0

The LTE small cell depicted in Fig. 10 is comprised by one NEC E-RAN RN-310 Radio Node and one NEC E-RAN SN-9000 Service Node,[4] both co-located at the same site and connected by a 1Gb/s Ethernet connection (the figure illustrates this pair as a single node). Each radio node supports up to 32 active users over LTE and 128 RRC connections. We configure the eNB with a 20 MHz channel bandwidth and a 2x2 MIMO configuration, and it thus supports a maximum aggregate throughput of 150 Mb/s in downlink and 50 Mb/s in uplink. The SN-9000 Service Node oversees the radio node control and management. It can handle a maximum number of 100 Radio Nodes powered via Ethernet (PoE), up to 8000 simultaneous sessions and 1 Gb/s of aggregated backhaul. As shown in Fig. 10, the small cell is connected to the EdgeLink™ in the mmWave transport island (details later).

eNB 1

This eNB is divided into DU 1 and CU 1 with a RLC-PDCP (HL) functional split. Both DU 1 and CU 1 are general-purpose PCs equipped with an Intel Core i7-3610QE and 8 GB of RAM. CU 1 deploys the eNB's PDCP, RRC, S1AP and GTP-U endpoint layers while DU 1 deploys the MAC layer and an emulated PHY with an attached emulated UE. All the software is provided by Core Network Dynamics[5] and is compliant with 3GPP Release 12.

eNB 2

This eNB is divided into DU 2 and CU 2 with a MAC-PHY (LL) functional split. DU 2 is a commodity PC with an Intel(R) Atom(TM) CPU D525 with 4 GB of RAM, whereas CU 2 is in an Intel NUC with an Intel Core i7-6770HQ and 32 GB of RAM. The softwarized protocol stack is also provided by Core Network Dynamics. CU 2 deploys the eNB's MAC layer in addition to RLC and PDCP, and DU 1 simply deploys an emulated PHY layer with an attached emulated UE.

eNB 3

This eNB is divided into DU 3 and CU 3 with a RF-PHY (C-RAN) functional split, that is, CU 3 is a traditional baseband unit (BBU) processing all LTE protocol stack (from PHY to PDCP and GTP-U tunnel endpoint). Both CU[6] and DU[7] are commercial equipment from Ericsson.

Evolved Packet Core (EPC) and User Equipment (UEs)

At the core of our system, we use a virtual EPC (vEPC) from Core Network Dynamics, as depicted in the figure. Our EPC is deployed on a Dell R630 server with two 16-core Intel Xeon CPU E5-2620 processor and 128 GB of RAM, and is directly connected to the XPFE island. In our measurements, we use the EPC as source (destination) of downlink (uplink) backhaul traffic for all eNBs. Our UEs are conventional LTE routers DWR-921[8] for both eNB 0 and 3, with RF cables connecting DU and UE to guarantee isolation. In case of next-generation functional splits (eNB 1 and 2), the UEs are virtual machines connected to the emulated physical layer of the respective DU.

4. https://www.nec.com/en/global/solutions/nsp/sc2/prod/e-ran.html

5. https://www.corenetdynamics.com/products
6. https://www.ericsson.com/ourportfolio/radio-system/baseband
7. https://www.ericsson.com/ourportfolio/radio-system/radio
8. https://eu.dlink.com/uk/en/products/dwr-921-4g-lte-router

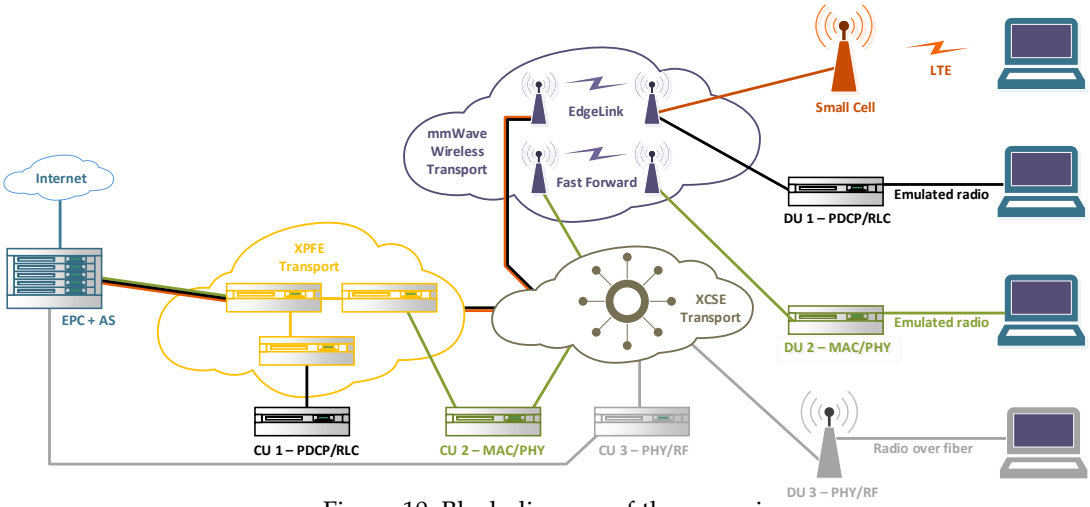


Figure 10: Block diagram of the scenario

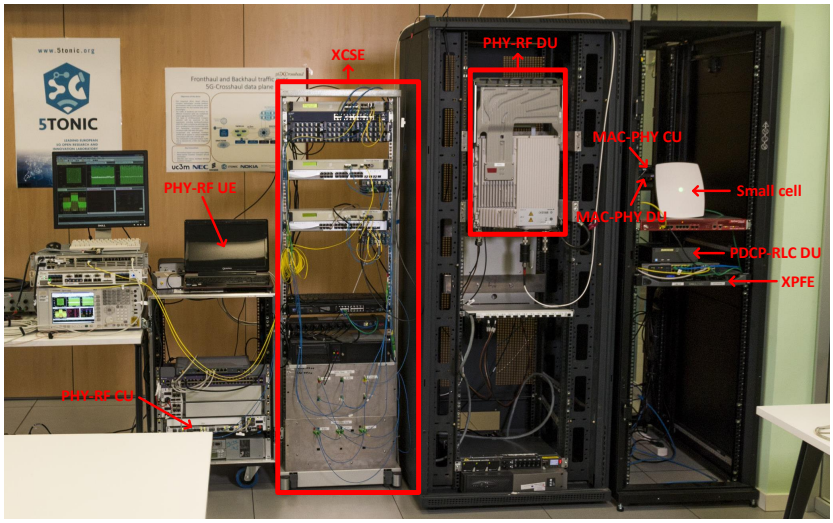


Figure 11: Deployed scenario

5.2 Traffic profiles

The four eNB configurations (splits) in our scenario generate representative heterogeneous traffic patterns for 5G networks. This diverse traffic must be transported across our Crosshaul network. This section briefly details these traffic profiles. Note that, although we focus on the downlink, the explanations can be extended to the uplink case, which we omitted for space reasons. Nonetheless, our experiments in Section 6 consider both uplink and downlink.

Backhaul

Our transport network must carry several backhaul flows, specifically, one downlink flow (and one uplink flow) per eNB. The first downlink one, caused by eNB 0 and marked in orange in Fig. 10, initiates at the EPC and ends at the LTE small cell. The remaining three backhaul flows also initiate at the EPC and have destinations, respectively, CU 1, CU 2 and CU 3, marked in black, green and grey in Fig. 10. All the transport islands deployed in our system carry some backhaul traffic, also as depicted by Fig. 10.

PHY-RF split

As mentioned before, with a PHY-RF split, the CU performs all L1-L2-L3 functionality of the protocol stack (PHY-layer modulation and coding, MAC scheduling, etc.) and sends digitized (raw) I/Q radio samples to its DU. The DU then performs the most basic RF functions (e.g. amplification and other basic analog processing tasks) and irradiates the composed signals. This configuration has been shown to provide certain gains in spectrum efficiency (via interference coordination, for instance) and pooling gains (via pooling computing resources in the CU). Our PHY-RF split flow is based on CPRI [13], introduced earlier. CPRI uses a serial line interface to transmit the data at a constant bit-rate, i.e., irrespective of the actual user load. In addition, CPRI requires tight synchronization between CU and DU, with an accuracy of 8.138 ns and, as shown in Table 2. Latency, jitter, throughput and reliability requirements are very demanding. The delay tolerance is as low as 250 μs, and the throughput demand is, for a configuration of 20 MHz bandwidth and 2x2 MIMO, around 2.5 Gb/s. For this reason, the deployment of CPRI interfaces must be done with high-capacity optical fiber and point-to-point links, which ren-

(a) Fast Forward node

(b) EdgeLink™ node

Figure 12: mmWave transport solutions

ders this configuration rather costly and motivates relaxed functional splits, analyzed next. In the scenario evaluated in this paper, we employ a DU/CU (DU 1 and CU 1) with PHY-RF split (traditional C-RAN) provided by Ericsson. Given the stringent network requirements, the C-RAN flow, marked in grey in Fig. 10, only traverses the high-capacity (delay deterministic) XCSE transport island (details later) to connect to its CU pair.

Low Layer (MAC-PHY split)

Due to the tight coupling of Hybrid Automatic Repeat Request (HARQ) protocol located in both MAC and PHY layer, the latency tolerance of this flow is the same as C-RAN, $250\mu s$. However, its throughput requirements are substantially relaxed, from 2.5 Gb/s to 152 Mb/s (downlink) or 49 Mb/s (uplink), assuming the same eNB configuration we discussed before (20 Mhz and 2x2 MIMO). In our scenario, the LL-split flow is marked in green in Fig. 10 and travels through both XCSE and mmWave transport islands.

High Layer (PDCP-RLC split)

Packet Data Convergence Protocol (PDCP) provides services for header compression and security. PDCP removes the IP header and adds a token of 1-4 bytes reducing the header size. Radio Link Control (RLC) does the concatenation and segmentation of the received segments, reordering, duplicate selection and protocol error detection. As a result, the transport requirements are further relaxed as compared to the LL split. The throughput requirements have a small decrease (around 1 Mb/s). However, the latency tolerance is now similar to that of a regular backhaul flow. As depicted in Fig. 10, marked in black, this flow traverses the XPFE transport island in addition to the XCSE and the mmWave islands.

5.3 Transport technologies

Crosshaul Circuit Switching Element (XCSE)

As explained in Section 4, our XCSE transport technology has been specifically designed to allow both (CPRI-based) C-RAN traffic and (packet-based) flows from other functional splits to share the same fiber by appropriately scheduled TDM frames. For this reason, we deploy this transport technology as shown in Fig. 10 where diverse flows must be processed. Specifically, the XCSE island shown in Fig. 10 consists of a complete fiber ring built by three spans, as shown earlier on in Fig. 5; each span with two fibers (one for each direction). The span lengths are:

- 4 kilometers from the XCSE Remote Node 1 to the XCSE Remote Node 2 (see Fig. 5);
- 14 kilometers from the XCSE Remote Node 2 to the Hub (Fig. 5)
- 6 kilometers from the Hub to the XCSE Remote Node 1 (Fig. 5)

The span lengths of the fiber rings were chosen to reproduce scenarios that could be found in a real fiber ring deployment in an urban area. The adopted lengths were chosen taking into account the following considerations:

- A minimum span length higher than 1 Km. This assumption implies that different nodes will be housed in different operator's premises, distributed around the territory, not too much close to each other;
- Certain amount of fiber attenuation to stress the system while keeping the maximum span from Hub to any Remote Node below 20 Km (considering also protection paths) to limit the round trip delay due to fronthaul constraints ($200\mu s$).

Crosshaul Packet Forwarding Element (XPFE)

As introduced in Section 4, our XPFE technology is specifically designed to transport packet-based protocols (backhaul, LL and HL splits) with minimum delay. In our deployment, each copper-based XPFE runs in a Dell PowerEdge R430 server powered by an 8 cores Intel Xeon E5-2609 processor, 16 GB of RAM and several 10-Gbps and 1-Gbps Intel DPDK compatible network cards. Our XPFE transport island oversees processing backhaul and HL-split flows.

Moreover, wireless XPFEs based on mmWave physical technology as introduced in Section 4 are also deployed. As depicted by Fig. 10, a EdgeLink™ (EL)link and a Fast Forward (FF) link are deployed within the mmWave island in our testbed. On the one hand, EL has the role of transporting less delay-sensitive backhaul and HL (PDCP-RLC split) traffic flows, connecting to the XCSE. On the other

hand, FF is used to transport the LL (MAC-PHY split) traffic flows, which are more demanding in terms of delay than the HL or the backhaul flows. Also, similar to EdgeLink™, Fast Forward connects to the XCSE.

6 Experimental Results

6.1 Methodology

The main tool we have used in our performance evaluation campaign is `MoonGen` [24]. `MoonGen` can generate network traffic profiles over 10-Gbps using only one CPU core. It is based on Intel's Data Plane Development Kit (`DPDK`) and provides packet loss information and latency measurements with sub-microsecond precision using hardware timestamping. `MoonGen` allows the development of `lua` scripts to adapt the packet generation to the developer needs (change packet size, headers, introduce load traffic, timestamped traffic, etc.). In addition, we have also used conventional network measurement tools like `iperf2` and `ping` to make basic throughput, packet loss and round-trip time (RTT) measurements. To collect one-delay measurements appropriately, we synchronize all the nodes in our system with `PTPv2` and ensure that 8 Mbps out of the aggregate load is timestamped. This gives us enough samples to make delay measurements without affecting the measurement itself.

In the next section, we first analyze each transport technology independently, and then evaluate the system as a whole, including all eNBs, UEs and the vEPC.

6.2 Characterization of transport technologies

We start off by characterizing the performance of each individual transport technology in isolation. To this aim, we deploy MoonGen in a dedicated Dell PowerEdge R430 server with an Intel Xeon E5-2609 with 8 cores, 16 GB of RAM and 1-Gbps Intel DPDK compatible NICs, and send Ethernet flows through each of the Crosshaul transport technologies.

6.2.1 Wireless XPFE (Fast Forward)

Fig. 13 shows the performance of the Fast Forward (FF) link in terms of mean delay and mean throughput using two different MCS indexes 4 and 11 [25]. Fig. 13a and Fig. 13b show the mean delay when the FF is using MCS index 4 and 11, respectively. We first note that, with low input rates, smaller packets have lower delays. This is because there are fewer bits to process. However, when the wireless link capacity is exceeded, internal queues fill up and packet delay increases abruptly, as expected. Due to the over-the-air overhead associated with each packet (STF/CEF, header, inter-packet gaps) it takes more time to clear out the queues when using many small packets rather than fewer larger packets. This behavior can be observed in both figures. An additional observation is that MCS 11 obtains better performance in terms of delay and throughput; also, expected due to the higher line rate. Fig. 13c and Fig. 13d depict the MAC-layer throughput obtained as a function of the input rate for the same MCSs. According to these results, the mean throughput attained to our FF solution is as high as 244 Mbps when using MCS4 and 544 Mbps when using MCS11 for 1500-byte packets. When the wireless link capacity is not exceeded the maximum average delay we achieve using MCS11 is $96.4\mu s$, with an input rate of 500 Mbps and 1500-byte packets, this value is lower than the MAC-PHY split delay requirement ($250\mu s$). We thus conclude that our FF solution is able to transport lower-layer (MAC-PHY) functional splits when large packet sizes are employed.

6.2.2 Wireless XPFE (EdgeLink™)

We now analyze our EdgeLink™ technology and show in Fig. 14 its performance in terms of delay, maximum throughput and packet loss. Fig. 14 depicts the performance using a rate adaptation system to select dynamically the MCS (from MCS1 to MCS9) based on the packet loss. Similar to the previous case for FF, the EdgeLink™ delay performance, shown in Fig. 14a, increases when the link becomes saturated, as expected; and smaller packets attain lower delay. The minimum average delay achieved with 100 Mbps is $230\mu s$, for the rest of the input rates tested the average delay is always higher than $250\mu s$, in any case the EdgeLink™ delay results are always lower than $6x10^3\mu s$ (High/Low MAC split delay requirement). From Fig. 14b we observe a maximum throughput equal to 900 Mbps when using 1000 to 1500-byte packets with no packet loss. Considering these results, we conclude that our EdgeLink™ solution can satisfactorily transport relaxed functional split traffic (MAC split and above) and it is thus suitable for transporting PDCP-RLC split and backhaul traffic.

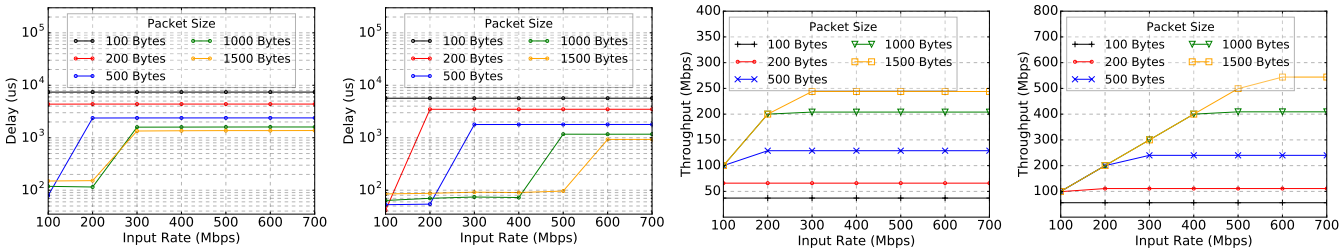
6.2.3 XPFE Link Layer

We now assessed the link layer performance of our XPFE, testing two use cases: *i)* Encapsulation-decapsulation (E-D): In this case, the scenario is formed by two XPFE nodes; one performing encapsulation and another one doing decapsulation. This use case emulates the path followed by the MAC-PHY split and the backhaul traffic going from the XCSE to the EPC passing through 2 XPFEs. *ii)* Encapsulation-forwarding-decapsulation (E-F-D): This scenario is formed by three XPFE nodes; the first node encapsulates traffic packets, the second one forwards packets based on the outer header, and the last one decapsulates traffic packets. This use case emulates the path followed by the PDCP-RLC traffic that goes from the XCSE to the CU passing through three XPFEs.

Fig. 15a and Fig. 15c detail the one-way delay and throughput performance in the E-D scenario. We first observe that the delay increases abruptly and packet loss occurs when packet size is 100 Bytes and the input rate is higher than 500 Mbps. This is so because the XPFE internal queues build up. We experience a maximum mean delay of 31.5 μs with 1500-byte packets.

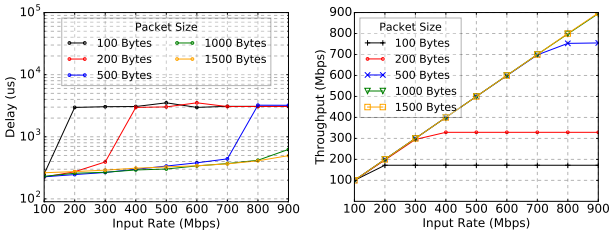
Similarly, Fig. 15b and Fig. 15d depict the delay and throughput performance in the E-F-D scenario. In this setup, packet sizes as high as 200 bytes cause system saturation (and delay increase) when the input rate reaches 800 Mbps or higher. In the remaining regimes, the maximum mean delay (for larger packet sizes) is $66.6\mu s$.

Considering these measurements, we conclude that XPFE is capable of satisfying the delay and throughput requirements of the lower functional splits (as long as packet



(a) Delay with MCS4 (b) Delay with MCS11 (c) Throughput with MCS4 (d) Throughput with MCS11

Figure 13: Fast Forward characterization



(a) EdgeLink™ delay (b) EdgeLink™ throughput

Figure 14: EdgeLink™ characterization

sizes are above 200 bytes), excluding the PHY-RF split due to the low line rate of our XPFE's 1 Gbps NICs.

6.2.4 XCSE

We now show experimental results of our XCSE switching technology in Fig. 16. Fig. 16a details its one-way delay performance. We first remark that the delay performance of our XCSE is substantially improved over the previous mmWave technologies, irrespectively of the input rate. Because the number of bits to process is lower, smaller packet sizes attain better delay. We note that, in this case, the capacity of the XCSE is never reached as it has 10-Gbps interfaces, which explains the fact that we do not observe any saturation point in the range up to the 900 Mbps considered in Fig. 16b. To conclude, given the high capacity and good delay performance of XCSE, it becomes a suitable solution for transporting any functional split flow, including the most demanding PHY-RF split (C-RAN).

6.3 Characterization of traffic profiles

We now characterize the performance of the remaining functional blocks of the scenario depicted in Fig. 10, *without* our transport network in between. This will give us a performance upper bound for comparison. To this aim, we deployed `iperf` on a machine attached to our vEPC and on each of the UEs, and carry out the following experiments:

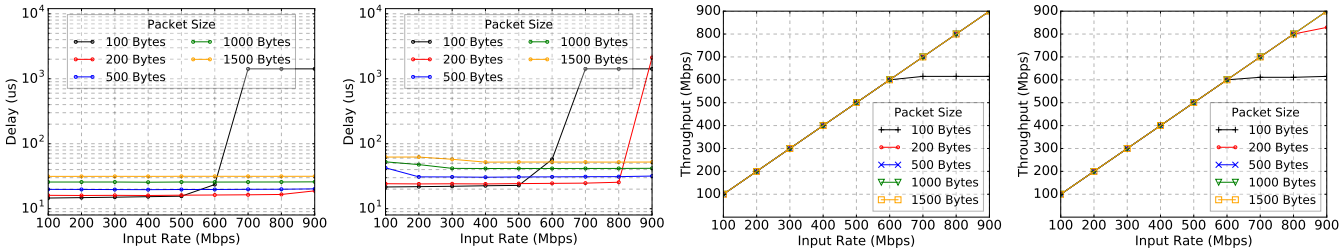
- Downlink and uplink TCP flows: To measure both downlink and uplink throughput in ideal conditions;
- `Ping` without background traffic: To measure round-trip-time (RTT) and ICMP packet loss rate;
- `Ping` with background traffic: To measure RTT under high-load regimes with both downlink and uplink flows simultaneously.

Fig. 17a first presents the throughput CDF (cumulative distribution function) for each of the eNBs with both downlink and uplink flows, independently. We observe that the MAC-PHY split shows the highest MAC throughput in both uplink and downlink, followed by the PDCP-RLC split and then backhaul. Now, Fig. 17b depicts the distribution of our RTT measurements without (solid lines) and with (dashed lines) background traffic. In the former case, we note that the PDCP-RLC split flow experiences the lowest delay, followed by the MAC-PHY and then backhaul. When background traffic is introduced, then the MAC-PHY split flow shows the highest RTT, as expected given that this split incurs in higher CU processing. Finally, Fig. 17c presents the packet loss rate experienced in the same scenarios. From the figure, we can see that some packet loss appears when background traffic is introduced, with the MAC-PHY split flow being the most noticeable case, followed by the backhaul and the PDCP-RLC split.

6.4 End-to-end performance metrics

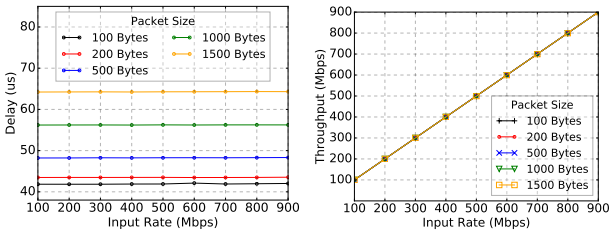
Finally, we evaluate the performance of our transport technologies in the context of the whole system shown in Fig. 10 when all flows contend for resources across the entire Crosshaul transport network. We compare these end-to-end results with the ones obtained in §6.3.

First, Fig. 18a shows the distribution of the throughput performance of each of the flows in the experiment. The MAC-PHY split flow shows in average the highest bitrate load in the downlink, followed by the PDCP-RLC and the backhaul flows. Conversely, the PDCP-RLC shows higher mean throughput in the uplink, followed by the MAC-PHY split and the backhaul flow. Evidently, the throughput performance observed in this experiment is noticeably lower than that shown in Fig. 17. This can be explained by the contention for network resources induced across all flows in this case. We show next, in Fig. 18b, the distribution of RTT samples. Here we can observe than the additional switching nodes of our transport network incurs additional delay when there is no background traffic, as compared to the baseline case in Fig. 17, as expected. Perhaps surprisingly, RTT performance improves (i.e. it is lower) with Crosshaul than in the baseline case when background traffic is present. This is explained by the additional packet loss rate (packets that do not contribute to delay samples) as shown by Fig. 18c. We also note a clear increase in packet loss average and variability across all traffic profiles with and without background flows. This is so because in this experiment the network is in heavy congestion.



(a) 2 XPFE delay (b) 3 XPFE delay (c) 2 XPFE throughput (d) 3 XPFE throughput

Figure 15: XPFE characterization



(a) XCSE delay (b) XCSE throughput

Figure 16: XCSE characterization

7 Lessons Learned

The integration of fronthaul and backhaul network segments in a unified transport network brings flexibility, scalability and efficiency advantages but it is also challenging due to the stringent requirements needed and the integration efforts to accommodate such heterogeneous technologies within a single transport solution.

Beware MTU fragmentation

We unexpectedly experienced overly high delay when packet sizes were large. Our troubleshooting process concluded that the reason lied upon the additional overhead introduced by our Provider Backbone Bridge (PBB) encapsulation (see §4 for details). This extra overhead made long packets violate MTU fragmentation requirements in our switching nodes. The fragmentation process in our scenario introduced a minimum overhead of 6% when adding the PBB header in a fully loaded Ethernet frame. Packet fragmentation issues are sometimes hard to track down and MTU configuration is rarely modified from default parameters although it is usually a cause for issues when dealing with encapsulation technologies.

IEEE 802.1ah (PBB) is incompatible with IEEE 802.11 (Wireless MAC/PHYs)

Our mmWave physical solutions (Edge Link™ and Fast Forward) are based on the standard IEEE 802.11ay, which, perhaps surprisingly, does not support PBB frame forwarding. In order to integrate mmWave technology into our XFE and preserve PBB compatibility (which provides us multi-tenancy and isolation support), we leveraged on IEEE 802.11ak (Generalized Link) standard, which defines mechanisms so that communication links can be established between general link (GLK) stations that are usable as transit links inside an IEEE 802.1Q network.

Synchronize your nodes but beware its overhead

Time synchronization is important for e.g. measuring one-way delay and preserve timely operation across different layers of the radio protocol stack when they are split. However, achieving sub-millisecond synchronization across a large-scale network may be quite challenging. particularly when using in-channel synchronization protocols like PTP. PTP is designed to operate in a relatively benign environment with low delay and jitter. PTP, as NTP, polls the synchronization servers at a frequency depending on the jitter detected in the link, the time synchronization in a high loaded link will increase the jitter and the synchronization exchanges will need to be more often, increasing the overhead. We found out that a proper planning of the synchronization channel (routing of control traffic) is of paramount importance to preserve synchronization without impairing sample measurements and mitigate its impact on the data plane performance.

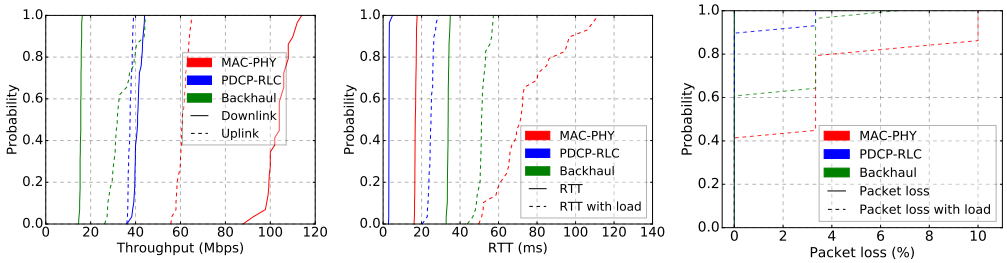
8 Related Work

In the following we revise the related work focusing on standardization and academic research efforts.

8.1 Standardization

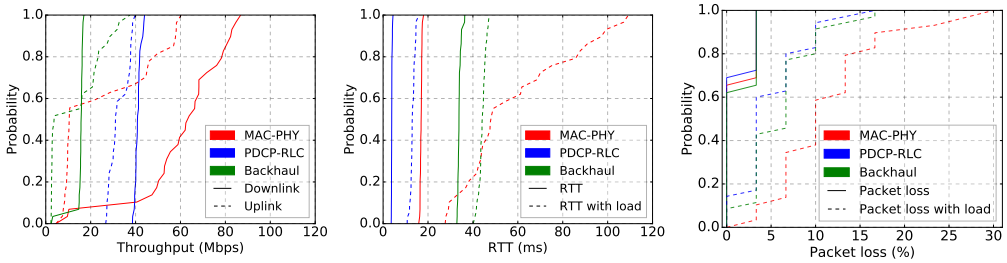
3GPP has been analysing for a long time the different options for the functional split in LTE and how these can be mapped to the 5G-NR specification [6]. Latest efforts in the industry have converged to a common understanding of using PDCP/High RLC as higher layer functional split (known in 3GPP as split option 2) for the interface connecting CUs and DUs (F1 in TS 38470). This functional split will be used for centralisation requiring control and data plane split (allowing aggregation of multiple technologies at the PDCP level). For deployments requiring a higher level of centralisation a split between the High and Low Phy layer (option 7), corresponding to eCPRI, has been chosen. eCPRI is a packet-based fronthaul interface, developed by the CPRI Forum, defined for the transmission of vendor specific radio sampling payloads over any data link-layer protocol to form the transport network, considering the interconnection of a DU and CU by option 7.

In addition to eCPRI, the IEEE 1914 Working Group (NGFI) was created in December 2014. Its target is to define an efficient and scalable fronthaul transport for 5G based on Ethernet and provides specification for the architecture for the transport of mobile fronthaul traffic (e.g., Ethernet-based), user data traffic, and management and control plane



(a) Throughput (b) RTT (c) Packet loss

Figure 17: Independent characterization of traffic profiles



(a) Throughput (b) RTT (c) Packet loss

Figure 18: Simultaneous end-to-end measurements

traffic (P1914.1) and the Radio over Ethernet encapsulation (P1914.3).

Finally, the IETF Deterministic Networking (DetNet) Working Group focuses on deterministic data paths that operate over Layer 2 bridged and Layer 3 routed segments, where such paths can provide bounds on latency, loss, and packet delay variation (jitter), and high reliability.

Analyzing the above options, the use of Ethernet for the fronthaul interface is a common point in all future alternatives for the fronthaul transport protocol. To reduce costs and facilitate the implementation of bridged networks supporting 5G's stringent demands, IEEE 802.1 Time Sensitive Networking (TSN) group is defining a new set of specifications to support bounded low latency and ultra-reliability. The benefits of using these extensions for the fronthaul transport have been already acknowledged by the industry, creating first the IEEE 802.1CM TSN profile for the transport of CPRI (published in 2018) and later extending the scope of this project to also cover eCPRI.

8.2 Research

A flexible split of the RAN functionally has also motivated substantial academic work, e.g., [10]–[12], [26], [27]. In [10], the authors study the joint optimization problem of selecting paths between CUs and DUs *and* the choice of RAN functional splits. This work was extended in [11], [12] to jointly compute the optimal placement of mobile edge computing platforms as well. The results show that existing backhaul networks can rarely support full RAN centralization (C-RAN) but can benefit substantially (up to 45% system cost savings) from a hybrid crosshaul approach over traditional D-RAN.

Despite the theoretical advantages that C-RAN and a flexible functional splitting of the RAN may provide to mobile operators, little work has been published to prove its feasibility in practice. The authors of [26] study the impact of packetization when facing flexible RAN functional splits and present early experimental prototype results in [27]. However, their work focuses on the RAN performance only and omits the need to adapt the segment that connects DUs and CUs (i.e. the Crosshaul) to transport the heterogeneous flows from different splits. The interested reader can find more related literature in [28] and references therein.

To the best of our knowledge, this is the first practical work that demonstrates the feasibility of transporting crosshaul traffic in realistic scenarios and therefore bringing such theoretical gains to production networks.

9 Conclusions

Future generations of mobile systems need to increase RAN centralization in order to achieve their data rate requirements. Classical C-RAN technology is deemed infeasible in most 5G deployments and thus, new RAN functional splits have been defined to enable a smoother migration towards 5G. As a result, the traditional border between fronthaul and backhaul networks blurs, leading to the need for an integrated fronthaul and backhaul transport network. In this paper, we have reviewed the technical challenges to achieve this integration and designed two new network elements, XPFE and XCSE, conveniently integrated into a standalone multi-layer forwarding element (XFE), that enable the convergence of fronthaul and backhaul into a common Crosshaul solution by consolidating a set of heterogeneous physical technologies (copper, fiber, mmWave), tailored to transport flows for a variety of RAN functional splits. We have introduced a prototype of our XFE and validated its feasibility in a proof-of-concept Crosshaul deployment. Our results demonstrate that the solutions designed allow for an effective support of novel and legacy fronthaul functional splits in the same transport infrastructure along with traditional backhaul traffic.

ACKNOWLEDGEMENT

Special thanks to the 5G-Crosshaul and 5G-TRANSFOERMER team, in particular to Jaime Jose Garcia Reinoso, Chenguang Lu, Daniel Cederholm and Jakub Kocur who helped during the experimentation. This work has been partially funded by the EU H2020 project "5G-TRANSFORMER: 5G Mobile Transport Platform for Verticals" (grant no. 761536).

REFERENCES

[1] C. I, J. Huang, R. Duan, C. Cui, J. . Jiang, and L. Li, "Recent progress on c-ran centralization and cloudification," *IEEE Access*, vol. 2, pp. 1030–1039, 2014.

[2] Mobile, China, "C-RAN: the road towards green RAN," *White Paper, ver*, vol. 2, 2011.

[3] A. de la Oliva and J. A. Hernandez and D. Larrabeiti and A. Azcorra, "An overview of the CPRI specification and its application to C-RAN-based LTE scenarios," *IEEE Communications Magazine*, vol. 54, no. 2, pp. 152–159, February 2016.

[4] C. P. R. I. eCPRI Interface Specification, "eCPRI specification v1.0," August 2017.

[5] 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, Oct 2015.

[6] 3GPP RAN3, "TR 38.801 V14.0.0," Mar 2017, Available at: http://www.3gpp.org/ftp/Specs/archive/38_series/38.801/.

[7] "5g deployment considerations, ericsson white paper, 2018." [Online]. Available: https://www.ericsson.com/4a5daa/assets/local/networks/documents/5g-deployment-considerations.pdf

[8] 3GPP, "TS38470, F1 general aspects and principles," July 2019.

[9] X. Costa-Perez, A. Garcia-Saavedra, X. Li, T. Deiss, A. de la Oliva, A. di Giglio, P. Iovanna, and A. Moored, "5g-crosshaul: An sdn/nfv integrated fronthaul/backhaul transport network architecture," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 38–45, February 2017.

[10] A. Garcia-Saavedra, J. X. Salvat, X. Li, and X. Costa-Perez, "Wizhaul: On the centralization degree of cloud ran next generation fronthaul," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2018.

[11] A. Garcia-Saavedra and G. Iosifidis and X. Costa-Perez and D.J. Leith, "FluidRAN: Optimized vRAN/MEC Orchestration," *IEEE INFOCOM 2018*, April 2018.

[12] A. Garcia-Saavedra, G. Iosifidis, X. Costa-Perez, and D. J. Leith, "Joint optimization of edge computing architectures and radio access networks," *IEEE Journal on Selected Areas in Communications*, 2018.

[13] A. de la Oliva, J. A. Hernández, D. Larrabeiti, and A. Azcorra, "An overview of the CPRI specification and its application to C-RAN-based LTE scenarios," *IEEE Communications Magazine*, vol. 54, no. 2, pp. 152–159, 2016.

[14] E. AB, H. T. C. Ltd, N. Corporation, A. Lucent, and N. Networks, "Common Public Radio Interface (CPRI); Interface Specification v7.0," 2015.

[15] S. C. Virtualization, "Functional splits and use cases," in *Small Cell Forum release*, vol. 6, 2016.

[16] Small Cell Forum, "Document 082.09.05, FAPI and nFAPI specifications, Release 10," May 2017, Available at: https://scf.io/en/documents/082_-_nFAPI_and_FAPI_specifications.php.

[17] I. . W. Group, "Next generation fronthaul interface," Available at: https://standards.ieee.org/develop/wg/NGFI.html.

[18] ——, "Time-sensitive networking (tsn)," Available at.

[19] IETF, "Deterministic networking (detnet)," Available at: https://datatracker.ietf.org/wg/detnet/about/.

[20] L. A. Wolsey and G. L. Nemhauser, *Integer and combinatorial optimization*. John Wiley & Sons, 2014.

[21] "Ieee approved draft standard for local and metropolitan area networks–bridges and bridged networks," *IEEE P802.1Q-REV/D2.2, January 2018 (Draft Revision of IEEE 802.1Q-2014, incorporating IEEE Qcd-2015, IEEE 802.1Qca-2015, IEEE 802.1Q-2014/Cor1-2015 IEEE 802.1Qbv-2015, IEEE 802.1 Qbu-2016, IEEE 802.1Qbz-2016 IEEE 802.1Qci-2017, IEEE 802,1Qch-2017)*, pp. 1–2000, Jan 2018.

[22] Open Networking Foundation (ONF), *OpenFlow Table Type Patterns, Version 1.0*, August 2014.

[23] N. Molner, S. González, T. Deiß, and A. De la Oliva, "The 5g-crosshaul packet forwarding element pipeline: measurements and analysis," in *Cloud Technologies and Energy Efficiency in Mobile Communication Networks (CLEEN), 2017 Fifth International Workshop on*. IEEE, 2017, pp. 1–6.

[24] P. Emmerich, S. Gallenmüller, D. Raumer, F. Wohlfart, and G. Carle, "Moongen: A scriptable high-speed packet generator," in *Proceedings of the 2015 Internet Measurement Conference*. ACM, 2015, pp. 275–287.

[25] L. Lu, X. Zhang, R. Funada, C. S. Sum, and H. Harada, "Selection of modulation and coding schemes of single carrier phy for 802.11ad multi-gigabit mmwave wlan systems," in *2011 IEEE Symposium on Computers and Communications (ISCC)*, June 2011, pp. 348–352.

[26] C. Chang, N. Nikaein, and T. Spyropoulos, "Impact of packetization and scheduling on c-ran fronthaul performance," in *2016 IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–7.

[27] 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.

[28] L. M. P. Larsen, A. Checko, and H. L. Christiansen, "A survey of the functional splits proposed for 5g mobile crosshaul networks," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2018.