Contenets available at SciVerse ScienceDirect

Computer Networks

Journal homepage: www.elsevier.com/locate/aebj

Latency Performance Analysis of Low Layers Function Split for URLLC Applications in 5G Networks

Yahya Alfadhli[a,*], You-Wei Chen[a], Siming Liu[a], Shuyi Shen[a], Shuang Yao[a], Daniel Guidotti[a], Sufian Mitani[b] and Gee-Kung Chang[a]

[a]Georgia Institute of Technology, School of Electrical and Computer Engineering, Atlanta GA 30332, USA.
[b] Telekom Malaysia R&D, 63000 Cyberjaya, Selangor, Malaysia.

ARTICLE INFO

Article history:
Received 00 December 00
Received in revised form 00 January 00
Accepted 00 February 00

Keywords:
Flexible functional split
URLLC
C-RAN
Option-9
Mobile fronthaul
Fiber-Wireless networking.

ABSTRACT

In 4G cloud radio access networks (C-RAN), latency at the mobile fronthaul was conceived mainly as a limiting factor of the fronthaul length. As long as fronthaul latency is not leading to violation of the hybrid automatic repeat request (HARQ) limit, the fronthaul latency would not have impact on the end-to-end latency. However, the advent of ultra-reliable low latency communications (URLLC) portends the need for effective latency reduction mechanisms, such as mini-slot and non-slot-based transmission, to enable URLLC data to be transmitted instantly. This implies that any delay at the fronthaul will impact the end-to-end round trip time (RTT) latency of URLLC applications and therefore should be carefully evaluated. On the other hand, consolidating more functions at the edge node, also known as remote unit (RU), has recently gained traction as a viable solution to reduce latency in 5G networks, with the sacrifice of increased complexity and capital/operational expenses (CAPEX/OPEX). In this paper, we provide experimental quantitative latency analysis of different low function split options at the fronthaul for URLLC applications using commercial off-the-shelf equipment (COTS). We also demonstrate that having the simplest remote unit design, which only contains an analog optical-to-electrical convertor, denoted by Option-9, can achieve the lowest fronthaul latency. Based on our findings, we design a flexible 5G architecture that can efficiently support different 5G applications.

1. Introduction

It is indisputable that in order for 5G systems to reduce the latency by ten folds, a set of drastic developments in different fields need to be jointly and severally achieved. Latency reduction must occur in the radio access network (RAN), core network, mobile edge computing, digital signal processing (DSP) algorithms and wireless/optical integration [1]. The target end-to-end RTT is specified to be in one to a few milliseconds range to facilitate the realization of a various types of new applications, that can have a significant impact on our lives, such as tactile internet, virtual reality, augmented reality, tele-medicine, tele-surgery, smart transportation, and autonomous vehicles. For example, an application such as real-time control for discrete automation has a maximum allowed latency of 1 ms for the end-to-end RTT, which is measured at the application layer of the user equipment [2].

In this low latency regime, few tens of microseconds are considered invaluable resource that justifies sacrifices in different aspects such as design complexity and transmission efficiency. The recent release of new radio (NR) 3rd Generation Partnership Project (3GPP) standard, release-15, introduces the concept of mini-slot to support the URLLC applications by reducing the transmission time interval [3]. The standard defines

* *Corresponding author.* Tel.: +1-404-980-3153.
E-mail addresses: yalfadhli@gatech.edu (Y. Alfadhli), yu-wei.chen@ece.gatech.edu (Y.-W. Chen), liu567c@163.com (S. Liu), sshen60@gatech.edu (S. Shen), syao65@gatech.edu (S. Yao), daniel.guidotti@ece.gatech.edu (D. Guidotti), sufian@tmrnd.com.my (S. Mitani), and geekung.chang@ece.gatech.edu (G.-K. Chang).

http://dx.doi.org/10.1016/j.aebj.2014.10.012

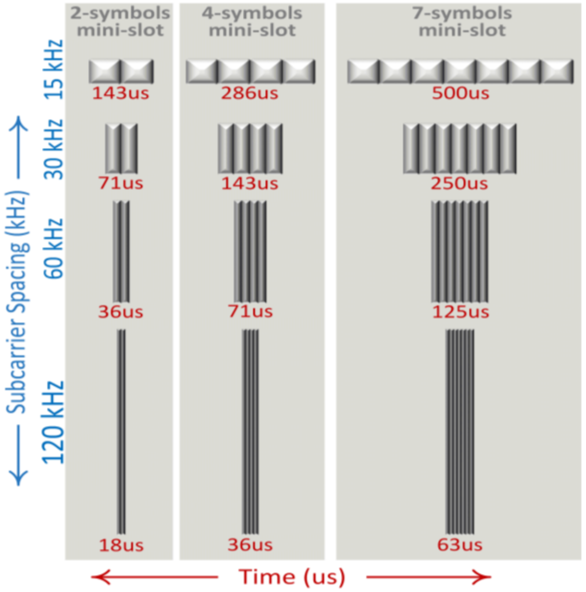


Fig. 1 – Defined mini-slots in 3GPP standard

three different sizes of mini-slot: 2-symbols, 4-symbols and 7-symbols, as summarized in Fig. 1. In order to further reduce the latency, these mini-slots will be transmitted instantly upon creation by puncturing the originally allocated resources for other types of applications. In the case that the coding is not robust enough to recover the interrupted data, then the punctured data should be scheduled for retransmission and some considerations need to be taken at the user and base station sides to be able to decode the data accurately. Moreover, the situating of the mini-slot should not be limited to the boundary of a slot. This enables the URLLC data to be transmitted instantly without the need to wait for the next slot boundary. Assuming that the URLLC data has just missed the slot boundary, this non-slot-based transmission can save a maximum of 125 microsecond (us) in case of 120 kHz subcarrier spacing and a maximum of 500 us in 30 kHz subcarrier spacing case, for example. This latency saving, however, comes at the cost of design complexity and transmission efficiency. Moreover, this dynamic mini-slot transmission mechanism makes the URLLC application sensitive to any additional latencies at the fronthaul as will be addressed in Section 6.

On the other hand, the adoption of portions of the millimeter wave (mm-wave) spectrum in 5G implies that there will be massive deployment of small cells. This is due to the fact that mm-waves are suitable for short transmission ranges while the conventional eNB stations, using lower radio frequencies, are less densely distributed and will not be sufficient to provide 5G required capacity and coverage. Thus, the cost of these new small cells' base stations should be reduced as much as possible and one effective way to achieve this goal is by assigning simple functions to these base stations. Another benefit of using the mm-waves is that much higher throughput can now be supported owing to the higher data carrying capacity of mm-wave channels. This raises the fronthaul link bandwidth requirement beyond what current technologies, such Common Public Radio Interface (CPRI), can provide, mandating the need for new fronthaul transport technologies [4], [5]. Therefore, recent studies of 5G architectures aim to design a network with low cost base stations, low fronthaul latency and high bandwidth capabilities to support various 5G applications.

A very promising method to achieve these design goals is by using the concept of function split that is defined by 3GPP standard [6], and it has been thoroughly surveyed in [7]. The standard defines 8 general function split options and it also provides some analytical comparison between them. The concept of function split is also known as RAN functional decomposition [8]. Figure 2 illustrates all the functions that are performed by a conventional eNB along with the 8 splitting points defined by the standard [6]. Option-7 and lower function split options are referred to as low function split options. Moreover, we add one extra split option denoted by "Option-9" for it is a very promising function split as will be discussed in the following sections.

Most general 5G deployments, however, propose the adoption of two cascaded function splits [9]–[11]. One function split at the higher network layers, collectively referred to as midhaul, will enable the function integration of the 4G and 5G networks. The most popular function split candidate for the midhaul is Option-2 that separates central unit (CU) functions from those of the distributed unit (DU). A second function split, which is a fronthaul split, lies between the DU and the "edge node". The edge node has been assigned different names by different mobile-networks related forums such as remote radio unit (RRU), eCPRI radio equipment (eRE), physical network functions (PNF), remote radio head (RRH) and radio unit (RU) [8]. All these names reflect that the RF layer is placed at the edge node. However, in our case, we will use the term "remote unit (RU)" since the RF layer is located at the DU. This terminology is also used by the 5G-XHaul project [9], [12].

Aside from the original motivation behind the movement towards function splits, it turns out that different function split options have different performance capabilities and can significantly facilitate or favor some applications [8]. Generally, the most accepted candidate function split option for fronthaul is Option-7, which can reduce the fronthaul bandwidth requirement and keep the RU quite simple. Higher function split options can be used to facilitate lower latency applications by placing more functions at the RU. The idea of moving some of the functions of the core network to the access network is also proposed to help to achieve the 1 ms end-to-end delay requirement [11].

However, high function split options at the fronthaul, such as Option-2, will not be practical because the complexity of the RUs will be higher, and the statistical multiplexing gain is reduced. As 5G deployment is based on small-cell architecture and because each of the Option-2 RUs works relatively as a full base station, there will be a need for frequent handovers. This will degrade the latency performance and greatly increase the control messaging to handle the frequent handovers procedures. Moreover, the placement of MAC layer at the RU will result in the loss of some functionalities related to centralized scheduling such as

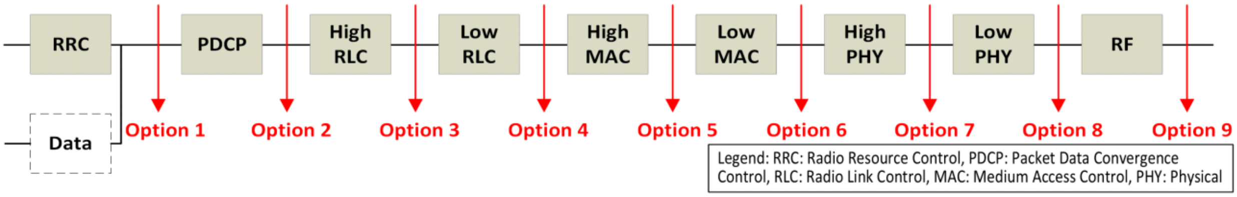

RRC
PDCP
High RLC
Low RLC
High MAC
Low MAC
High PHY
Low PHY
RF
Data
Option 1
Option 2
Option 3
Option 4
Option 5
Option 6
Option 7
Option 8
Option 9
Legend: RRC: Radio Resource Control, PDCP: Packet Data Convergence Control, RLC: Radio Link Control, MAC: Medium Access Control, PHY: Physical

Fig. 2 - Function split options.

interference management and coordinated transmission in multiple cells. Compromising these functionalities degrades the communication reliability which impacts the feasibility of URLLC applications, as increasing reliability is one of the URLLC major targets. On the other hand, centralizing the MAC layer at the DU can improve interoperability, system scalability, inter-cell coordination and mobility in ultra-dense small cells deployments [6]. Thus, in most scenarios and applications, function split at the fronthaul is placed at the MAC or lower layers, which contradicts with the low latency requirements.

In a related work [13], the authors provide a detailed analysis of the impact of function split options on fronthaul bandwidth requirements and maximum number of supported RRH. They also study different packetizing methods such as sub-frame based and symbol-based packets. They show that symbol-based packets can reduce latency but on the other hand, it limits the number of supported RRH as it decreases the transmission efficiency. Paper [14] presents a cost analysis of some of the PHY layer function split options. Another interesting work studies the impact of packetizing CPRI traffic over the Ethernet, for example, in wide area networks (WAN), and experimentally verifies if it can meet the latency and jitter requirements [15], [16]. Authors of [15] have proposed a scheduling policy in an effort to reduce jitter.

On the other hand, there are several works that offer an in-depth analysis of the impact that virtualization has on latency and further test the performance under different virtualization environments e.g., kernel-based virtual machine (KVM) and VirtualBox [17]–[20]. Lastly, [21] presents an interesting fine-grained flexible function split virtualization to enable flexible placement of functions in the network. The paper also shows the impact of different function splits in 4G networks on the fronthaul latency under different fronthaul traffic conditions using mininet emulation platform.

In this paper, we provide a detailed experimental analysis of the impact of a low fronthaul function split choice particularly on latency performance for 5G-URLLC applications, using commercial off-the-shelf equipment. Latency evaluation presented in this paper includes three major parts: (*i*) PHY processing, (*ii*) fronthaul interface, and (*iii*) optical transceiver delays. Then, we evaluate the latency requirements of URLLC applications to estimate the available roundtrip fronthaul latency budget and the supported fronthaul link lengths under Option-9 function split. Finally, based on our analysis, we propose a flexible architecture that can greatly benefit low latency and other 5G applications.

The rest of this paper is organized as follows. First, we discuss the most promising low function split options for the fronthaul. In Section 3, we describe our experimental setup to measure different parts of the delay. In the following section, we conduct the latency measurements pertaining to PHY processing and interface delays at both the DU and RU. Then, in Section 5, we present the experimental setup and results analysis of the latency generated from the optical transceiver including analog and digital transponders. After that, in Section 6, we further analyze the results from the two previous sections to arrive at a comprehensive conclusion of the effect of functional splits on fronthaul latency and how they can impact the performance of URLLC applications. Finally, we present our proposed flexible function split architecture that can facilitate achieving the target end-to-end latency.

2. Candidate Function Split Options for the Fronthaul

At the fronthaul level, RUs can be classified or named based on their function split. For example, the simplest RU that only has an optical-to-electrical convertor, an electrical amplifier and an antenna will be denoted as Option-9-RU. Whereas a more complex RU would have the whole PHY layer and can be denoted as option-6-RU. Including layers higher than PHY at the RU will increase the RU complexity, cost, and reduce the benefits of statistical sharing. In general, simple RUs, such as Option-8-RUs, have many advantages over complex RUs such as lower units' cost and power consumption, higher statistical sharing, and better small cells coordination by enabling functions such as: coordinated multiple point (CoMP), multiple input multiple output (MIMO), etc.

One disadvantage of Option-8-RUs is that they require higher bandwidth fronthaul links. However, this requirement can be alleviated by moving the FFT and CP processing to the RUs, known as Option-7-RU, depicted in Fig. 3, which can reduce the bandwidth requirement of Option-8 by about 33% and keep the RU reasonably simple at the same time [6]. This enables Option-7-RUs to be more suitable to support bandwidth hungry applications. All these advantages make simple RUs more able to support most types of applications except for those that require low latency.

Low latency applications, in general, call for more complex RUs. However, there are several disadvantages related to complex RUs, which are basically the opposite of the simple RUs such as higher cost and system complexity. Theoretically, the RU can be as complex as Option-2 where all the processing is done at the edge node near to the user. While this option sounds efficient in terms of lowering the latency, increasing reliability and reducing the fronthaul bandwidth requirements, it is considered to be cost prohibitive. Moreover, in practice, such extremely distributed scheme is not practical since the centralization gain and centralized scheduling benefits are minimal. Hence, the choice of simple RUs is generally preferred.

The major concern with simple RUs is how they can support low latency users. In order to overcome this issue, we propose using analog radio over fiber (RoF) technology to move the RF layer to the DU. Even though RoF technology is known for its suitability for distributed antenna systems (DAS) technology, there are several standardization efforts, in the International Telecommunication Union (ITU-T) Group 15, led by advance research institutes promoting RoF technology for optical access systems [22]–[25]. In RoF schemes, the RF layer basically contains the analog to digital convertors (A/D) and digital to analog convertors (D/A) for RF signal transmission and reception. By consolidating these RF convertors in the DU, they will be shared among all RUs, which reduces the implementation cost. Another advantage of shared RF layer is to eliminate the need for synchronization between RUs, which greatly simplifies the RU design [26]. This split can be called Option-9-RU or Analog-Option-8-RU.

Another version of RoF is described in [27] where authors split the RF layer into high and low RF layers using delta-sigma modulation method. In [27], the RF layer is greatly modified where the D/A is replaced by up-conversion in the digital domain then the signal is modulated using delta-sigma modulator. Then, at the RU, a band pass filter is used as the Low-RF layer. Using delta-sigma based RU can reduce the fronthaul bandwidth requirements compared to Option-8, yet Option-7 is still more bandwidth efficient. However, in this paper, we will use the term Option-9 to refer to the case where the whole RF layer is placed at the DU.

There are several limitations imposed by the use of Option-9, among which is the need for point-to-point connections at the fronthaul. However, to reduce the cost of fiber deployment, optical network with shared feeder fiber, in the form of star topology, can be realized by employing some multiplexing techniques such as wavelength division multiplexing (WDM) or frequency division multiplexing (FDM). The adoption of star topology for 5G fronthauling is recommended by different technical communities including ITU-T (GSTR-TN5G) [28]. Moreover, star topology is also used by several existing access networks technologies such as gigabit passive optical networks (GPON), hybrid fiber coaxial network (HFC) and CPRI-based C-RAN. This makes the integration of any of these technologies with RoF scheme an economic design choice. Reference [29] discusses some possible RoF configurations and characteristics over optical distribution networks (ODN).

Another drawback is the distance limitation due to non-linearity impairments. However, even though this factor was of high priority under 4G cloud radio access network (C-RAN) scenario, distance limitation becomes less significant since the expected length of 5G fronthaul is less than 20 km [30]. In fact, based on statistical analysis results reported in [31], 82% of deployed fronthaul links are less than 10 km long while 95% are less than 15 km for urban areas. Moreover, there are several recent advances in the mitigation of non-linearity impairments such as the use of novel fiber-wireless integration methods and machine learning algorithms. Reference [32] experimentally demonstrates a transmission of 60-GHz radio frequency signals over a 200-km length of optical fiber by using a novel dual-stage optical and electrical filtering. On the other hand, the use of machine learning algorithms such as Artifactual Neural Networks (ANN) is experimentally demonstrated in [33] to mitigate the non-linear interference in a 15 km RoF-based fronthaul link.

All in all, the reduction in the fronthaul length requirement to below than 10 km as well as the recognition of RoF technology as a viable solution for optical access networks have opened the doors for the RoF technology to mobile fronthaul application territory. From another angle, RoF technology promises to bring several intriguing benefits such as the simplicity of the RU design and the low latency performance. With the latency being a very stringent requirement, 50 us fronthaul RTT as defined by recent eCPRI specifications, a handful of microseconds is becoming a precious resource [34]. Considering this new fronthaul latency threshold, the maximum fronthaul length for URLLC applications is limited to 5 km assuming that the fronthaul latency is equivalent to the fiber propagation latency only. Therefore, in the following four sections, we are experimentally analyzing the latency performance of RoF technology and comparing it with other fronthaul function split options.

3. Experimental Setup for Evaluating Function Split Options in the Mobile Fronthaul

Figure 3 demonstrates three major latency components under study. First is the processing latency, discussed in Section 4, which is the time used for PHY processing. Second latency component, addressed in the same section, is the interface delay, which is the time consumed to prepare the data for transmission and the time needed for acquiring a reasonable amount of data for processing including (de)-packetizing and (de)-compression operations. Third component is the optical transceiver latency generated from electrical to optical conversion stage and it is discussed in Section 5. Then, in Section 6, all the results from the previous two sections are collectively analyzed.

In order to experimentally analyze the performance of different function split options in terms of latency, we make use of the Open Air Interface (OAI), which is an open source implementation of the long term evolution (LTE) standards [35]. The source code is written mostly in C and built on low latency Linux kernels to ensure real time functionality. Most of the processing is done on general commodity computers and a compatible software defined radio (SDR) platform is used as an RF interface to facilitate A/D and D/A operations and RF signal transmission and reception. This feature makes the OAI a very

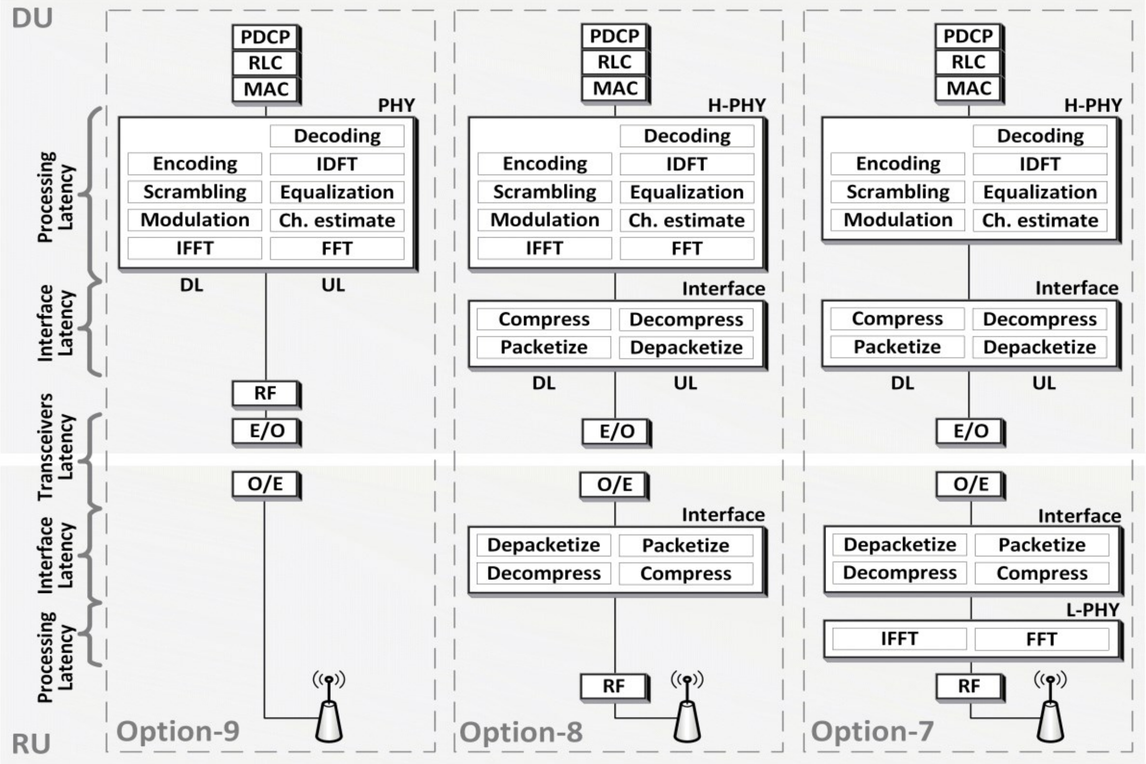


Fig. 3 - Experimental setup for measuring processing delay and interface delay at the DU and RU for Option-9, Option-8 and Option-7.

efficient platform with which to evaluate the different function split implementations. However, there are several assumptions need to be considered.

First, our major objective is to compare different low function split options for the URLLC applications in 5G. Therefore, in our measurements, we don't consider the end-to-end roundtrip measurements in order to isolate latencies specific to LTE such as HARQ and transmission time interval (TTI). We further split the one-way trip measurement into small sections pertaining to the parts impacted by the change of function split. Hence, even though the used OAI platform is following the LTE standards, the results of the following experiments provide an insight on the latency performance of different function split options for URLLC applications. Second, measuring the processing latency of the PHY layer, in general, is impacted by several factors such as the use of general-purpose processors (GPP) opposed to dedicated hardware, the CPU frequency, the used virtual environment [17]–[20], etc. Therefore, we measure the PHY processing latency just to demonstrate if the function split options will impact the PHY processing latency while we primarily focus on measuring latencies induced by the integration of the fronthaul to mobile networks.

The experimental setup for the candidate functions split options is shown in Fig. 3. To begin with, three different function split options are implemented: Option-7, Option-8 and Option-9 as indicated by the dashed boxes. In Option-7, the CP removal and FFT processing are done at the RU for the uplink case, while IFFT and CP addition are done in the RU for the downlink case. This variation of Option-7 is known as "Option 7-1" in the 3GPP standard [6]. Before transmission, the frequency domain IQ samples are compressed using A-law compression algorithm [36], and then packetized and converted to the optical domain using a commercial electrical to optical convertor (E/O) manufactured by (10Gtek). The convertor uses distributed feedback laser working at 1550 nm wavelength with output power of 1 dBm. In Option-8, the FFT and CP processing are performed in the DU while the RF processing is done in the RU. The time domain IQ samples are compressed and packetized before conversion to the optical domain. The third function split option is Option- 9 where the time domain IQ samples are fed into analog to digital convertor before being converted into optical domain using a commercial analog optical transceiver manufactured by Zonu (OZ600). The optical transceiver works at 1550 nm and at power level of 5 dBm. A list of the hardware used in the experiment is listed in Table 1.

Table 1 - List of testbed hardware

Component	Specifications
EPC	Intel(R) Core(TM), Quad CPU @ 2.40GHz
CU+DU	Intel(R) Core(TM), i7-4790 CPU @ 3.60GHz
RU	Intel(R) Core(TM), i7-4770 CPU @ 3.40GHz
RF interface	NI USRP-b210

EPC: evolved packet core network.

To evaluate the latency of different function splitting options, common layers, (e.g., RLC, MAC and RF) are excluded. Measuring PHY processing and interface delays is different for DU and RU. DU measurement is comprised of two parts as shown in Fig. 3: the PHY processing and interface delays measurements. For Option-7 and Option-8, the PHY processing delay for the downlink is measured from the beginning of the PHY layer to just before the beginning of transmission. Then, the interface delay is measured from this point to the point where the packet is sent by the Ethernet interface. For the uplink measurements, a time stamp is taken once a packet is received and another time stamp is taken after the de-packetizing and decompression are completed so as to obtain a measure of the uplink interface delay. Then, another time stamp is taken after the encoding to measure the uplink PHY processing delay. For Option-9, we use the same points we used for Option-7 and Option-8. The only difference is that for Option-9, neither compression nor packetizing is performed. So, for the downlink, the measurement stops just after the IFFT is performed and the time domain IQ samples are ready to be fed into the Universal Software Radio Peripheral (USRP) for transmission. The delay of the RF layer is excluded from the measurement as it is a common layer for all three options under study. For the uplink, the delay starts just after receiving the samples from the USRP and stops at the same points as the other split options.

Likewise, the RU delay measurement contains the measurements of both PHY processing and interface delays. The interface delay includes reception/transmission of packets, (de)-packetizing and (de)-compression. The RU delay measurement, however, is different for different split options. For Option-9, there is no RU PHY processing nor interface delay since the RU, in this case, does not perform any processing, and it merely performs an optical to electrical conversion, which will be covered in the optical transceiver section. For Option-8, there is no PHY processing done in the RU and the generated delay comes from the interface only. So, for the downlink, the latency measurement starts from the beginning of reception of the Ethernet frame until the point where data is fed to the USRP for transmission. For the uplink, the measurement starts once the samples are received by the USRP until the point where the packet is sent by the Ethernet interface. However, for Option-7, the frequency domain IQ samples are extracted from the packets and go through the FFT/CP stage. Hence Option-7-RU measurement includes both PHY processing and interface delays. Thus, we use the same measurement points as Option-8 with the exception that we add one measuring point at the FFT/IFFT stage to capture the PHY processing delay as well.

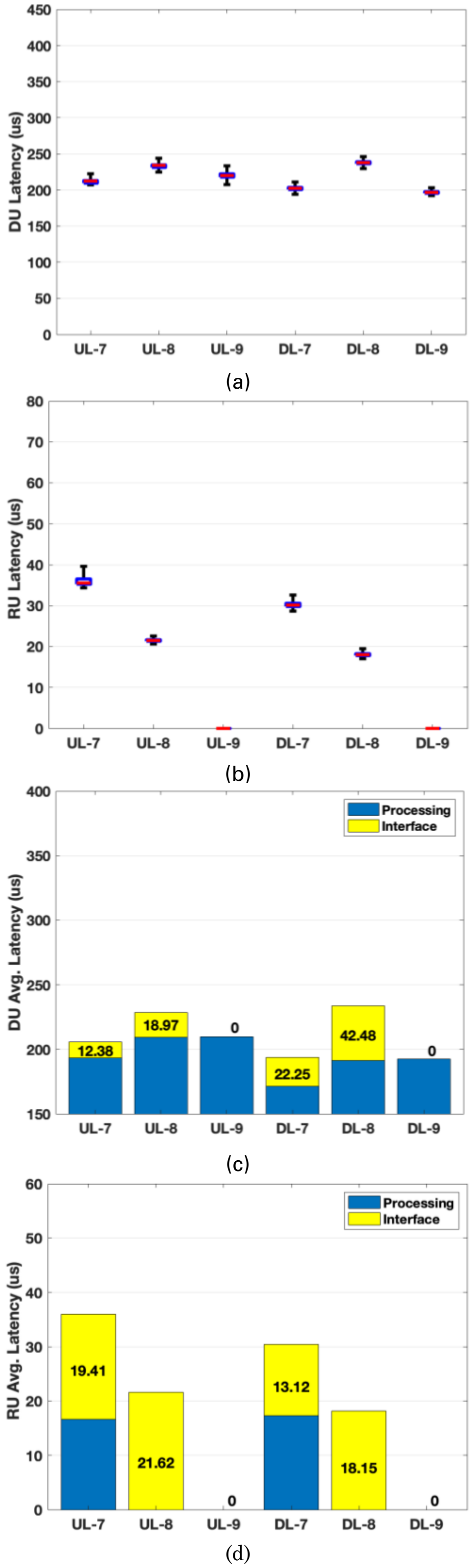
4. Experimental Results and Analysis: PHY Processing and Interface Latencies

Figure 4(a) illustrates the overall delay at the DU for different function splits for both uplink and downlink in the case of 5 MHz wireless signal. The results reveal that Option-8-DU has the highest overall latency because it performs all the processing including H-PHY, L-PHY and fronthaul interface. Compared to Option-8-DU, Option-9-DU has lower latency because it does not include the interface latency. On the other hand, Option-7-DU has the H-PHY, as other options, but instead of performing the L-PHY processing, it includes the interface latency. As the latency of L-PHY processing is equivalent to the interface latency in the case of 5 MHz wireless signal scenario, the DUs of both Option-7 and Option-9 have relatively similar latencies.

The overall RU latency is shown in Fig. 4(b) and it is noticeable that Option-7-RUs have the highest overall latency. Option-9-RU, on the other hand, has zero overall latency since there is no delay coming from the interface and all the processing is done at the DU. Figures 4(c) and 4(d) illustrate the breakdown of the latency including the processing and interface delays components for the DU and RU, respectively. These two figures show that the interface latency is dependent on the choice of the function split option, where Option-7 has lower interface latency compared to Option-8 as it has lower fronthaul bandwidth.

The experiment is repeated using 10 MHz LTE signal and the total latencies for both DUs and RUs are shown in Figs. 5(a) and

5(b), respectively. Using higher RF channel bandwidth increases the total latency for all function split options. Moreover, Figs. 5(c) and 5(d) indicate that the interface delays at both DUs and RUs are also higher, compared to Figs. 4(c) and 4(d). In other words, the interface latency is a function of the wireless bandwidth, for Option-7 and Option-8 only, while it remains zero for Option-9.



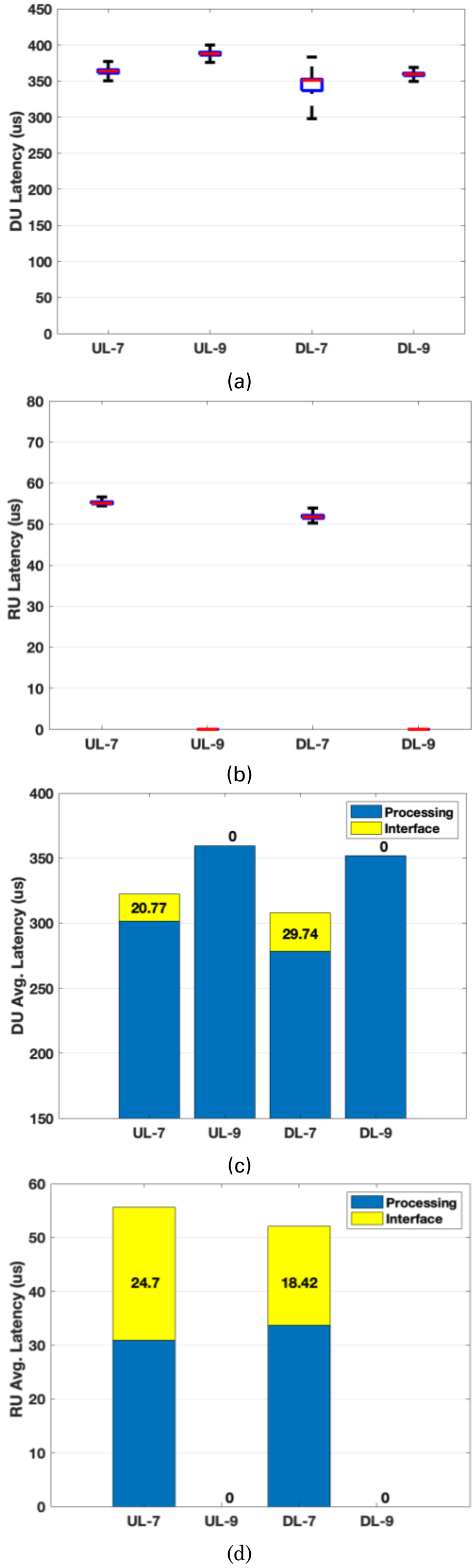
(a)

(b)

(c)

(d)

Fig. 4 - PHY processing and interface delays for 5 MHz wireless signal for different function split options. Total delay at: (a)DU and (b)RU. Breakdown of average delay at: (c)DU (d)RU.



(a)

(b)

(c)

(d)

Fig. 5 - PHY processing and interface delays for 10 MHz wireless signal for different function split options. Total delay at: (a)DU and (b)RU. Breakdown of average delay at: (c)DU (d)RU.

Even though the interface latency is in tens of microseconds in our experiment, its impact is expected to be more significant in case of mm-wave bandwidth capabilities. Increasing the wireless signal bandwidth means increasing of the number of resource blocks transmitted by RUs. This consequently will increase the fronthaul data traffic, which will increase the interface delays at the DU and RU, denoted by $\mathrm{I}_{\mathrm{DU,RU}}$. These interface delays comprise of (de)-packetizing delay "$t_{(De)-packetizing}$" and (de)-compression delay "$t_{(De)-compression}$" and can be written as:

$$I_{DU,RRU} = t_{(De)-packetizing} + t_{(De)-compression} \qquad (1)$$

A good indication of the impact of the wireless signal bandwidth increment on the latency can be obtained by comparing Option-7 results in Fig. 4(d) with Fig. 5(d), which show the latencies of L-PHY and interface processing. We can see that the L-PHY processing latency increased about 66% while the interface latency increased about 27% only. As the L-PHY layer resides in the RU in case of Option-7, the total latency at the Option-7-DU is lower than Option-9-DU. However, when considering all latencies in the DU, RU and fronthaul, we can see that Option-9 has the lowest latency as will be discussed further in Section 6.

Even though the results for 10 MHz Option-8 fronthaul are not reported, due to system limitations, one can expect that the interface latency for 10 MHz Option-8 will be higher than that for 10 MHz Option-7. As a summary, the interface latency for Option-7 and Option-8 is dependent on the used fronthaul bandwidth, which is a function of both: the function split choice and the used wireless signal bandwidth.

5. Optical Transceiver Delay Analysis

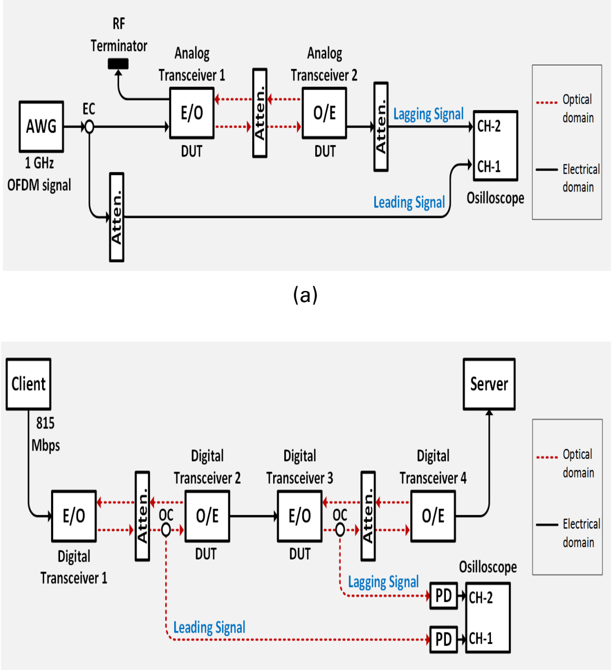
The transceiver delay captures the delay of the electrical to optical convertor pairs for all function split options as shown in Fig. 3. The propagation delay coming from the fiber is not included in our measurements as the light wave propagation speed in fiber can be approximated by 2×10^8 m/s. Transceiver delay is different for analog and digital fronthauls. In the analog case, the optical convertor is composed of a photodiode for reception and a laser diode for transmission. Digital convertors or transponders, on the other hand, contain additional logic circuitry along with the photodiode and laser diode. This logic circuitry performs PHY layer processing and some data link functions, which introduces additional delay that can significantly increase the overall fronthaul latency. This experiment is conducted using the same analog and digital transceivers that are used for the experiment in the previous section. The experimental setup for measuring analog optical transceiver latency is shown in Fig. 6(a), while the setup for measuring the digital optical transceiver delay is depicted in Fig. 6(b).

In order to test the analog transceiver latency, an arbitrary wave generator (AWG) is used to produce 1 GHz bandwidth orthogonal frequency division multiplexing (OFDM) signal at a carrier frequency of 1 GHz. The signal from the AWG is split into two links at the electrical coupler (EC), one arm is connected to the oscilloscope channel-1 (CH-1) as a leading signal and the second is connected to the first analog optical transceiver (E/O). The output of the analog optical transceiver is connected using a short fiber to an optical attenuator (Atten.) for protection. Another equal length fiber is used to connect the output of the attenuator to the second analog transceiver. The output of the second transceiver is connected to the oscilloscope as the lagging signal at CH-2. Figure 7(a) shows the time difference between these two signals to be less than 30 nanosecond (ns), which is the delay of a pair of analog optical transceivers. Then, the latency is measured using *finddelay* function of MATLAB, which is consistent with the oscilloscope measurement.

The experimental setup for measuring the digital optical transceiver delay is shown in Fig. 6(b). To measure the latency of the digital transceivers, two computers are connected, one as a client and another as a server. To precisely measure the latency of the transceivers only, latency measurement applications such as *ping* cannot be used since these applications include the delay of some of higher Open Systems Interconnection (OSI) layers at both the client and server ends, namely PHY, MAC and Transport layers. Moreover, *ping* test results are affected by the computational load on the tested machine, which increases the results variations.

Two pairs of digital optical transceivers connected by a short Cat-5 cable are used as the oscilloscope has RF interface only. The first transceivers pair is connected at the client end and an optical coupler (OC) is used to split the signal. One arm of the OC goes to photodiode (PD) at the oscilloscope to provide the leading signal while the other is connected to the digital transceiver. An identical connection is made to the second pair of transceivers that feeds the lagging signal to the oscilloscope. In this way, we measure the delay due to digital optical transceiver-2 and transceiver-3, each labeled as devices under test (DUT).

Digital transceivers follow optical fiber Gigabit Ethernet standard and therefore, even if there is no data being transmitted, the two transceivers will continuously transmit some synchronization patterns for example, special coding K28.5 from 8b/10b line coding standard. This continuous stream makes the task of measuring the latency using the previous method, used for analog transceivers, quite difficult. It is also not possible to use a trigger signal from an external signal generator since these transceivers are working in real-time and need to be bidirectionally synchronized [37]. In order to measure the delay of such active devices, an oscilloscope with 8b/10b serial pattern



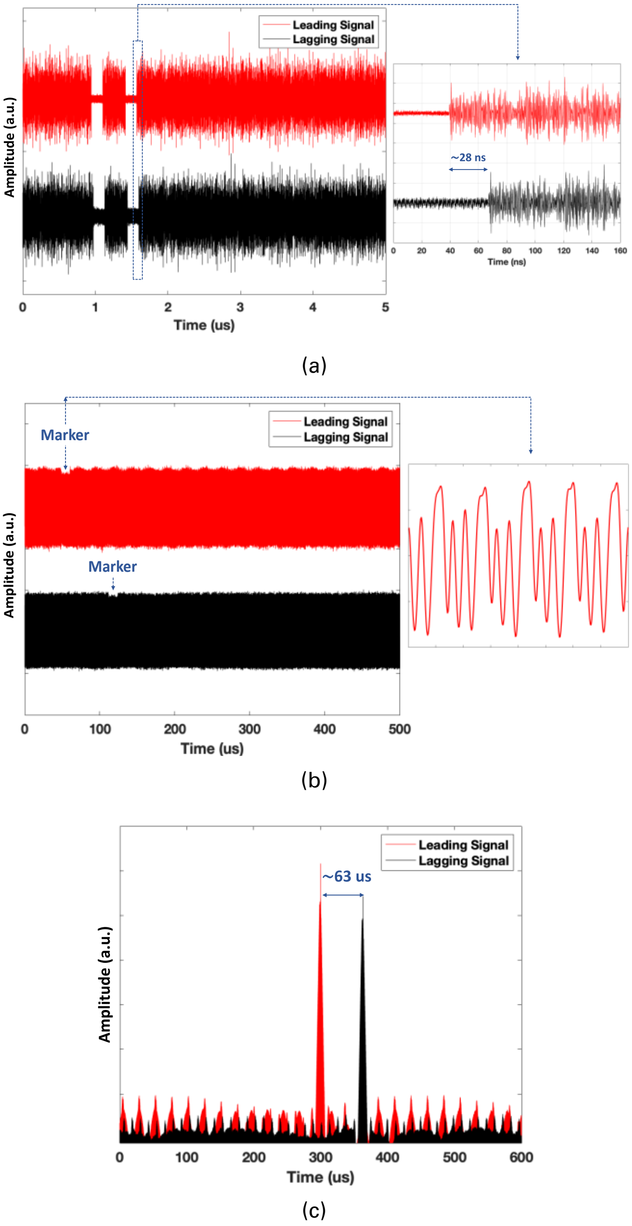
(a)

(b)

Fig. 6 - Experimental setup for measuring the optical transceiver latency for: (a) Option-9 and (b) Option-7 and Option-8.

triggering feature is needed. However, since such equipment is not available in our lab, we use an alternative method that enables us to accurately measure the delay using the available resources. We think it can be useful to describe our testing methodology in some details for a reference.

The major task is to generate a unique pattern, called a "Marker", inside the data stream to act as a timing reference. There are some conditions that the marker should meet. First, this marker should be of known duration and pattern to make the identification task easy. Second, the marker should be in a form that is acceptable to the DUT. If the marker is some random analog signal generated by an external source, the transceiver will generate an unexpected output. Basically, this external signal acts like noise and may cause loss of connection and synchronization between the transceivers. Third, the marker repetition period should be large enough to allow only one marker per acquisition window. Having more than a marker per acquisition window can lead to confusion in measuring the latency. Fourth, the acquisition window should be larger than the expected delay. The expected delay in our case is measured using the *ping* application. The *ping*, in its basic format, returns the RTT and includes some additional OSI layers, such as PHY, MAC and Transport layers. So, the expected delay of our devices should be less than a half of *ping* test result. Accordingly, a marker is inserted into the signal every 1 ms. This repetition period is chosen to be greater than the delay shown by a simple *ping* test to make sure that only the distance in time between any two consecutive markers is greater than any possible delay values attributable to other sources.



(a)

(b)

(c)

Fig. 7 - Optical transceiver latency. (a) Option-9 results. (b) The leading and lagging signals for Option-7 and Option-8 with subplot of the marker signal. (c) Option-7 and Option-8 result.

The marker is generated using the *Extended ping* application. A packet of 1500-bytes size is produced with a payload of zeros to be used as our marker. Since the packet size is known, we can calculate the marker duration using $t_{transmission} = packet\ size/link\ throughput$. In our case, the marker duration is about 12 us since the link throughput is around 1 Gbps. A portion of the time domain signal of this marker is shown in Fig. 7(b). The delay is measured both on the oscilloscope and in MATLAB and both methods yield equivalent results. To measure the digital transceiver latency using MATLAB, we first normalize both signals and perform a cross-correlation (X-correlation) between the signal and the marker. From the x-correlation results, we find the distance between the maximum correlation values. The test is performed when there is no traffic carried on the link and when the link is fully occupied by 815 Mbps data stream, generated using *iperf* application. The results in both traffic cases remain constant over time even though *ping* test yields continuously fluctuating results.

The results displayed in Fig. 7(c) demonstrate that the one-way trip delay corresponding to a digital transceiver pair is about 63 us. This means that, considering uplink and downlink, the digital optical transceiver introduces a net delay of about 130 us. This is to be compared with the delay of about 60 ns for the analog case. Considering that optical fiber propagation is about 5 us/km, the digital optical transceiver delay of 130 us is equivalent to a 25 km long fiber propagation delay. This result is consistent with measurements in [38], wherein authors reported that for a 10-Gbps operator-grade optical transponder, the delay is about 70 us per card.

6. Analysis of Total Fronthaul Latency

In this section, we first summarize the results from the previous two sections to present a complete picture of the latency at the fronthaul. From Figs. 4 and 5, we can see that placing of functions on different network units doesn't change the overall PHY processing time. For example, the FFT stage for 5 MHz signal consumes about 18 us regardless if it is performed in the DU or RU. This might not be always the case since in a real implementation, it is safe to assume that the DU has more computational power than the low-cost RU. However, we assume here that both machines have similar computational power. Therefore, the total PHY processing time of the RU and DU should be the same for all function split options. Accordingly, one-way trip delay of the fronthaul for either uplink or downlink "$t_{FH_{UL,DL}}$" can be expressed as:

$$t_{FH_{UL,DL}} = I_{DU} + t_{transceiver} + t_{propagation} + t_{switching} + I_{RRU} \quad (2)$$

Where I_{DU} and I_{RRU} are the interface delays described in Eq. 1. On the other hand, $t_{propagation}$ is the time taken by the signal to travel through the fronthaul link, which is independent from the choice of function split option but only depends on distance and transmission medium. The time taken by packets to be routed through a switch if a switch or a router was used, as would be the case for the digital function split options, is denoted by $t_{\text{switching}}$. This delay can vary depending on the number of connections, fronthaul traffic, etc. Lastly, $t_{\text{transceiver}}$ is the delay generated by the optical transceivers and has been discussed in the previous section. This delay can be further broken down into conversion delay and transmission delay as in Eq. 3:

$$t_{\text{transceiver}} = t_{\text{conversion}} + t_{\text{transmission}} \quad (3)$$

Where $t_{conversion}$ is the time taken by optical convertors (E/O and O/E) and $t_{\text{transmission}}$ is the time taken to transmit a packet. It is worth mentioning that $t_{\text{transmission}}$ is dependent on packet size and link throughput, which means that shorter packets are faster and save some transmission delay [39]. However, reducing the packet size can congest the network faster, resulting in reduction in the number of supported RUs per link due to the resulting increase in the overhead to data ratio [13]. Another aspect to consider when reducing the packet size is that it can increase the interface delay at the DU and RU since these units will now have to handle a greater number of packets. Moreover, the rate of packet loss will increase, which can increase the end-to-end latency due to retransmission mechanism.

For 4G networks, the maximum allowed one-way trip delay in the fronthaul, t_{FH}, is about 250 us as per recommendation in [6], [40], which implies that $t_{FH_{UL}} + t_{FH_{DL}} \leq 500\ us$. It is important to mention that this number is calculated based on LTE numerology and it is not specific to a certain application category [6]. Hence, we add up the uplink and downlink measurements from Figs. 4(c) and 4(d) and normalize all the results to the 500 us threshold value for each function split option. The results are summarized as percentages of 500 us in Table 2 with the assumption that the packet size is 1500 bytes and the wireless channel bandwidth is 5 MHz.

Table 2 summarizes the available fronthaul time budget after considering the interface and optical transceiver latencies, denoted as t_{budget}. This t_{budget} can be spent on other latency components such as $t_{switching}$ and $t_{propagation}$. This implies that, an Option-9 fronthaul range can be extended by roughly a factor of two with approximately the same latency when comparing with an otherwise equivalent link in Option-7. In other words, using the same fiber length, Option-9 can save more than 190 us of the fronthaul latency even when switching time is not included. However, it is important to mention that these numbers can change based on various variables including hardware, software, DSP efficiency, etc. Our goal is just to provide an overall generic analysis without loss of generality. These results are especially useful in that they point out principal sources of delay in the fronthaul.

Nonetheless, even with some system improvements in the foreseeable future, we can safely assume that considerations analogous to those discussed here will continue to arrive at similar conclusions. For example, the total interface delay $I_{DU} + I_{RRU}$ for Option-9 is zero since there is no data compression nor packetizing of packets. Improved hardware and software for Option-7 or Option-8 can reduce the latency but it will still be higher than what Option-9 can achieve. A similar argument can be applied to the transceiver delay. It is very important to mention that the difference in delay between Option-9 and other digital function split options will further increase due to other several factors, such as: wider wireless bandwidth, large packets size, network traffic conditions and the use of virtualization and switching devices. Therefore, it is quite challenging for other function split options to even approach the performance of Option-9, when only latency is considered.

Table 2 - Normalized round-trip fronthaul latency components

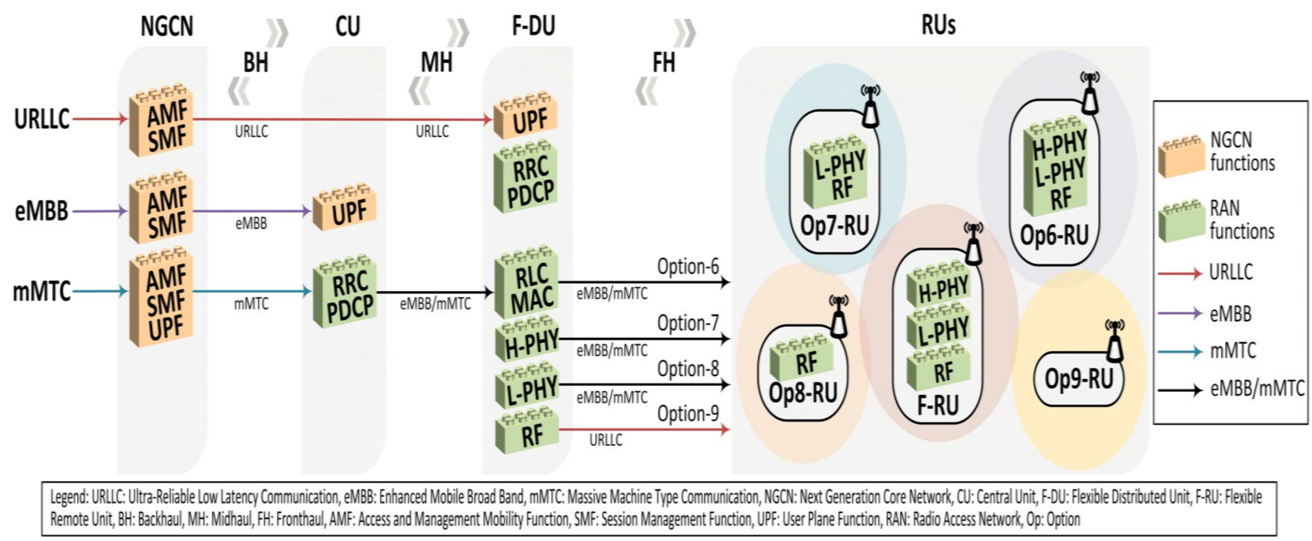
Split option	$I_{DU} + I_{RRU}$	$t_{transceiver}$	t_{budget}
Option-7	13.43%	25.2%	61.37%-$t_{switching}$
Option-8	20.24%	25.2%	54.56%-$t_{switching}$
Option-9	0%	0.012%	99.99%

Moreover, for 5G URLLC applications, the available fronthaul latency budget and the impact of the additional fronthaul latency on the end-to-end RTT are greatly different than the LTE case. In 4G-LTE networks, one of the major contributors to the high end-to-end RTT latency is the HARQ RTT, which is equal to 8 ms. This delay is composed of the transmission and MAC and PHY processing delays at both the eNodeB (eNB) and user equipment (UE). The transmission time is equivalent to the subframe duration, which is 1 ms. This leaves about total latency budget of 6 ms to be spent on the PHY processing at both eNB and UE. Any additional latencies generated by the fronthaul such as propagation and interface latencies are deducted from this latency budget. Therefore, the maximum allowed roundtrip latency budget is limited to 500 us for LTE, which represents about 8.33% of the PHY processing latency budget. As long as these limits are respected, saving some latency at the fronthaul will not benefit the end-to-end RTT but it can have other benefits such as relaxed processing time at eNB and UE or the ability to use longer fronthaul link lengths. In other words, in LTE networks, the end-to-end RTT is independent from the fronthaul latency as long as the HARQ RTT limit is not violated. In the case that HARQ RTT is violated, then retransmission mechanism will be triggered which can impact the latency.

However, for 5G URLLC applications, the HARQ RTT is greatly reduced to achieve the 1 ms end-to-end latency goal. First, the TTI is reduced to mini-slot of 2, 4, or 7 symbols durations, instead of a whole subframe, as shown in Fig. 1. The required PHY processing budget for each UE and gNodeB (gNB) depends on the used sub-carrier spacing. In case of 15 kHz sub-carrier spacing, the processing latency is 3-symbols durations, which results in a HARQ RTT of 10 symbols durations. For a carrier spacing of 30 kHz, the processing time budget is 4.5-symols durations and consequently, the HARQ RTT is 13 symbols durations. However, since the mini-slot in this case is 2-symbols durations, the HARQ RTT is rounded up to be 14 symbols duration [41].

In order to estimate the available fronthaul latency budget under URLLC applications scenarios, we can follow the same method used in LTE case and assume that about 8.33% of the processing latency budget can be allocated for fronthaul latency. Table 3 shows the expected fronthaul latency budget for URLLC applications and the maximum supported fronthaul link length if Option-9 is used. Considering the case of 2-symbols mini-slot with subcarrier spacing of 30 kHz, the HARQ-RTT is 466.62 us. Out of this 14-symbols-duration budget, 9-symbols duration is used for PHY processing at the gNB and UE. In addition to that, 1-symbol duration is not utilized in this numerology, and hence can be used for fronthaul latency budget. Therefore, the 30 kHz carrier spacing has lower overall HARQ RTT, and yet can support longer fronthaul links. In similar fashion, the fronthaul latency budgets for higher subcarrier spacings of 60, 120 and 240 kHz can be estimated once the corresponding processing time budgets are determined.

All in all, the available fronthaul roundtrip latency budget has decreased from few hundreds of microseconds, in C-RAN case, to few tens of microseconds so as to support URLLC applications.



Legend: URLLC: Ultra-Reliable Low Latency Communication, eMBB: Enhanced Mobile Broad Band, mMTC: Massive Machine Type Communication, NGCN: Next Generation Core Network, CU: Central Unit, F-DU: Flexible Distributed Unit, F-RU: Flexible Remote Unit, BH: Backhaul, MH: Midhaul, FH: Fronthaul, AMF: Access and Management Mobility Function, SMF: Session Management Function, UPF: User Plane Function, RAN: Radio Access Network, Op: Option

Fig. 8 - Flexible function split options assignment to support different 5G applications.

In the considered examples, the estimated fronthaul RTT budgets are 33 and 58 us for 15 and 30 kHz subcarrier spacings, respectively. It is also worth mentioning that the recent eCPRI standard defines four different classes of fronthaul latencies [34]. One of those classes is "high25" that targets URLLC application by limiting the fronthaul RTT budget to 50 us. This greatly reduced fronthaul latency budget is barely sufficient for optical signal propagation through a 5 km link, which makes it difficult for Option-7 and Option-8 to be able to support URLLC applications. Low layers function split options, except Option-9, add more challenges in terms of latency on both user and data planes due to the additional required processing such as (de)-compression, (de)-packetizing, electrical/optical conversions, etc. Therefore, adopting one of these split options can come at the cost of both increased latency for URLLC applications and limited fronthaul links length.

Table 3 – Expected roundtrip fronthaul latency budget for URLLC applications

Sub-carrier spacing	HARQ RTT (symbols)	HARQ RTT (us)	FH RTT budget (us)	Option-9 FH length
15 kHz	10	666.67	33.32	3.3 km
30 kHz	14	466.62	58.32	5.8 km

FH: fronthaul; Assumptions: mini-slot = 2-symbols durations, 8.33% of processing latency budget can be allocated for fronthaul latency.

7. Flexible Fronthaul Architecture: Design and Discussion

5G applications are categorized into three main categories, namely, massive machine type communication (mMTC), enhanced mobile broadband (eMBB) and ultra-reliable low latency communication (URLLC). Recently, there have been several proposals that discuss the concept of flexible function split to optimize the performance for each of these categories and to enable dynamic distribution of the processing workload [21], [42]–[47]. In order to optimize the performance of different 5G applications, several options need to be supported by a single network paradigm. Figure 8 illustrates different assignments of functions in the network for different application types. Functions can be classified into core functions, denoted by next generation core network (NGCN) and shown in orange color, and RAN functions that are shown in green color blocks. Many sources argue that more functions need to be placed near to the user end, including some NGCN functions, to reduce latency of the backhaul and midhaul, as described in IEEE Next Generation Fronthaul Interface (NGFI) standard [11]. However, the main accepted function split option for the fronthaul is Option-7 which, as we have shown, can introduce intolerable latency to the URLLC applications.

Based on our analysis, we propose to implement Option-9 for URLLC applications, which can bring two major benefits to the architecture. First, due to its better latency performance, it allows for longer distance transmission at the fronthaul. This enables more centralized locations of the DUs. In other words, by implementing Option-9, the DU can be placed closer to the CU, which results in higher statistical multiplexing gain due to greater sharing of resources. It also implies that, fewer number of DUs will be required for the same number of RUs. This reduces deployment cost since the URLLC-based DUs are expected to be costlier and more complex. Second major advantage of implementing Option-9 is latency reduction. Based on our results, Option-9 can achieve lower fronthaul latency, equivalent to $t_{propagation}$. As the maximum expected fronthaul length in 5G networks is about 20 km, the maximum one-way trip latency is about 100 us if Option-9 is used [30].

Figure 8 provides an example architecture that can facilitate URLLC applications. Reducing the latency at the backhaul and midhaul can be achieved by moving part of functions of the core network and all functions of the CU to the DU at the access level. Then, to reduce the latency at the fronthaul, Option-9 can be used as the function split for the edge node, based on our experimental results. The figure also depicts the overall heterogeneous fronthaul architecture where different RUs can have different function split options for any dynamically allocated time interval. The concept of flexible-RU (F-RU) wherein a single RU can support different function splits at the same time should also be considered [42]. Designing such a unified interface for different heterogeneous function splits with the lowest possible overhead and level of complexity constitutes an interesting research topic. To support flexibility, there should be functional overlap between different units (i.e. CU, DU, etc.) in the architecture. For example, to support both Option-7 and Option-8, both DU and RU should be able to perform the FFT/CP operations. This implies that there will be some redundancy, which can increase the cost and complexity of the system.

Flexible function splits can enable the F-RU and flexible-DU (F-DU) to support different function split options at the same time. This can promote UE-scale flexibility, which is a greater granularity of flexibility. Another lower granularity of flexibility can be achieved by changing the function split of the F-RU at different times. In this method, there is only one function split running at any point of time. This type of flexibility is referred to as RU-scale flexibility and can achieve several benefits such as saving energy at the RU and dynamic allocation of computational resources in the network [30]. We believe that adding Option-9 to the RU-scale flexible fronthaul can help to reduce power consumption at the RU at high traffic loads and provide an additional flexibility dimension for network operators [26].

By utilizing UE-scale flexibility, different types of applications can be supported by multiple function splits simultaneously. Adding Option-9 to the UE-scale flexible fronthaul can help to achieve the low latency requirements of URLLC applications. An intuitive flexible fronthaul control algorithm can be as follows: During idle periods, the RU will be operating in Option-8 mode to save energy at the RU. When the traffic profile starts increasing and Option-8 will be no longer able to accommodate the increasing number of connections, another higher split, for example, Option-7, can be added to the RU. The newly added Option-7 will serve all types of applications except the low latency ones, which will be using Option-9 and the RU will operate in dual-split mode.

8. Conclusion

As the research thrust to reduce latency for 5G applications is gaining momentum recently, identifying and quantifying the sources of latency has become an integral part of the development process. It has become clear that in order to achieve 1 ms end-to-end RTT requirement for URLLC applications, significant improvements in different aspects of network technologies have to be simultaneously considered. The new mechanisms designed to facilitate the URLLC applications make 5G systems very sensitive to any additional latencies at the fronthaul. Therefore, we have experimentally investigated the latency performance for different edge node complexity levels pertaining to the most promising function split options for the mobile fronthaul: Option-7, Option-8 and Option-9. The lowest function-split, based on Option-9 for the fronthaul interface design, increases the statistical multiplexing gain and allows the RU to have the simplest structure with the lowest fronthaul latency. Accordingly, we have presented a full 5G system architecture design, integrating Option-9, to support different futuristic 5G applications.

Our experimental results show that using Option-9 based edge nodes, also named remote units (RU), can eliminate the interface delay at the fronthaul while greatly reduce the optical conversion delay. The latency adhere to these three function split options was measured and compared experimentally. A significant diminution of fronthaul delay, in our experimental testbed, amounting to about 190 us for a 5 MHz wireless signal has been measured when Option-9 is used. This figure constitutes about 77% of the one-way trip latency limit for LTE fronthaul. Under 5G numerology, the fronthaul latency budget is greatly reduced to tens of microseconds range, which makes the usage of lower function split options very challenging. However, we have experimentally demonstrated that Option-9 can support lower latency at the fronthaul and allow longer fronthaul link lengths in integrated fiber-wireless access networks.

References

[1] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, "A survey on low latency towards 5G: RAN, core network and caching solutions," *IEEE Commun. Surv. Tutor.*, 2018.

[2] "Service requirements for the 5G system," 3GPP, TS 22.261, 2019.

[3] "Study on new radio access technology: Radio access architecture and interfaces." 3GPP TR 38.912, 2016.

[4] "Common public radio interface (ecpri); interface specification," IEEE Standards Association, 2017.

[5] T. Pfeiffer, "Next generation mobile fronthaul and midhaul architectures," *J. Opt. Commun. Netw.*, vol. 7, no. 11, pp. B38–B45, 2015.

[6] "Study on new radio access technology: Radio access architecture and interfaces." 3GPP TR 38.801, 2016.

[7] L. M. Larsen, A. Checko, and H. L. Christiansen, "A survey of the functional splits proposed for 5G mobile crosshaul networks," *IEEE Commun. Surv. Tutor.*, 2018.

[8] N. Alliance, "NGMN Overview on 5G RAN Functional Decomposition," *V1.0*, Feb. 2018.

[9] I. Chih-Lin, H. Li, J. Korhonen, J. Huang, and L. Han, "RAN Revolution with NGFI (xHaul) for 5G," *J. Light. Technol.*, vol. 36, no. 2, pp. 541–550, 2018.

[10] L. Cai, "RAN Node Definition for NGFI," IEEE Standards Association, 2017.

[11] F. Alam, "Dimensioning Challenges of xhaul (Reference document for discussion in meeting)," IEEE Standards Association, 2018.

[12] M. Anastasopoulos, A. Tzanakaki, D. Simeonidou, J. Bartelt, J. Zou, and M. Eiselt, "Architecture of Optical/Wireless Backhaul and Fronthaul and Evaluation," 2017, vol. D2.3.

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

[14] C. Ranaweera, E. Wong, A. Nirmalathas, C. Jayasundara, and C. Lim, "5G C-RAN with Optical Fronthaul: An Analysis from a Deployment Perspective," *J. Light. Technol.*, 2017.

[15] D. Chitimalla, K. Kondepu, L. Valcarenghi, M. Tornatore, and B. Mukherjee, "5G fronthaul-latency and jitter studies of CPRI over Ethernet," *IEEEOSA J. Opt. Commun. Netw.*, vol. 9, no. 2, pp. 172–182, 2017.

[16] L. Valcarenghi, K. Kondepu, and P. Castoldi, "Time-versus size-based CPRI in ethernet encapsulation for next generation reconfigurable fronthaul," *IEEEOSA J. Opt. Commun. Netw.*, vol. 9, no. 9, pp. D64–D73, 2017.

[17] F. Giannone *et al.*, "Impact of RAN Virtualization on Fronthaul Latency Budget: An Experimental Evaluation," in *Globecom Workshops (GC Wkshps), 2017 IEEE*, 2017, pp. 1–5.

[18] H. Gupta *et al.*, “How much is fronthaul latency budget impacted by RAN virtualisation?,” in *Network Function Virtualization and Software Defined Networks (NFV-SDN), 2017 IEEE Conference on*, 2017, pp. 315–320.

[19] N. Nikaein, “Processing radio access network functions in the cloud: Critical issues and modeling,” in *Proceedings of the 6th International Workshop on Mobile Cloud Computing and Services*, 2015, pp. 36–43.

[20] N. Nikaein *et al.*, “Closer to cloud-RAN: RAN as a service,” in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, 2015, pp. 193–195.

[21] M. Kist, J. A. Wickboldt, L. Z. Granville, J. Rochol, L. A. DaSilva, and C. B. Both, “Flexible Fine-Grained Baseband Processing with Network Functions Virtualization: Benefits and Impacts,” *Comput. Netw.*, 2019.

[22] FiberHome Technologies Group, “Comparison of three technical approaches of PON adaptation function in G.ROF.” ITU-T SG 15, Contribution# 1577, 2016.

[23] Futurewei Technologies US R&D Center, “G.RoF system architecture.” ITU-T SG 15, Contribution# 1723, 2016.

[24] Georgia Institute of Technology and Telekom Malaysia Berhad, “Proposal for Multiservice G.RoF optical architecture and configuration.” ITU-T SG 15, Contribution# 1967, 2016.

[25] Electronics and Telecommunications Research Institute (ETRI) and SK Telecom, “Proposal of G.RoF system architecture and reference configuration.” ITU-T SG 15, Contribution# 1733, 2016.

[26] A. Haddad and M. Gagnaire, “Radio-over-fiber (RoF) for mobile backhauling: a technical and economic comparison between analog and digitized RoF,” in *Optical Network Design and Modeling, 2014 International Conference on*, 2014, pp. 132–137.

[27] J. Wang, Z. Jia, L. A. Campos, and C. Knittle, “Real-Time Demonstration of 5-GSa/s Delta-Sigma Digitization for Ultra-Wide-Bandwidth LTE and 5G Signals in Next Generation Fronthaul Interface,” in *2018 European Conference on Optical Communication (ECOC)*, 2018, pp. 1–3.

[28] “Transport network support of IMT-2020/5G.” ITU-T GSTR-TN5G, Oct-2018.

[29] “Radio-over-fibre (RoF) technologies and their applications.” ITU-T Series G Supplement 55, Jul-2017.

[30] Y. Yoshida, “Mobile Xhaul evolution: enabling tools for a flexible 5G Xhaul network,” in *2018 Optical Fiber Communications Conference and Exposition (OFC)*, 2018, pp. 1–85.

[31] S. Delaitre *et al.*, “Intelligent Converged network consolidating Radio and optical access aRound USer equipment,” Deliverable report D2.3, 2017.

[32] S. Liu, P.-C. Peng, M. Xu, D. Guidotti, H. Tian, and G.-K. Chang, “A Long-Distance Millimeter-Wave RoF System With a Low-Cost Directly Modulated Laser,” *IEEE Photonics Technol. Lett.*, vol. 30, no. 15, pp. 1396–1399, 2018.

[33] S. Liu, Y. M. Alfadhli, S. Shen, M. Xu, H. Tian, and G.-K. Chang, “A Novel ANN Equalizer to Mitigate Nonlinear Interference in Analog-RoF Mobile Fronthaul,” *IEEE Photonics Technol. Lett.*, vol. 30, no. 19, pp. 1675–1678, 2018.

[34] “Common Public Radio Interface: Requirements for the eCPRI Transport Network,” IEEE Standards Association, V1.2, 2018.

[35] “Open Air Interface,” May-2019. [Online]. Available: https://www.openairinterface.org.

[36] M. Xu, F. Lu, J. Wang, L. Cheng, D. Guidotti, and G.-K. Chang, “Key technologies for next-generation digital rof mobile fronthaul with statistical data compression and multiband modulation,” *J. Light. Technol.*, vol. 35, no. 17, pp. 3671–3679, 2017.

[37] “Analyzing 8b/10b Encoded Signal with a Real-Time Oscilliscope,” Tektronics, 2011.

[38] M. Freiberger-Verizon and M. T. Watts-Verizon, “Low Latency Networks: Future Service Level Use Cases and Requirements,” in *2018 Optical Fiber Communications Conference and Exposition (OFC)*, 2018, pp. 1–3.

[39] G. Durisi, T. Koch, and P. Popovski, “Toward massive, ultrareliable, and low-latency wireless communication with short packets,” *Proc. IEEE*, vol. 104, no. 9, pp. 1711–1726, 2016.

[40] N. Alliance, “5G white paper,” *Gener. Mob. Netw. White Pap.*, pp. 1–125, 2015.

[41] “New Services & Applications with 5G Ultra-Reliable Low Latency Communications,” 5G Americas, White paper, Nov. 2018.

[42] Y. M. Alfadhli, M. Xu, S. Liu, P.-C. Peng, and G.-K. Chang, “Real-Time Demonstration of Adaptive Functional Split in 5G Flexible Mobile Fronthaul Networks,” in *Optical Fiber Communication Conference*, 2018, pp. Th2A–48.

[43] I. F. Akyildiz, A. Kak, E. Khorov, A. Krasilov, and A. Kureev, “ARBAT: A flexible network architecture for QoE-aware communications in 5G systems,” *Comput. Netw.*, vol. 147, pp. 262–279, 2018.

[44] I. Koutsopoulos, “Optimal functional split selection and scheduling policies in 5G Radio Access Networks,” in *Communications Workshops (ICC Workshops), 2017 IEEE International Conference on*, 2017, pp. 993–998.

[45] A. Maeder *et al.*, “Towards a flexible functional split for cloud-RAN networks,” in *Networks and Communications (EuCNC), 2014 European Conference on*, 2014, pp. 1–5.

[46] D. Harutyunyan and R. Riggio, “Flex5G: Flexible Functional Split in 5G Networks,” *IEEE Trans. Netw. Serv. Manag.*, vol. 15, no. 3, pp. 961–975, 2018.

[47] 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)*, 2018, pp. 1–6.