Giannone, F., Kondepu, K., Gupta, H., Civerchia, F., Castoldi, P., Franklin, A. A. y Valcarenghi, L. (2019). Impact of Virtualisation Technologies on Virtualised RAN Midhaul Latency Budget: A Quantitative Experimental Evaluation. *IEEE Communications Letters*, 23 (4), pp. 604-607

DOI: https://doi.org/10.1109/LCOMM.2019.2899308

Impact of Virtualisation Technologies on Virtualised RAN Midhaul Latency Budget: A Quantitative Experimental Evaluation

F. Giannone, *Student Member, IEEE,* K. Kondepu, *Member, IEEE,* H. Gupta, *Student Member, IEEE,*
F. Civerchia, *Student Member, IEEE,* P. Castoldi, *Senior Member, IEEE,* Antony Franklin A, *Member, IEEE,*
and L. Valcarenghi, *Senior Member, IEEE*

***Abstract*—In the Next Generation Radio Access Network (NG-RAN) defined by 3GPP for the fifth generation of mobile communications (5G), the next generation NodeB (gNB) is split into a Radio Unit (RU), a Distributed Unit (DU), and a Central Unit (CU). RU, DU, and CU are connected through the fronthaul (RU-DU) and midhaul (DU-CU) segments. If the RAN is also virtualised RAN (VRAN), DU and CU are deployed in virtual machines or containers. Different latency and jitter requirements are demanded on the midhaul according to the distribution of the protocol functions between DU and CU.**

This study shows that, in VRAN, the virtualisation technologies, the functional split option, and the number of elements deployed in the same computational resource affect the latency budget available for the midhaul. Moreover, it provides an expression for the midhaul allowable latency as a function of the aforementioned parameters. Finally, it shows that, the virtualised DUs featuring a lower layer split option shall be deployed not in the same computational resources where other vDUs are deployed.

***Index Terms*—5G, functional split, fronthaul, midhaul, virtualisation.**

I. Introduction

In the Next Generation Radio Access Network (NG-RAN), the next generation NodeB (gNB) protocol stack is divided (i.e., it is functionally split) among the following network components [1] [1]: (i) the Central Unit (CU), where, the gNB protocol stack upper layers (e.g. Packet Data Convergence Protocol — PDCP, Radio Resource Control — RRC) are hosted; (ii) the Distributed Unit (DU), where, the lower layers (e.g. Physical — PHY, Medium Access Control — MAC, Radio Link Control — RLC) are hosted, and (iii) the Radio Unit (RU) where the Radio Frequency (RF) functionalities reside. RU and DU communicate using a fronthaul interface (also called fronthaul I) while DU and CU communicate through a midhaul interface (also called fronthaul II). Several functional split options have been planned by 3GPP technical report TR 38.801 [2]. Each functional split option has got specific requirements in terms of data rate and latency [3].

Among the considered fronthaul interfaces, the Next Generation Fronthaul Interface (NGFI) [4] and the new Common Public Radio Interface (CPRI) specification for 5G called eCPRI [5] are the most mature ones. Moreover, the future 5G network architecture is expected to be massively based on Network Functions Virtualisation (NFV) [6]. The virtualisation of NG-RAN components (i.e., CU and DU) allows to move toward a Virtualised RAN (VRAN) and achieve the full potential of cost saving with rapid deployment of new services [7]. Because of the additional hypervisor layer (the fundamental building block of virtualisation [8]), the midhaul segment requirements in terms of latency and jitter, reported in 3GPP TR 38.801, may change and become more stringent.

So far, several works studied the impact of virtualisation on physical infrastructure sharing, isolation, cost, and energy saving of Long Term Evolution (LTE) networks [9]. However, the estimation of the effect of virtualising the NG-RAN components on the fronthaul/midhaul latency budget (i.e., the maximum allowable latency) has not been conducted in details yet. In [10] and [11], an evaluation of how different virtualisation technologies (i.e., VirtualBox, Kernel-based Virtual Machine, and Docker Container) decrease the midhaul latency budget considering CU virtualisation and Option 7-1 functional split is performed.

The current study stems from [10] and [11] but it evaluates many additional scenarios. In particular, several virtualisation technologies are utilised to virtualise not only the CU but also the DU. Both split Option 8 and Option 7-1 are considered. The midhaul latency budget and packet jitter (i.e., packet delay variation) budget are computed in all the possible combinations and compared with the scenario when CU and DU are deployed in bare metal. A mathematical model, expressing the midhaul latency budget as a function of the considered channel bandwidth, functional split options, and virtualisation technologies is provided and validated through experimental results. In an additional experimental analysis, how a vDU performance is impacted by virtualised elements (i.e., CU and DU) deployed in the same computational resource is studied. In this way, the need for anti-affinity constraint when a vDU is deployed is evaluated. Finally, the impact of deploying several vDUs/vCUs in the same host (i.e., the VRAN scalability) is experimentally evaluated.

F. Giannone, K. Kondepu, F. Civerchia, P. Castoldi, and L. Valcarenghi are with Scuola Superiore Sant'Anna, Italy, e-mail: name.surname@santannapisa.it

H. Gupta and A. Franklin are with Indian Institute of Technology Hyderabad, India, e-mail: {cs16mtech01001, antony.franklin}@iith.ac.in

[1] The LTE-Advanced terminology is utilized.

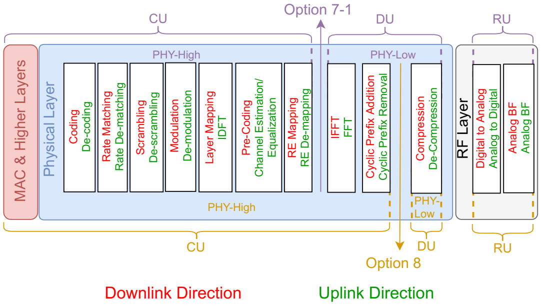

Fig. 1: Functional Split Options.

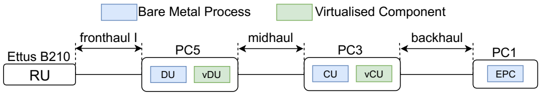

Fig. 2: Experimental Setup I.

II. Performance Evaluation Parameters and Evaluation Scenarios

The experimental analysis is carried out in the 5G segment of the Advanced Research on Networking (ARNO-5G) testbed [12]. The ARNO-5G testbed supports the emulation of the behavior of a 5G network and allows to perform specific measurements to calculate latency and jitter. OpenAirInterface (OAI) is utilized as a mobile network platform. The RU consists of an Ettus B210 Universal Software Radio Peripheral (USRP) while the UE consists of a Huawei E3372 dongle connected to a PC. The core network is implemented by means of openair-cn, which is an implementation based on 3GPP specifications of the Evolved Packet Core (EPC). OAI provides a C-RAN implementation of the IF5 functional split (equivalent to the split Option 8) and IF4p5 functional split (equivalent to the split Option 7-1). Both functional splits are Physical layer functional splits, as depicted in Fig. 1. For both the considered split options the midhaul latency requirement is about 250μs one way, as specified in 3GPP TR 38.801.

The considered performance evaluation parameters are the Allowable Latency Budget (ALB) and the Allowable Jitter Budget (AJB) of the midhaul segment[2]. The ALB and the AJB are defined as the maximum one-way latency and the maximum latency variation (i.e., delay jitter) supported by the midhaul segment without disconnection. A disconnection occurs when the latency and the jitter of the midhaul segment cause loss of synchronization between CU and DU. To emulate latency and jitter in the midhaul, the Linux traffic control (tc) tool is utilized. The tc utility is based on a token bucket filter and it is able to artificially add latency and jitter to a packet by caching it in the output interface before sending it on the link. Delays d0 and d1 are set to the midhaul Ethernet interfaces between DU and CU respectively.

In all the analyses presented in this work and described here below, the scenarios summarized in Tab. I are considered. Three virtualisation technologies are considered: Docker-Container, Kernel-based Virtual Machine (KVM), and VirtualBox (VB). We used general purpose PCs with Linux-based operating system. In all the scenarios two channel bandwidths i.e., 5 MHz and 10 MHz are considered.

TABLE I: Experimental Scenarios

Scenarios	Bare Metal	Virtualisation Technologies (Docker, KVM, VB)
Scenario 1 (S1)	DU and CU	x
Scenario 2 (S2)	DU	vCU
Scenario 3 (S3)	CU	vDU
Scenario 4 (S4)	x	vDU and vCU

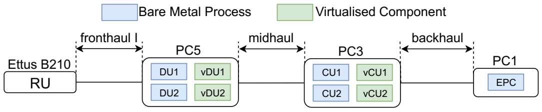

Fig. 3: Experimental Setup II.

In the first experimental analysis, ALB and AJB are measured by exploiting the ARNO-5G testbed configuration shown in Fig. 2. The EPC is deployed in PC1. The CU is deployed in PC3 either in bare metal (i.e., CU) or virtualised (i.e., vCU), and one UE is considered. Similarly, the DU is deployed in bare metal or virtualised in PC5. The selected combination depends on which of the four aforementioned scenarios and summarized in Tab. I, is considered. Based on the considered virtualisation technology, the vCU and the vDU are installed in a Docker Container or virtual machine (VM). The ALB of the midhaul segment is then obtained by increasing d0 and d1 delays until a disconnection event occurs. To evaluate the AJB, a fixed mean latency (i.e., a percentage of the ALB) and a supplementary random delay based on a normal distribution whose standard deviation is progressively increased are applied to the midhaul Ethernet interfaces of the DU and CU. The following two cases are examined. In the first case, the mean latency is set to 95% of the ALB and the supplementary random delay is increased to discern if jitter could be the origin of an ALB reduction. In the second case, the mean latency is set to 42.5% of the ALB and the supplementary random delay is increased to discern if jitter could be the limitation for the midhaul segment.

The second experimental analysis aims at understanding if an anti-affinity constraint is necessary when multiple virtualised mobile functions with different functional split options are deployed in the same host. The anti-affinity constraint forces Virtualised Network Functions (VNFs) to be deployed in different computational resources. To perform such analysis, a more complex network is deployed by doubling the involved NG-RAN components as shown in Fig. 3. An EPC, two CUs, two DUs, two RUs, and two UEs are deployed. The EPC is deployed in PC1. Either two bare metal processes of the CU or two vCUs are deployed in PC3. Similarly, either two bare metal processes of the DU or two vDUs are deployed in PC5. The vCUs and the vDUs are installed in a Docker Container or VM according to the considered virtualisation technology. The bare metal processes or the virtualised components are activated according to scenarios summarized in Tab. I. In such analysis, three cases are examined. In the first case, only the Option 7-1 functional split is implemented. In the second case, the Option 7-1 functional split is implemented between a CU-

[2] The contribution of the fronthaul to the overall latency is assumed to be negligible because the RU is not virtualised and the link between RU and DU is assumed to be short.

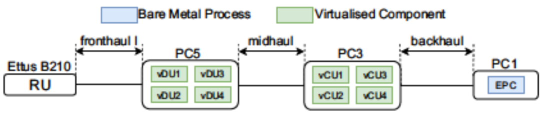

Fig. 4: Experimental Setup III.

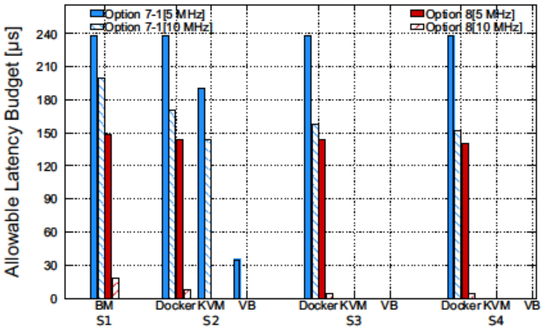

Fig. 5: ALB results for Experimental Setup I.

DU pair and the Option 8 functional split is implemented between the second CU-DU pair. In the third case, only the Option 8 functional split is considered.

In the third analysis, the scalability of the system is verified by increasing the deployed NG-RAN components up to four as shown in Fig. 4. The experiment is conducted as in the first considered experiment.

III. Experimental Results

Fig. 5 depicts the obtained ALB in the first experimental setup. The ALB obtained in S1 (i.e., the bare metal scenario) is considered as benchmark. As shown, the utilization of the Docker Container allows to reach the highest allowable latency budget in all the considered scenarios with virtualised elements because dockers are a lightweight virtualisation technology. Indeed, Docker Containers are a native application with respect to the host. Thus, they have a smaller footprint than the VMs implemented by means of KVM and VB. Furthermore, in KVM and VB, I/O virtualisation is performed by means of a hardware emulation layer under the control of the hypervisor, introducing additional delay. In addition, ALB heavily depends on the channel bandwidth: wider channel bandwidths mean a larger number of Physical Resource Blocks (PRBs), thus a high computing effort and a growing processing time are needed. In S2, if KVM is utilised, the ALB is zero when Option 8 is considered. Instead, with VB the UE is able to connect only when Option 7-1 and a 5 MHz channel bandwidth is used. In S3 and S4, ALB values are greater than zero with Docker Container only.

From the results reported in Fig. 5, it is possible to obtain an empirical formula that relates the ALB to the considered channel bandwidth, the functional split options, and the utilized virtualisation technologies. Based on [13] and [14], the ALB can be expressed as:

$$ALB = T_{TH}^{3GPP} \quad T_{proc}, \tag{1}$$

where T_{TH}^{3GPP} is the midhaul latency threshold [2] and the T_{proc} is the sum of processing time at the DU and the CU. Based on the experimental results T_{proc} can be linearly fitted as $T_{proc} = \alpha x+\beta$ where x is the considered channel bandwidth

TABLE II: α and β coefficients

Platform	Option 7-1		Option 8	
	α	β	α	β
BM	-7.6	276	-22.8	263
Docker	-12.6	301	-24.4	266
KVM	-13	247	x	x

and α and β are coefficients depending on the virtualisation technology and split option. Tab. II shows the α and β values estimated in the ALB experimental analysis performed by using the testbed shown in Fig. 2. Fig. 6 (top) shows the ALB trend in S1. The results depicted in Fig. 6 (middle) and Fig. 6 (bottom) are obtained in S2 with Docker Container and KVM, respectively. Note that only the Docker Container and KVM coefficients are calculated in S2 because the ALB values obtained when VB is used are too small to obtain a good fitting. As shown in Fig. 6, if split Option 7-1 is considered, Docker Container and KVM perform similarly as a function of the channel bandwidth (their α values are similar). As well, if split Option 8 is considered, BM and Docker Container perform similarly as a function of the channel bandwidth. These results confirm the capabilities of the Docker Container to achieve performance close to the ones of bare metal when the CU is virtualised.

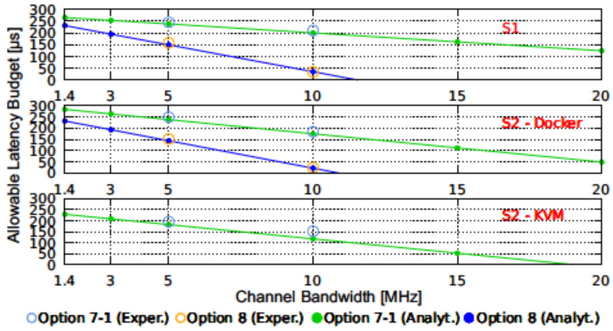

Fig. 6: ALB trends as a function of channel bandwidth.

Fig. 7 shows the AJB when the fixed mean latency is set equal to 95% of the ALB. The AJB values obtained in S1 are considered as benchmark. Fig. 8 shows the obtained results when the fixed mean latency is set equal to the 42.5% of the ALB. The results show that in the former case even a small jitter can cause a disconnection between the CU and DU. In the latter case, higher AJB is allowed. In both considered cases the AJB is in the interval between 20μs and 40μs. Thus the midhaul is very sensitive to jitter. In all the considered scenarios for split Option 8, AJB is zero. In S2 with VB, the UE is able to connect considering Option 7-1 and 5

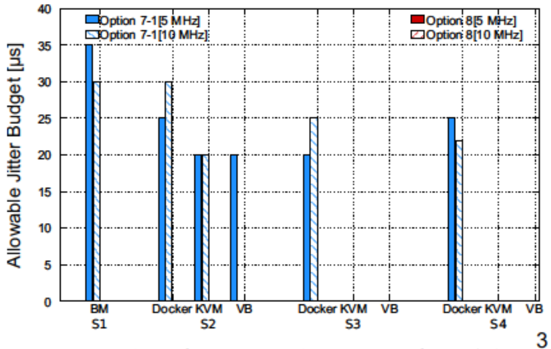

Fig. 7: AJB results when mean latency is the 95% of ALB.

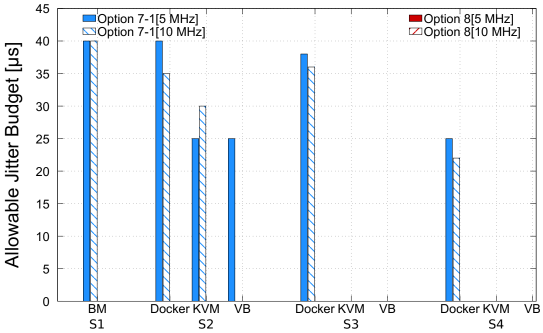

Fig. 8: AJB results when mean latency is the 42.5% of ALB.

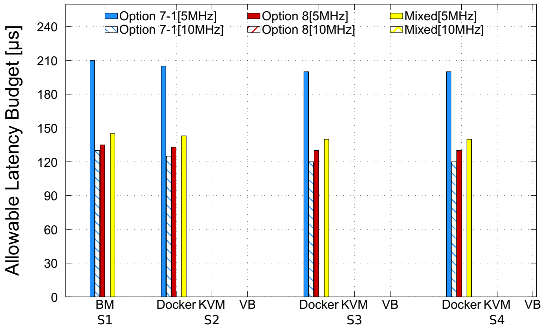

Fig. 9: ALB results in the anti-affinity constraint analysis

MHz channel bandwidth only. In S3 and S4, AJB values are obtained with Docker Container and Option 7-1 split only.

Fig. 9 shows the results obtained in the second experimental setup (i.e., anti-affinity experimental analysis). The obtained results show that, in the case of 5 MHz channel bandwidth, if split Option 8 and Option 7-1 coexist in the same computational resource (labelled as Mixed), the ALB decreases with respect when only Option 7-1 is considered (note that in this case the ALB is the time of the first UE disconnection, that is the disconnection of the UE whose data plane utilizes split Option 8). In case of the 10 MHz channel bandwidth, in the considered setup, split Option 8 and split Option 7-1 cannot be deployed together (the achieved ALB is zero). In addition, it is not possible to deploy two split Option 8 with 10 MHz channel bandwidth in the same computational resource. Thus, anti-affinity constraint shall be imposed if VNFs featuring split Option 8 are deployed to avoid that split Option 8 ALB is heavily impaired. For KVM and VB technologies UEs and DUs are not capable to communicate.

Fig. 10 shows the third experimental setup results (i.e., the scalability experimental analysis). Since the Docker Container resulted the best one among the analyzed virtualisation technologies, the scalability experimental analysis is performed only with Docker Container. Results show that, the ALB decreases if the number of virtualised DU-CU components increases due to the greater traffic load injected in the midhaul segment.

IV. Conclusion

In this work an experimental analysis of the effect of virtualising NG-RAN components (e.g., CU and DU) on the maximum latency and jitter that the midhaul can support has been performed. The first set of results showed that by using heavier virtualisation technologies and a higher number of physical resource blocks (i.e., channel bandwidth), the midhaul maximum latency decreases due to a heavier elaboration requested to the hardware. An empirical equation expressing the midhaul maximum latency as a linear function of the number of physical resource blocks (i.e., channel bandwidth), functional splits, and virtualisation technologies confirm the aforementioned trends. Moreover, even the midhaul jitter can be critical if it reaches values above 40μs. A second set of results showed that if virtual DUs and CUs featuring split Option 8 are deployed, the utilization of the anti-affinity constraint is advisable to avoid large impairment in terms of maximum supported latency. A third set of results showed that by increasing the number of NG-RAN components in the same computational resource the maximum midhaul latency heavily decreases.

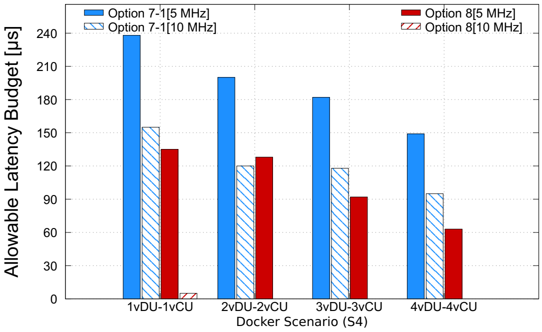

Fig. 10: ALB results in the scalability analysis.

Acknowledgment

This work was partly funded by the project H2020-ICT-2016-2 “5G-TRANSFORMER" (761536).

References

[1] Y. Yoshida, “Mobile Xhaul Evolution: Enabling tools for a flexible 5G Xhaul network,” in *OFC 2018*, San Diego, California, US, Mar. 2018.

[2] “Technical Specification Group Radio Access Network, Study on new radio access technology; radio access architecture and interfaces,” 3GPP, Technical Report (TR) 38.801, Mar. 2017, version 2.0.0.

[3] P. Sehier *et al.*, “Transport network design for fronthaul,” in *VTC-Fall 2018*, Toronto, ON, Sept. 2017.

[4] “Next Generation Fronthaul Interface (1914) Working Group,” http://sites.ieee.org/sagroups-1914/, accessed: February 11, 2019.

[5] CPRI, “Common Public Radio Interface: eCPRI Interface Specification,” eCPRI Specification, Interface Specification, Aug. 2017, version 1.0.

[6] “5G Radio Access Capabilities and Technologies,” White Paper, ERICSSON, Apr. 2016.

[7] “Network operator perspectives on NFV priorities for 5G,” White Paper, ETSI, Feb. 2017.

[8] A. Blenk *et al.*, “Survey on network virtualization hypervisors for software defined networking,” *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 655–685, Firstquarter 2016.

[9] L. Zhao *et al.*, “LTE virtualization: From theoretical gain to practical solution,” in *ITC 2011*, San Francisco, California, USA, Sept. 2011.

[10] H. Gupta *et al.*, “How much is fronthaul latency budget impacted by ran virtualisation ?” in *NFV-SDN 2017*, Berlin, Germany, Nov. 2017.

[11] F. Giannone *et al.*, “Impact of RAN Virtualization on Fronthaul Latency Budget: An Experimental Evaluation,” in *GC Wkshps 2017*, Singapore, Dec. 2017.

[12] “ARNO-5G testbed,” http://arnotestbed.santannapisa.it/, accessed: February 11, 2019.

[13] C. Chang *et al.*, “Impact of packetization and scheduling on C-RAN fronthaul performance,” in *GLOBECOM 2016*, Washington, DC USA, Dec. 2016.

[14] N. Nikaein, “Processing radio access network functions in the cloud: Critical issues and modeling,” in *MCS Wkshps 2015*, Paris, France, Sept. 2015.