HW/SW Development of Cloud-RAN in 3D Networks: Computational and Energy Resources for Splitting Options

Stefano Bonafini, Claudio Sacchi, Fabrizio Granelli
University of Trento
Department of Information Engineering
and Computer Science (DISI)
Via Sommarive 9
Trento, Italy 38123
stefano.bonafini@unitn.it
claudio.sacchi@unitn.it
fabrizio.granelli@unitn.it

Sisay Tadesse Arzo, Michael Devetsikiotis
University of New Mexico
Electrical and Computer
Engineering Department (ECE)
498 Terrace St NE
Albuquerque, New Mexico, 87106
sarzo@unm.edu
mdevets@unm.edu

Koteswararao Kondepu
Department of Computer Science and Engineering
Indian Institute of Technology Dharwad
WALMI Campus, PB Road, near High Court
Karnataka 580011, India
k.kondepu@iitdh.ac.in

2023 IEEE Aerospace Conference | 978-1-6654-9032-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/AERO55745.2023.10115760

Abstract—**The continuous increase in demanding for availability and ultra-reliability of low-latency and broadband wireless connections is instigating further research in the standardization of next-generation mobile systems. 6G networks, among other benefits, should offer global ubiquitous mobility thanks to the utilization of the Space segment as an intelligent yet autonomous ecosystem. In this framework, multi-layered networks will take charge of providing connectivity by implementing Cloud-Radio Access Network (C-RAN) functionalities on heterogeneous nodes distributed over aerial and orbital segments. Unmanned Aerial Vehicles (UAVs), High-Altitude Platforms (HAPs), and small satellites compose the Space ecosystem encompassing the 3D networks. Recently, a lot of interest has been raised about splitting operations to distribute baseband processing functionalities among such nodes to balance the computational load and reduce the power consumption. This work focuses on the hardware development of C-RAN physical (PHY-) layer operations to derive their computational and energy demand. More in detail, the 5G Downlink Shared Channel (DL-SCH) and the Physical Downlink Shared Channel (PDSCH) are first simulated in MATLAB environment to evaluate the variation of computational load depending on the selected splitting options and number of antennas available at transmitter (TX) and receiver (RX) side. Then, the PHY-layer processing chain is software-implemented and the various splitting options are tested on low-cost processors, such as Raspberry Pi (RP) 3B+ and 4B. By overclocking the RPs, we compute the execution time and we derive the instruction count (IC) per program for each considered splitting option so to achieve the mega instructions per second (MIPS) for the expected processing time. Finally, by comparing the performance achieved by the employed RPs with that of Nvidia Jetson Nano (JN) processor used as benchmark, we shall discuss about size, weight, power and cost (SWaP-C) metrics related to the UAV payload design.**

TABLE OF CONTENTS

978-1-6654-9032-0/23/$31.00 ©2023 IEEE

1. INTRODUCTION

We are living in an era of very fast, and sometimes overlapping, evolution of mobile standards. 4G standard (LTE and LTE-A) is still providing everywhere cellular connectivity to smartphones and other kinds of devices. In parallel, mobile 5G is taking place in many parts of the World, with the aim of becoming the dominant standard within the term of 2030. The statistics collected by Ericsson account 56 countries in the World currently reached by 5G services [1]. However, research and development on mobile communications is continuously moving forward, investigating the networks of the future “beyond 5G”. In this framework, some position papers have been published in the open literature. In [2], the 6G vision is summarized into some fundamental concepts, namely: digital twins, ubiquitous intelligence, softness, native AI and security. In the work of Zhu, *et. al.* [3], the foundations for mobile intelligence networks are given in form of ABC, where (A) stands for artificial intelligence, (B) for big data and (C) for cloud computing. One of the key points of 5G and 6G is the integration of terrestrial mobile communications with satellite communications. In [4], a two stage of system integration of satellite and terrestrial segments: the first stage is compatible with 5G, where the current satellite system can be developed upon 5G guidelines with the maximum reuse of 5G technology; the second stage will be integrated within 6G with high, medium and low orbit payloads harmonically cooperating as a whole. Such a concept has been already stated in [5] as one of the key points of the “Space 2.0” vision. We can say that 6G development and a renewed vision of Space communications are proceeding together.

Multi-layered connectivity involving terrestrial and aerial connections will be one of the distinguishing features of 6G. In [6], Wang *et. al.* investigate the potential of multi-layered hierarchical non-terrestrial networks (NTNs) in a 6G perspective. NTNs, made of integrated terrestrial nodes, UAVs, HAPs and small satellites play a leading role in 5G and beyond by covering different verticals, including healthcare, intelligent transportation, public safety, and many others [6]. They can offer highly desirable features like: service continuity, service ubiquity and service scalability.

One of the key enabling technologies of 6G will be the Cloud Radio Access Network (C-RAN). The concept of C-RAN has been already formalized by 3GPP group in the framework of 4G and 5G [7]. The Base-Station (BS) is separated into two units: Remote Radio Head (RRH) and Base Band Unit (BBU), as it was since the 3G advent. The RRH is located close to the antenna in the cell site tower, while the BBU, which takes care of all the baseband processing, is conveniently centralized, or better “cloudified”, in a remote location, called BBU-pool. The fronthaul network connects through a fiber link between the RRH and BBU. The Common Public Radio Interface (CPRI) is the protocol currently used to transmit the fronthaul signals and its specifications are given in [8]. C-RAN advantages are recognized in terms of: i) simpler implementation and easier maintenance of RRH, ii) improved spectral efficiency, service scalability, agile interference management thanks to BBU coordination, iii) efficient implementation of self-organizing networks (SONs), iv) faster handover among cells in the same BBU-pool and, last but not the least, v) improved energy efficiency obtained by relieving computational-demanding tasks from RRH located supplied by batteries and/or renewable power sources. However, C-RAN suffers from the high demand of capacity on the fronthaul network. To this aim, *functional splits* can effectively manage and allow to select the amount of functions to be operated at the antenna site, and the amount of functions centralized at BBU-pool side. In 5G NR the radio and baseband processing are divided into radio unit (RU), distributed unit (DU) and centralized unit (CU). From a physical perspective, the RU and DU are collocated at the antenna mast, thus closer to the user, while the CU is moved at the BBU-pool [7]. More functions run by DU involve a bitrate reduction on the fronthaul link, vice-versa, operations at CU can exploit the high processing capabilities of the BBU-pool hardware. 3GPP standardized eight splitting options to detach at different layer levels some radio or baseband processing tasks and distributing them through the cloud. It is evident at a glance that functional split adds to the network a high degree of flexibility and reconfigurability.

More recently, the C-RAN concepts have been profitably applied to multi-layered NTNs, namely 3D networks, with the functional splitting managed among UAVs and small CubeSats. In [9] and [10] the use of C-RAN for 3D network in border monitoring applications has been considered and assessed. Such kind of techniques have also been introduced in a completely different and very challenging use-case, i.e.: the provision of broadband mobile connectivity on Mars in the framework of future manned missions. In the Martian environment, it is not realistic to install on-site mobile networking infrastructures, like base stations, mobile switching centers, etc. So, the mobile connectivity should be obtained by virtualizing the various network functions over processing units (PU) that will be conveniently hosted on board of vehicles (rovers, landers), UAVs, CubeSats and orbiters that operates on Mars for exploration purposes. In [11], the feasibility of functional splitting in 3D Martian networks has been analytically assessed, while the fronthaul network design has been tackled in [12]. Finally, the end-to-end (E2E) performance evaluation of the Martian C-RAN setup is dealt in [13], where some interesting design and implementation trade-offs have been clearly highlighted.

This paper tackles, from an experimental viewpoint, the issues related to the computational burden inherent to the various available splitting options in the framework of 3D networks. In particular, we focus our attention on the most critical point of the network chain: the UAV. It is known that UAVs can host very limited resources and their service continuity represents a potential bottleneck for the entire system. A complete BBU-pool implementing all the required functionalities would require a power consumption of 1-2KW, that is mostly unfeasible. Therefore it is of paramount importance to move a proper amount of computational load to the upper-layer of the network, as already formalized in [9] and [11].

Our experimental setup consists of a laptop (Macbook Pro equipped with Apple M1 Pro chip), Raspberry Pi processors (3B+ and 4B series) and a multimeter, which is used to check the voltage swing and current flow. We have taken advantage from the NR DL-SCH and PDSCH implementation offered by the 5G toolbox of MATLAB 2021b version. We have measured first the achievable throughput per slot and processing time to perform the whole NR DL-SCH and PDSCH transmission. The scenario consider two different multi-antenna configurations, namely: single-input multiple-output (SIMO), which means hardware complexity at user equipment (UE) side, and multiple-input single-output (MISO), which moves hardware complexity at UAV side. This was considered to better clarify in which case the number of antennas at TX or RX side would significantly impact on the optimal selection of the splitting options, which have been analyzed thanks to timestamps added through the sequence of processing operations. Some gradient plots will clearly highlight the benefits led by performing opt. 6, 7.3, 7.2, 7.1 and 8 in terms of saved processing time. Moving on, we have deployed a 8 × 1-TX DL-SCH and PDSCH on low-budget PUs, namely the Raspberry Pi (RP) 3B+ and 4B with, respectively, $1\,GB$ and $2\,GB$ of RAM. The RPs have been overclocked to visualize the processing improvement. Thanks to the *perf* command-line utility, we have derived the instruction count (IC) per program, meaning the number of instructions composing the 8 × 1-TX functionalities chain. The mega instructions per second (MIPS) that should be assured by the PUs have been estimated upon an achievable clock frequency and expected processing time per single slot of the 5G NR frame. From this, the number of PUs needed to equip the UAV with is easily obtained by looking at some PU data sheets. Thus, we finally compare size, weight, power and cost (SWaP-C) metrics for a possible system design based on RPs and on Nvidia Jetson Nano, this last used as benchmark. In such a manner, a further analysis step of the actual feasibility of 3D networks for future 6G connectivity will be completed, thus providing useful practical hints for the development of such complex systems.

The paper is structured as follows. Sect.2 will discuss the innovative contribution of the paper by means of a comparison with the literature. Sect.3 will detail the proposed methodology for computational and energy consumption assessment, along with the HW/SW experimental setup adopted for measurements. Sect.4 will present and discuss some selected experimental results. Paper conclusion are finally drawn in sect.5.

2. Related work

Network function splitting

Network function splitting and decomposition provide maximum flexibility for a distributed deployment of network functions, which was previously designed as a cumbersome monolithic system. Splitting enables the monolithic system to be decoupled into sub-functions that can be designed as microservice or multi-agents. These smaller units can be deployed in distributed environments such as fog, edge or cloud-based computational resource provisioning environments. A recently published article in [14] proposed a multi-objective-based network function splitting considering 5G networks with a particular focus on network slicing. Focusing on different perspectives of a functional split between the radio access points and the edge/cloud platform, they presented a mixed integer quadratically constrained programming (MIQCP) framework for efficient placement of virtualized network functions (VNF) chains in future 5G systems. The authors detailed a deep performance analysis of split points between central cloud and distributed radio units.

In [15], a network function splitting and migration for SDN controller load balancing and overhead reduction is shown. Similarly, a microservice-based full decomposition and containerization of SDN controller are discussed in [16]. The authors provided a decomposition framework for the design performance evaluation of a microservice-based SDN controller. However, the authors didn't consider a resource-constrained environment, which may affect the deployment of the decomposed functions.

The application of AI for function splitting has also been studied. In [17], the authors discussed a deep reinforcement optimization for a virtualized radio access networks function splitting. The author discussed how to optimize the exact placement and location of the decomposed RAN functions by developing a learning paradigm. The problem of placing functions either in a central (cloud) or a distributed environment is formulated to minimize the total network cost using constrained neural combinatorial reinforcement learning. The discussion is primarily focused on the functions relocation in classic edge/fog versus cloud-based environment.

The work in [18] presented another complete decomposition of monolithic network system towards achieving a zero-touch autonomous network. The authors considered a multi-agent approach for service design. They proposed the functions to be designed as an atomic unit with autonomous capability of implementing the smallest functions. Using these units as bricks, a fully autonomous network management system could be designed. The work in [19] is another interesting approach to virtual network function splitting.

Functional decomposition demonstrates greater advantages in terms of flexibility, modularity, streamlining of cellular sites, and improved coordination with a migration towards v-RAN. However, the classic backhaul network would not support the improvement required by 5G. Therefore, in 5G we are witnessing a transition from backhaul (including fronthaul, midhaul and backhaul) to x-haul to improve 4G's mobile backhaul considering higher data rates, lower latency, hard isolated slices, higher reliability, and dynamic connectivity targeting on-demand services which consists of multiple physical and/or virtual components. To this aim, by exploiting splitting concepts, an E2E slicing to create a protected and dedicated path is proposed in [20]. The proposed technique targets to minimize the physical resource required for a given network slice. Such an approach augments system reliability by improving a single node/link failure.

As a use case analysis, the work in [21] presented a 5G network connectivity between the drone and the ground control station dimensioning the latency for ultra-reliable low latency (URLLC) applications with limited analysis for resource availability and energy consummation.

3D Networks-based Cloud-RAN

A significant application of split and decoupled system design is when there is the need for the deployment of functions in a resource-constrained environment such as resource-constrained IoT devices, UAVs, satellites, etc. As briefly introduced before, the splitting of virtualized BBUs (v-BBUs) using both small satellite platforms, e.g. CubeSats, and UAVs is explored in [9] and [10]. The authors targeted a high-level functional splitting of BBU for border security applications introducing the concept of multi-layered (3D) networks for providing connectivity through a UAV-based RU. The functional splitting is aimed at reducing the required function deployment on the UAV. This would enable reduced energy consumption, computational resources with improved battery life and UAV carried weight. However, there is a lack of knowledge about the required resources to design the system. Recently, along the same lines, splitting options to implement C-RAN in 3D networks and furnish "Towards 6G" connectivity to human personnel on Mars have been studied in [11] and in [12], where, respectively, deep attention has been paid to latency and bandwidth requirements to perform, as defined from 3GPP Release 15, opt. 6, 7.3, 7.2, 7.1, and 8, which will be the core theme of the successive discussion. Finally, the study presented in [13] shows E2E performance of 3D Networks for "Towards 6G" Martian connectivity.

Contribution of the paper

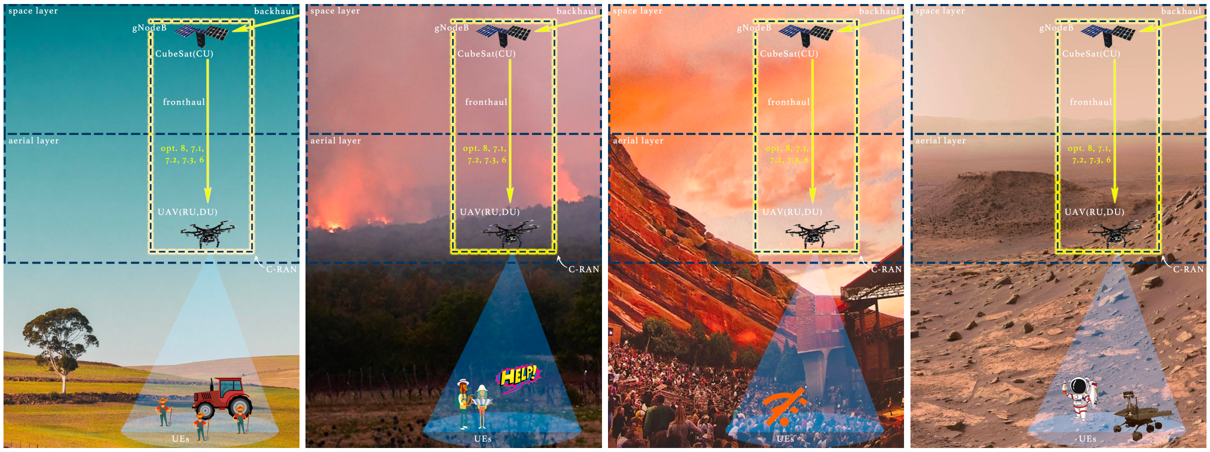
Although, the trade-offs analysis in [13] for system design strongly highlights the burdens and complexity of such networks, for instance the interdependence between design metrics in terms of latency, data rate, coverage, satellite lifetime and radio resources, considerations around the amount of energy and computational resources on-board of UAVs still need to be made to, especially, increase the service time. This applies to the Martian case, but also to the plethora of possible applications that we foreseen in the near future. Some of them are depicted in Fig. 1.

In this regard, the software-based implementation of DL-SCH and PDSCH functionalities on low-cost and resource-constrained PUs needs to be experimentally evaluated and linked with splitting opportunities to be statically, or even dynamically, adopted on heterogeneous nodes within 3D networks. To the best of authors' knowledge, our work is novel in providing a practical methodology to dimension and evaluate 3D networks-based C-RAN from the UAV point of view.

3. The proposed methodology

3D network architecture and specifications

Next generation networks will rely on an integration between Terrestrial Networks (TN) and NTN ones. In particular, 6G networks are expected to integrate TN and NTN in a joint TN-NTN framework [22], thus providing an autonomous yet



(a) Agricultural areas (b) Disaster areas (c) Massive events (d) Martian missions

Figure 1: Possible use-cases supported by 3D network implementing C-RAN to bring efficient mobile connectivity. (a) Connectivity for agricultural applications. (b) Connectivity for disaster and high-priority areas. (c) Connectivity for massively populated events. (b) Connectivity for future (human) activities on Mars.

reconfigurable multi-layered ecosystem. In such a panorama, 3D networks will play a major role in the transition from 5G to 6G connectivity. 3D networks are ground-to-space networks based on heterogeneous nodes. We can divide 3D networks into three main segments: *on-ground*, *aerial* and *space*. Sensors, communications peripherals or even bigger data processing devices constitute the *on-ground* segment and are regarded as the UEs, or end-users of the infrastructure [23]. The *aerial* segment is composed of flying nodes able at hosting communication payloads. The recent state-of-the-art focused, especially, on fleets of UAVs, or drones, however, High Altitude Pseudo Satellites (HAPS) are also mentioned to be part of the in-flight segment. Their main difference is on the altitude at which they can orbit. The National Aviation Authorities (NAA) across the world generally impose 120 m. as maximum flight height for commercial drones, which are classified as nano UAVs according to the US department of defense [24]. Apart from that, there are other 4 groups of UAVs, which are differentiated on normal operating height. Group 5 takes into account all those devices able to fly over ≈ 550 m. with a take-off weight of more than ≈ 600 kg.

On the other hand, ITU Radio Regulations (ITU-RR) about HAPS specifies them as radio stations flying at an altitude between $20 - 50$ Km. [25]. HAPS advantages are related to the possibility to be equipped with more complex payloads, to offer longer service time and a larger on-ground footprint. On the other side, UAVs ease the deploying phase, increase the network flexibility due to their dynamicity and maneuverability, ideally reduces the UEs complexity being closer to them, and, overall, improve the network scalability and maintainability.

The upper layer of the 3D network is the *space* segment, where satellite platforms reside. In this context, CubeSats are the smallest satellite platforms, which received the greatest interest from the scientific community but also from private companies. Indeed, Space X already developed a sort of NTN through the use of CubeSats in Low-Earth Orbit (LEO). CubeSats are cubic shaped satellites whose dimensions are standardized by Cal Poly in [26]. 1U CubeSats are $10 \times 10 \times 10$ aluminium cases hosting payload in an available volume which linearly increases as more CubeSats unit are integrated together. 1.5U, 2U, 3U, 6U and 12U CubeSat form factors have been also standardized [26], while other formats should be considered on a mission-to-mission basis and discussed with the launch providers. CubeSats' LEO altitude is selected depending on mission requirements. Numbers suggest that the 75% of on-orbit CubeSats missions are flying in the 350–700km range. However, implementing C-RAN based 3D networks strictly bounds the altitude range.

CRAN requirements and constraints analysis

C-RAN is the technology pooling baseband resources between the base stations (BSs) [27]. The advantages of such an infrastructure have been already discussed in sect.1, but the requirements and constraints need to be better characterized. First of all, C-RAN needs to deliver data from the UE to the BS, or vice-versa, within a limited amount of time T_{radio}, which varies accordingly to the application we want to serve. For 5G NR and low-latency applications, ITU fixes the one-way latency $T_{radio} = 0.5$ms [28]. Adapting the formulation of T_{radio} to 3D networks, the following relation holds [29]:

$$T_{radio} = t_Q + t_{FA} + t_{TX} + t_{bsp} + t_{mpt} \tag{1}$$

$$t_{tx} = t_{tx}^{UE-UAV} + t_{tx}^{UAV-CS} \tag{2}$$

$$t_{bsp} = (t_{bsp}^{UAV} + t_{CPRIp}^{UAV}) + (t_{bsp}^{CS} + t_{CPRIp}^{CS}) \tag{3}$$

t_Q being the queueing delay, t_{FA} a delay contribution due to frame alignment, t_{tx} the transmission delay, t_{bsp} and t_{mpt} the baseband processing latency and the user processing latency, respectively. Under the hypothesis that all the delay terms are negligible except the transmission delay, the maximum allowed distance between UE and CubeSat would be $d^{UE-CS} = 150$km. Consequently, the CubeSats should orbit in the very-low-Earth orbit (VLEO), which would strongly influence the system performance in terms of lifetime and coverage. Moreover, t_{tx}^{UAV-CS} is the time during which the baseband processing related to splitting should be completed. As per 3GPP, ideally $t_{tx}^{UAV-CS} < 0.25ms$ for low-level splitting options (e.g., opt.6, 7.1, 7.2, 7.3, and 8). On the other hand, practical E2E evaluations demonstrated the possibility to relax this constraint up to $t_{tx}^{UAV-CS} = 0.67ms$, thus increasing also t_{tx} without worsening too much the overall system performance [13]. However, assuming the

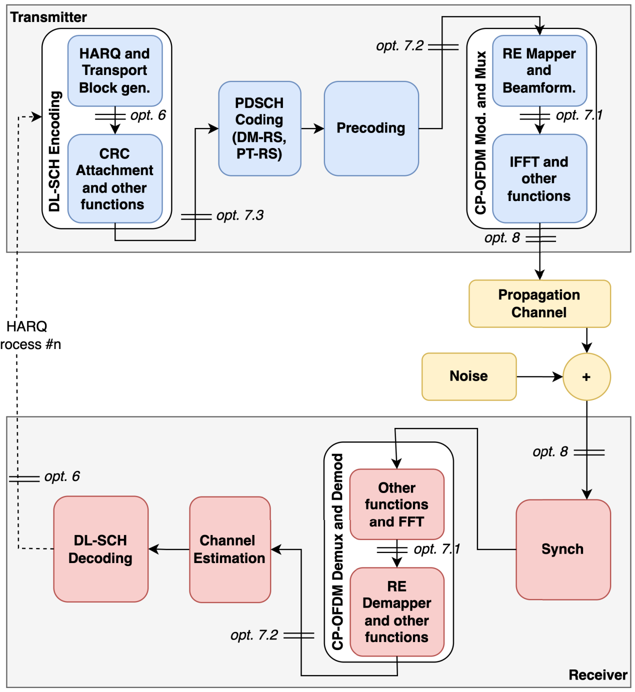


Figure 2: Encoding, modulating, demodulating and decoding functionalities composing the DL-SCH and PDSCH.

processing delay to be null is an unrealistic consideration, in particular, when dealing with complex functionalities, as the baseband processing, and limited resources available on UAV and CubeSats. We previously mentioned that the C-RAN architecture also requires a broadband fronthaul link between the RRH and the cloudified BBU, especially in 3D networks, where the fronthaul must be wireless and its quality-of-service (QoS) may be bounded by the CubeSat and UAV payload limitations. Again, this is why splitting options have been introduced. Indeed, splitting the processing chain at different levels translates into a lowered midhaul (fronthaul when decoupling BS into RU-DU-CU) data rate and an optimized usage of computational and energy resources. This means that, as we split the computational load $\mathbb{L}$, we are also splitting the radio equipment size, weight and power consumption between the UAV and CubeSat. In Eq.3, t_{bsp} is the sum of $t_{bsp}^{UAV} + t_{CPRIp}^{UAV}$, which are the splitted baseband processing time and CPRI processing time at UAV side, respectively. The latter has been reported [30] to be $t_{CPRIp}^{UAV} + t_{CPRIp}^{CS} = 10\mu$s, considering the round-trip time of terrestrial fiber connections. In sect.4 t_{bsp}^{UAV} will be referred to the baseband processing per slot.

As clearly understandable, the study regarding t_{bsp} becomes fundamental for 3D networks. Saving processing time means to increase the UE-to-CubeSat distance, which is necessary to increase satellite lifetime and system coverage. t_{bsp}^{UAV} may seriously represent a system bottleneck. Staying in-orbit, we can deploy larger satellite platforms into higher orbits, while on-air UAVs performance are limited by the reduced flight time, which is usually around 25 min. [31], reduced volume, supported load and energy availability.

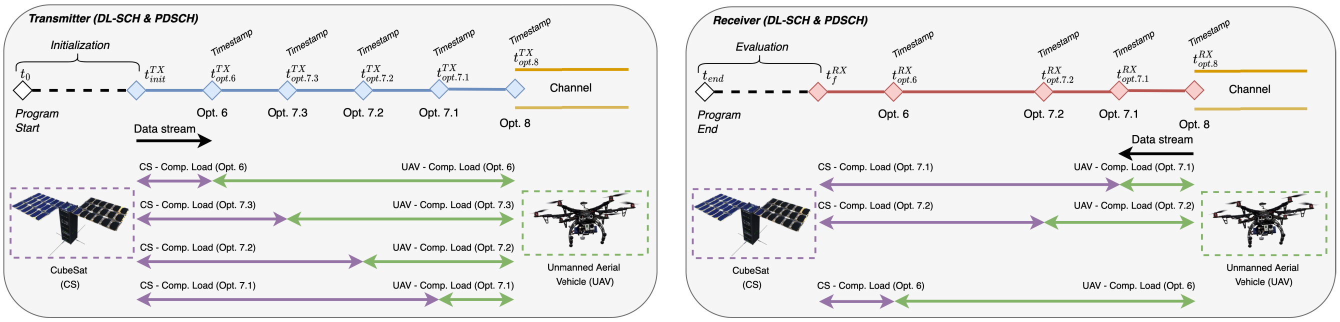
New Radio (NR) processing functionalities

Fig. 2 shows the processing chain, implemented by using the MATLAB 5G toolbox, namely: NR Downlink Shared Channel (DL-SCH) and Physical Downlink Shared Channel (PDSCH). Here, the implemented system is primarily subdivided into three different modules: *Transmitter*, *Propagation channel* & *Noise*, and *Receiver*. As shown in Fig. 2, *Transmitter* module provides DL-SCH transport channel encoding, which internally performs Hybrid ARQ (HARQ), transport block generation, and Cyclic Redundancy Check (CRC) attachment along with other functions. The transport channel is used for the transmission of user data, dedicated control bits, user-specific and downlink system information. The PDSCH is the physical channel carrying the DL-SCH coded data. Here, Demodulation Reference Signal (DM-RS) and Phase Tracking Reference Signal (PT-RS) are associated with the PDSCH. The DM-RS is used to estimate the channel at demodulator side, and the PT-RS is used to compensate the Common Phase Error (CPE). Upon obtaining the perfect channel estimation for the considered subframes and averaging all the allocated resource blocks over time and frequency, precoding is obtained using singular value decomposition (SVD). The output of the precoding is passed to the Cyclic-Prefixed Orthogonal Frequency Division Multiplexing (CP-OFDM) modulator and multiplexer, which manages the transmission by supporting variable subcarrier spacing, e.g., 15kHz, 30kHz, 60kHz, 120kHz. Clustered Delay Line (CDL) and Tapped Delay Line (TDL) are the two channel models supported in the considered implementation. The *Receiver* module, finally, performs PDSCH and DL-SCH demultiplexing, demodulation and decoding functions.

Fig. 2 also shows the different splitting options (namely: opt.6, 7.3, 7.2, 7.1 and 8) that can be performed at transmitter and receiver side. Fig. 3a shows the corresponding computational load distributed between CubeSat and UAV at the transmitter side. The computational load and the required corresponding PUs at the UAV might change based on the selected split option. Hence, it is important to know which resources are required to perform the considered functions. Similarly, Fig. 3b shows the computational load offloaded at the CubeSat and UAV for the receiver. Finally, the HARQ process either transmits new transport data or retransmits the previously delivered transport data depending on Acknowledgement (ACK) or Negative Acknowledgement (NACK) determined by the CRC check.

The above implementation is realized by setting some basic parameter values. As per 3GPP specifications, we consider the "Numerology 1", thus $N_{slot} = 20$ within a radio frame of 10ms (the selection of one between the 6 numerologies depends on the physical channel conditions). This means that the slot duration is $t_{slot} = 0.5msec.$ with a subframe composed of 2 slots. Each slot contains 14 OFDM symbols. An OFDM symbol lasts 33.33μsec. and the CP length is 2.34μsec., leading to an overall symbol duration of 35.68μsec. The slot is divided into two segments. We fixed DL-SCH transport channel coding codewords to 1, as well as the number of transmission layers. The frequency domain sub-carrier spacing is set to $\delta f = 30$ KHz and 12 sub-carriers are contained in a resource block (RB). Assuming a signal bandwidth of 20MHz, 51 RBs are taken into account.

Other significant parameters are the number of HARQ processes set to 16, the LDPC code-rate $k/n = 490/1024$ with maximum number of decoding iterations $k_{iter} = 6$. The selected modulation constellation is the 16-QAM one.
While for the simulations on PC we changed the number of N_{ant}^{TX} transmitting and N_{ant}^{RX} receiving antennas, for testing purposes over RPs we kept $N_{ant}^{TX} = 8$. For obvious reasons, i.e. due to the processing of 8 data streams, on a hand, this



(a) Computational Load - Splitted TX

(b) Computational Load - Splitted RX

Figure 3: (a) Pictorial representation of the TX processing chain splitted up by opt.6, 7.3, 7.2, 7.1 and 8. In violet, the segments showing the computational load for TX processing moved to the CubeSat constellation, in green, the segments showing the computational load for TX moved to the UAV fleet. Opt.8 is not depicted because it assigns all the load to the CubeSat's PUs. (b) Pictorial representation of the RX processing chain splitted up by opt.8, 7.1, 7.2, and 6. In violet, the segments showing the computational load for RX moved to the CubeSat constellation, in green, the segments showing the computational load for RX moved to the UAV fleet. Opt.8 is not depicted because it assigns all the load to the CubeSat's PUs.

was the more time-consuming scenario, on the other, it is feasible to assume to mount more antennas to take advantage from the spatial diversity.

To conclude this subsection, just a couple of words on the channel parametrization. We assumed to adopt a Clustered Delay Line channel model of type C (CDL-C), which is defined by 3GPP in Release 14 as a statistically realistic representation of a non-line-of-sight (NLOS) urban environment [32]. The delay spread is $\tau = 0.3\mu$sec. and the Doppler spread is $f_{shift} = 5$ Hz, that corresponds to a relative speed of 2.5 Km/h at a carrier frequency $f_c = 2.4$ GHz.

SWaP-C for system design

To dimension computational and energy resources to be hosted on communicating nodes is a non-trivial task. More in detail, t_{bsp} is a function of $\mathbb{R}$, thus $t_{bsp}(\mathbb{R}, IC)$, with IC the Instruction Count per program. The amount of resources to be made available on the UAV or CubeSat is dependent, on one side, on the targeted processing time, and on the other side, on the number of instructions per program to be run. In our case, IC varies with the performed splitting operations. As we choose lower splitting options, the IC will decrease. More precisely, $IC^{dynamic}_{split\,opt.}$ is the dynamic number of instruction per splitting option to be executed. The dynamic IC takes into account all those instructions repeatedly computed, for instance, within processing loops.

$$IC^{dynamic}_{split\,opt.} = IPC^{PU}_{split\,opt.} \cdot t^{PU}_{split\,opt.} \cdot f^{PU}_{CPU} \tag{4}$$

The instructions per cycle $IPC^{PU}_{split\,opt.}$ depends on the PU and the workload [33]. As shown in sect.4, we have measured it on hardware to retrieve $IC^{dynamic}_{split\,opt.}$. The PU processing time for the considered splitting option is $t^{PU}_{split\,opt.}$, while f^{PU}_{CPU} is the achievable PU clock frequency. Being $IC^{dynamic}_{split\,opt.}$ a constant, by inverting the equation and varying the objective processing time $t^{obj}_{split\,opt.}$, we estimate the required $IPC^{obj}_{split\,opt.}$ as follows:

$$IPC^{PU-obj}_{split\,opt.} = \frac{IC^{dynamic}_{split\,opt.}}{t^{PU-obj}_{split\,opt.} \cdot f^{PU}_{CPU}} \tag{5}$$

From here, it is straightforward to predict the needed $MIPS^{PU-obj}_{split\,opt.}$ with respect to the required processing time per splitting opt.

$$MIPS^{PU-obj}_{split\,opt.} = \frac{IPC^{PU-obj}_{split\,opt.} \cdot f^{PU}_{CPU}}{10^6} \tag{6}$$

Now, let's assume to parallelize the workload on PUs. Under such assumption, we obtain the $N^{PU-obj}_{split\,opt.}$ necessary PUs to be mounted and available, for instance, on a UAV

$$N^{PU-obj}_{split\,opt.} = \frac{MIPS^{PU-obj}_{split\,opt.}}{(MIPS^{PU}_{max})} \tag{7}$$

where $MIPS^{max}_{PU}$ is the PU maximum reachable MIPS value.

$$V^{PU-obj}_{split\,opt.} = N^{PU-obj}_{split\,opt.} \cdot V^{PU} \tag{8}$$
$$m^{PU-obj}_{split\,opt.} = N^{PU-obj}_{split\,opt.} \cdot m^{PU} \tag{9}$$
$$P^{PU-obj}_{split\,opt.} = N^{PU-obj}_{split\,opt.} \cdot P^{PU} \tag{10}$$
$$c^{PU-obj}_{split\,opt.} = N^{PU-obj}_{split\,opt.} \cdot c^{PU} \tag{11}$$

Finally, $N^{PU-obj}_{split\,opt.}$ allow us to provide guidelines regarding the volume $V^{PU-obj}_{split\,opt.}$ (m^3), weight $m^{PU-obj}_{split\,opt.}$ (kg), power $P^{PU-obj}_{split\,opt.}$ (W) and cost $c^{PU-obj}_{split\,opt.}$ (USD). (SWaP-C) metrics for the system design. V^{PU}, m^{PU}, P^{PU} and c^{PU} refers to the volume, mass, power consumption and cost of a single PU. SWaP-C is a common set of metrics for the optimization of hardware and software systems. The knowledge about SWaP-C is a priority for dimensioning complex systems like the one presented in this work.

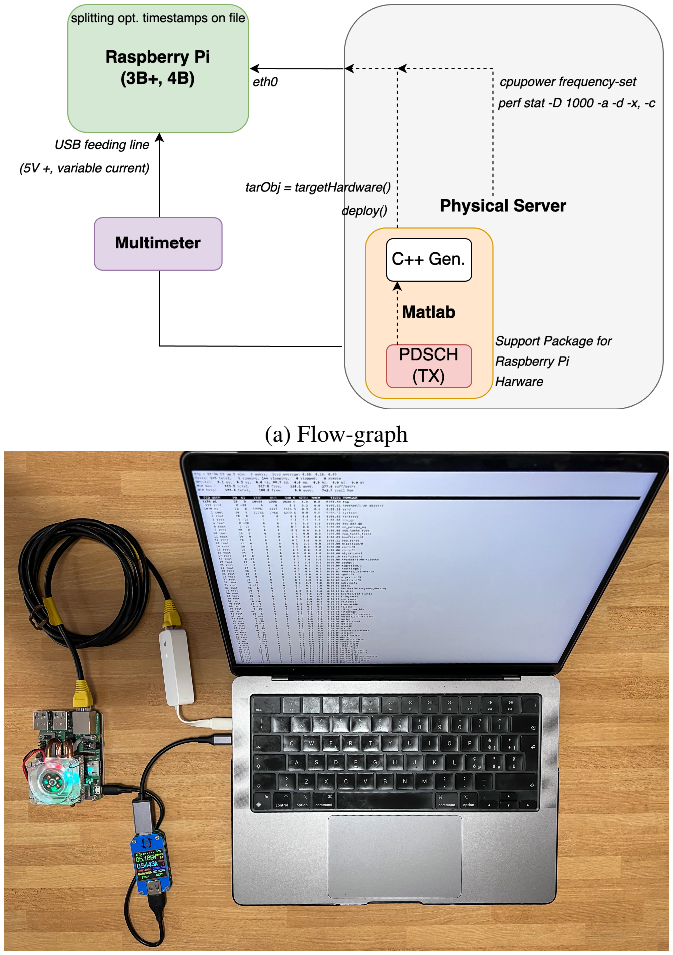
4. Experimental results

Experimentation strategy

In order to acquire reliable experimental results, we followed a top-down approach, moving from theory to simulations and concluding with hardware experimentation. Results discussion will analyze post-processed data gathered through

Table 1: Raspberry Pi Data Sheet

	3B+	**4B**
Processor	Broadcom BCM2837B0 (4 Cores)	Broadcom BCM2711 (4 Cores)
Clock Frequency (Operative) f_{CPU}^{w}	1.4GHz	1.5GHz
Clock Frequency f_{CPU}^{MIN}	0.6GHz	0.6GHz
Clock Frequency f_{CPU}^{MAX}	1.4GHz	2.1GHz
Mega Instructions Per Second (MIPS)	~ 527	~ 2037
RAM	1GB	2GB
Power Supply	5V/2.5A DC via micro-USB	5V DC via USB-C (min 3A*)
Temperature (Operative)	45°	50°
Power Consumption (400% CPU Load)	$5.1W$	$6.4W$
Weight	45g	46g
Volume	$80.92cm^3$g	$99.53cm^3$g



(a) Flow-graph

(b) Setup photo

Figure 4: (a) Setup flow-graph detailing the deployment of PDSCH functionalities from MATLAB over external hardware, which is controlled by the physical server through *ssh*, by converting the program into C++ code. (b) Setup photo showing the deployment of PDSCH functionalities over a Raspberry Pi 3B+ (Raspberry Pi 4B was used as well) through MATLAB® Support Package for Raspberry Pi™ Hardware installed on a Macbook Pro with Apple M1 Pro chip. The feeding line is connected by a Ruideng UM25 USB multimeter for power measurements. The RPs mount the ICE Tower Cooling Fan from S2Pi to cool down themselves during the overclock process.

simulations and test on RP 3B+ and 4B, and, to conclude, the SWaP-C analysis for implementing and operating splitting options on UAV's PUs within a bounded processing time t_{bsp}^{UAV}.

Experimental setup

The first step of the analysis is to simulate the DL-SCH and PDSCH processing chain on a 2021 Macbook Pro equipped with an Apple M1 Pro chip and a 16GB RAM. Our aim was to obtain, through these simulations, data about the achievable throughput with respect to the system parameter setting and the variation of the execution time depending on the considered splitting options and the number of antennas both at TX and RX sides. On the other hand, the analysis regarding the system dimensioning through SWaP-C metrics has been performed starting from low-cost PU architectures.

Two Raspberry Pi (RP) boards have been used as external PUs to test on hardware with the NR split functionalities. The details about these devices are shown in Tab. 1 [34] [35].

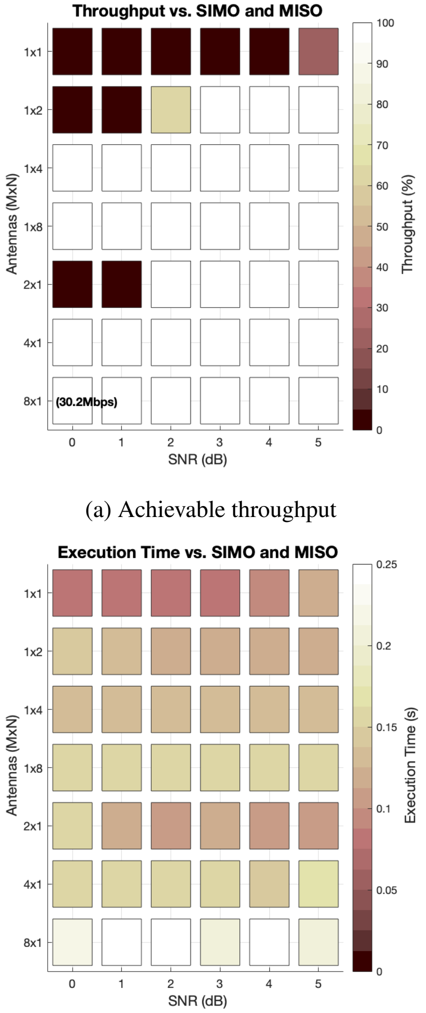
The RPs 3B+ and 4B mount an ICE Tower Cooling Fan from S2Pi, i.e. a liquid cooling, as shown in Fig. 4a. This was necessary to overclock the PUs and to avoid unexpected crashes or even damages to the electronic board. The RPs are fed by a micro-USB and a USB-C respectively, assuring an input voltage of about $5V$. A Ruideng UM25 USB multimeter is used to check the possible voltage swings. It allows to measure voltage with a time step of one second, as well as for the current flowing in the RPs, through a simple graphical user interface receiving data via Bluetooth. An Ethernet cable connects the PC with the RPs for data and Wi-Fi sharing. An Ethernet-to-USB-C adapter has been employed due to the lack of a Gigabit Ethernet port on the Macbook.

The program has been developed in MATLAB 2021b environment by exploiting the "5G Toolbox" and following 3GPP specifications about 5G NR. Deploying functions on Raspberry Pi is allowed thanks to the MATLAB Support Package for Raspberry Pi Hardware, which is a collection of functions to interface MATLAB and the RPs, and the *codegen* command. Through *codegen* is possible to generate C++ code starting from MATLAB code. Most of the functions are supported by *codegen* for the C++ code generation. The remaining functions have been ad-hoc customized to be exclusively composed of functions already supported by *codegen*.

RPs are controlled by the PC through *ssh* and overclocked by launching *cpupower frequency-set* with *–min* and *–max* field to fix the working clock frequency f_{CPU}^{w}. The timestamps for each splitting option are printed on text file and then post-processed. Information about the IC per program have been retrieved by exploiting *perf stat*, which allows to visualize the IPC depending on the workload. As it is difficult to isolate the IPC per splitting option, we averaged the gathered data between 25 realizations of the same test, running the whole DL-SCH and PDSCH chain per slot.

Software simulation results

The simulation results average data coming from the transmission of 5 slots. For each slot, the channel is re-initialized. First of all, we acquired the E2E throughput for the gNodeB-to-UE link, where the gNodeB is the binomial composed by Cubesat and UAV. For the CubeSat-to-UAV link we have considered a transparent channel. Indeed, this evaluation is simply aimed at visualizing the signal-to-noise ratio (SNR) improvement yielded by increasing N_{ant}^{TX} or N_{ant}^{RX} antennas numbers.



(a) Achievable throughput

(b) Execution time

Figure 5: (a) 5G NR PDSCH achievable throughput in percentage for MISO and SIMO systems. (b) 5G NR PDSCH execution time over an Apple M1 Pro chip MISO and SIMO systems.

By increasing the number of antenna units, the SNR required to assure the same throughput decreases, as clearly visible in Fig. 5a. Assuming the configuration described above - thus a maximum achievable throughput $R_b^{MAX} \approx 30.2Mbps$ - for single-input single-output (SISO) system and $SNR =$ 5dB, we reach a 20% of R_b^{MAX}, while by doubling the TX or RX antennas, we obtain an important gain of $3dB$ to achieve 100% of throughput performance. This is due to the spatial diversity guaranteed by multi-antenna MISO and SIMO systems.

As expected, we can verify, from Fig. 5b, that there is not any significant correlation between the execution time, i.e. $t_{bsp}^{UAV} = (t_f^{RX} - t_{opt.8}^{RX}) + (t_{opt.8}^{TX} - t_{init}^{TX})$, and the required SNR. We cannot verify the same for what concerns a higher number TX or RX antennas. Indeed, the amount of processing yielded by DL-SCH and PDSCH to the multiple streams transmitted by the physical antennas justifies a higher execution time t_{bsp}^{UAV}, and, consequentially a higher computational load $\mathbb{L}$ assuming $\mathbb{L} \propto t_{bsp}^{UAV}$. Again from Fig. 5b, it is remarkably noticed that the execution time for a 2×1 or 1×2 system is almost twice that of the SISO system. Moreover, looking at the 1×8 versus 8×1 configurations, the former configuration takes 69% of the time taken by the latter one to complete the simulation.

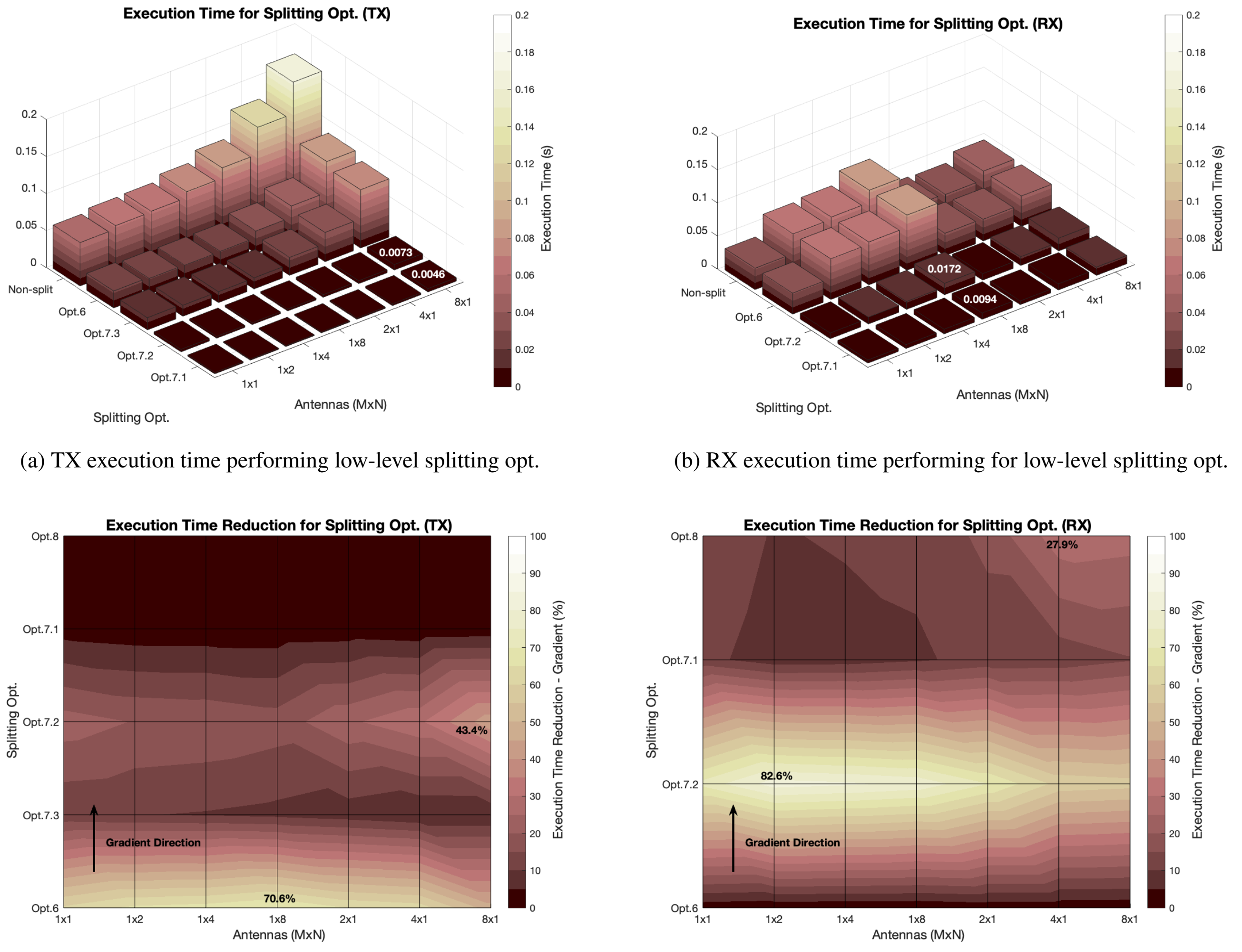
From Fig. 6, assuming a DL communication between the UAV-based RRH and an on-ground UE, it is important to notice how much the execution time increases as far as N_{ant}^{TX} increases, if we split the TX processing. Vice-versa, as N_{ant}^{RX} increases, the execution time ramps up to perform demodulation, demapping and decoding operations at RX side. As already mentioned above, the linear increase in the number of data streams (therefore in the baseband processing functionalities to be run) is the main cause of an increased execution time. Consequently, still considering the UAV-to-UE scenario, it appears that, as we add complexity to the TX, some splitting options become more prone to be exploited. In general, opt.6 yields, both for MISO and SIMO systems, to a $\sim 60 - 70\%$ reduction of the execution time with respect to not exploiting any splitting opportunities. However, this does not occur for an 8×1 configuration, where opt.6 carries a reduction of about 47%. Hence, opt.7.2 lowers down the processing latency of another 43.4%, as shown in Fig. 6d. This is due to precoding and mapping operations left at Cubesat side. The formed PDSCH grid is a 3D matrix, whose third dimension is equal to N_{ant}^{TX}. From here, we can deduce that the more N_{ant}^{TX} we mount on the communicating node, e.g. UAV, the more is the importance of opt.7.2, at least in terms of saved computational and energy resources at UAV side. Other reasoning in this sense should take care of the communication system dimensioning according to the midhaul data rate to be maintained between UAV and CubeSat [12].

The analysis provided here about RX split processing is made upon the use of DL-SCH and PDSCH. To estimate RX processing at UAV side, physical uplink shared channel (PUSCH) should have been implemented for the uplink communication. However, with a small degree of approximation, we assume DL-SCH and UL-SCH very similar in their basic functionalities, as stated in [36]. Thus, the evaluations made about DL-SCH and PDSCH apply also to UL-SCH and PUSCH.

From Fig. 6b, it is clear that an increase in N_{ant}^{RX} corresponds to a higher execution time t_{bsp}^{UAV}, and, consequently, to a heavier processing load. Apparently, opt.7.2 is the operation leading to the major variations of the execution time at UAV or CubeSat side. In particular, the execution time reduction ranges between $\sim 70 - 80\%$ for all the simulated configurations. The gradient ∇ increases also in correspondence of opt.8 for 2×1, 4×1 and 8×1 MISO systems. However, opt.8, which would set to zero the processing time at UAV side (Fig. 6a and Fig. 6b do not depict opt.8 for this reason), is mostly prohibitive for the data rate in the order of Gbps required to the fronthaul link [12]. To conclude this subsection, we should highlight that opt.7.3 is not present for RX splitting, because 3GPP standardized it only for DL transmission [21] [37].

On-hardware results

To perform on-hardware emulations, we have deployed the TX processing chain of a 8×1 MISO system on the external PUs, namely RP 3B+ and 4B. This choice was done to test the worst case scenario. As visible from the comparison of Fig. 6a with Fig. 6b, the 1×8 SIMO RX requires 49% less time than the 8×1 MISO TX to be executed.
The program runs over a single core, which led to a power



(a) TX execution time performing low-level splitting opt.

(b) RX execution time performing for low-level splitting opt.

(c) TX execution time (∇) performing low-level splitting opt.

(d) RX execution time (∇) performing low-level splitting opt.

Figure 6: (a) Execution time of a 5G NR TX in percentage over an Apple M1 Pro chip to perform opt.6, opt.7.3, opt.7.2, opt.7.1, opt.8. (b) Execution time of a 5G NR RX in percentage over an Apple M1 Pro chip to perform opt.6, opt.7.3, opt.7.2, opt.7.1, opt.8. (c) Discrete gradient (∇) showing the saved amount of computational load by performing opt.6, opt.7.3, opt.7.2, opt.7.1, opt.8 at TX side with respect to the whole PDSCH computation. (d) Discrete gradient (∇) showing the saved amount of computational load by performing opt.6, opt.7.3, opt.7.2, opt.7.1, opt.8 at RX side with respect to the whole PDSCH computation.

consumption of about $3.72W$ (multimeter computation). The task parallelization is assumed for the SWaP-C analysis. We followed the methodology presented in sect.4.1 to acquire data related to RP 3B+ and 4B performance.

In regards of splitting options 6, 7.3, 7.2 and 7.1, we evaluated in Fig. 7 the execution time on RPs against the clock frequency f^{w}_{CPU}. On the RP 3B+ and 4B, the execution time for the DL-SCH and PDSCH is in the order of seconds. More precisely, for a $f^{w}_{CPU} = 1$GHz, the processing power $\mathbb{P}^{UAV(RP\,3B+)}_{bsp}$ of a RP 3B+ allows to execute the whole series of functions within $t^{UAV(RP\,3B+)}_{bsp} = 4.86$ sec., while the RP 4B decreases the execution time to $t^{UAV(RP\,4B)}_{bsp} = 2.47$ sec. This is due to the augmented processing capabilities $\mathbb{P}$ of the RP 4B with respect to the 3B+, at the price of a higher power consumption as shown in Tab. 1. As in the previous simulation series, shown in Fig. 6c, opt.6 dramatically reduces the execution time to $t^{UAV(RP\,3B+)}_{opt.6} = 0.24$s and $t^{UAV(RP\,4B)}_{opt.6} = 0.22$s. An increase of the clock frequency f^{w}_{CPU}, while it solidly impacts on the execution time as clearly noticeable from Fig. 7, however, does not seem to cause a variation in the gradient ∇ characterizing the benefits of a splitting option with respect to another one.

The number of instruction per cycle (ICP) is directly related to the choice of the processor and to the clock frequency. Practically, it is retrieved through *perf-stat*, which we have launched on the overclocked RP 4B ($f^{w}_{CPU} = 1.5$GHz) through *ssh*. The achieved averaged value is $IPC^{RB\,4B}_{bsp} = 0.8152$, as shown in Fig. 8, which led to $IC^{dynamic}_{opt.6} = 159.6 \times 10^6$, $IC^{dynamic}_{opt.7.3} = 110.5 \times 10^6$, $IC^{dynamic}_{opt.7.2} = 68.4 \times 10^6$ and $IC^{dynamic}_{opt.7.1} = 40.2 \times 10^6$. By the way, reducing the number of functions to be operated straightforwardly involves a reduction of the information count per program, which largely justifies the important gap between $IC^{dynamic}_{opt.6}$ and $IC^{dynamic}_{opt.7.1}$. From here, we have searched for the MIPS reachable by the RP 4B and comparing them with those of Nvidia Jetson Nano to estimate the number of PUs

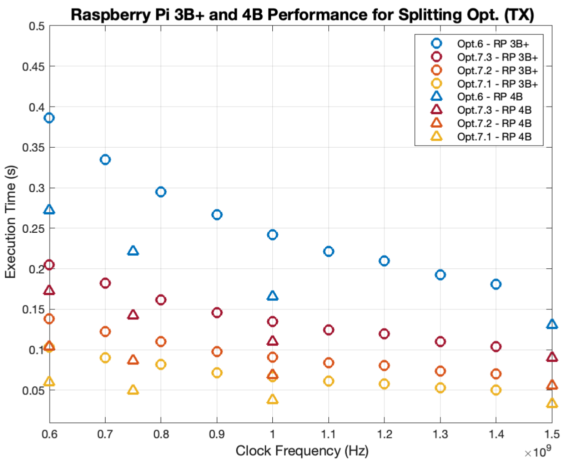


Figure 7: Execution time of a 5G NR splitted TX over a Raspberry Pi 3B+ and Raspberry Pi 4B vs. various clock frequencies $f_{CPU} = [0.6,\ 1.5]$GHz reached by overclocking the RPs.

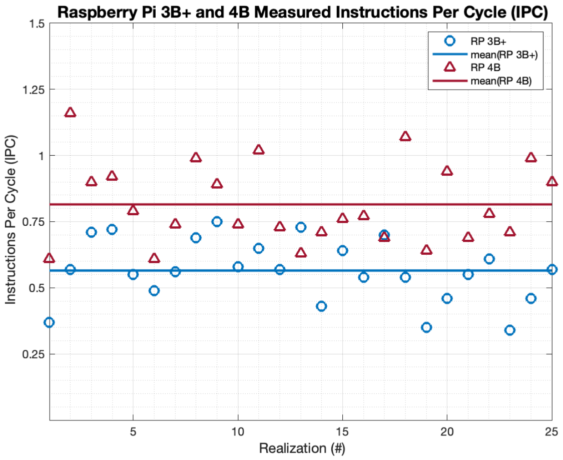


Figure 8: Measured IPC while running the whole DL-SCH and PDSCH chain on Raspberry Pi 3B+ and 4B.

needed on-board of the UAV to execute the splitting options within a time span $t_{bsp}^{UAV} = [0.5,\ 1.5]$ msec. This range was empirically selected, however, the slot duration is $t_{slot} = 0.5$ msec., thus it is feasible to assume an added processing time per slot of the same order of magnitude. Obviously, by accepting a higher execution time and keeping into account that, for generic applications, 5G commonly accepts $T_{radio} \approx 2msec.$, we will require a lower distance between UAV and CubeSat, and, consequentially, lower CubeSat lifetime, coverage and increased energy demand to correct the orbits. To better explore the topic, which is not the main core of this work, the following papers highlight all these trade-offs [11] [13].

In order to estimate the the SWaP-C metrics, we fix $MIPS^{RP\,4B} = 2037$ and $MIPS^{JN} = 3900$ [38] [39]. Although, these are not precise values as they come from benchmarking the PUs, we assume them as a feasible and easily achievable MIPS. Nvidia Jetson Nano is a small powerful PU whose envelope dimensions are $0.100 \times 0.080 \times 0.029$m. The maximum power consumption is around $10W$ and the list price for the "Jetson Nano Developer Kit" is 99$ with respect to 55$ for a RP 4B with 2GB of RAM [40]. The weight is 250g per board [41].

Finally, we shall point out how splitting options will impact on the system dimensioning. Indeed, size or volume, weight, power consumption and cost are strongly linked with the IC and t_{bsp} achieved per splitting option. This behavior is shown in Fig. 9 and Fig. 10. Requiring a tighter execution time turns into much more resources to be embarked on the UAV. Consequently, a system designer may think to assure enough computational and energy resources to perform one of the proposed splitting option. On average, by looking at the plots in Fig. 9d and Fig. 10d, we can say that in order to achieve the same system performance the required volume and the overall cost to host the RP 4B PUs or JN PUs on UAV will be almost the same. To be fair, costs depend on many factor. For instance, the latest chip shortage brought the cost of RPs to a 400% markup [42]. However, our aim is to provide a rough costs estimate. On the other hand, the weight to deploy RP-based or JN-based PUs on UAV is of particular interest. A single JN unit weighs the 543% more than the RP 4B. This is obviously reflected on the results in Fig. 9b and Fig. 10b. Assuring the processing capabilities $\mathbb{P}$ to perform opt.6 by RP 4B within $t_{opt.6}^{UAV(RP\,4B)} = 0.5$ msec. involves a weight $m \approx 7.5$ Kg, while $m \approx 20$Kg would be the added weight for a JN-based UAV, and $P_{opt.6}^{UAV(RP\,4B)} = 820$ W of power consumption with respect to $P_{opt.6}^{UAV(JN)} = 942$ W.

Once we highlighted the macro differences between RP-based and JN-based systems, it is important to notice how a system design would change upon a static splitting option choice at the UAV side. Let us consider to host JNs on UAV for their lower total power consumption and set a required slot processing time $t_{opt.6}^{UAV(JN)} = 0.5$ms. The required weight would be $m_{opt.6}^{UAV(JN)} = 20$Kg, as said before, and $m_{opt.7.3}^{UAV(JN)} = 14$ Kg for opt.7.3, $m_{opt.7.2}^{UAV(JN)} = 5.5$Kg for opt.7.2 and $m_{opt.7.1}^{UAV(JN)} = 2$Kg for opt.7.1. Roughly speaking, there is an order of magnitude between $m_{opt.7.1}^{UAV(JN)}$ and $m_{opt.6}^{UAV(JN)}$. The same applies for costs, where from more than $cost_{opt.6}^{UAV(JN)} \approx 8000$\$ we move to $cost_{opt.6}^{UAV(JN)} \approx 2000$\$, which for mass production is for sure a great saving. The volume decreases too as well as the power consumption, which from opt.6 and opt.7.1 is reduced of $\approx$ 600 W. The amount of saved energy resources would be then dedicated to the maximization of the flight time and, above all, service continuity.

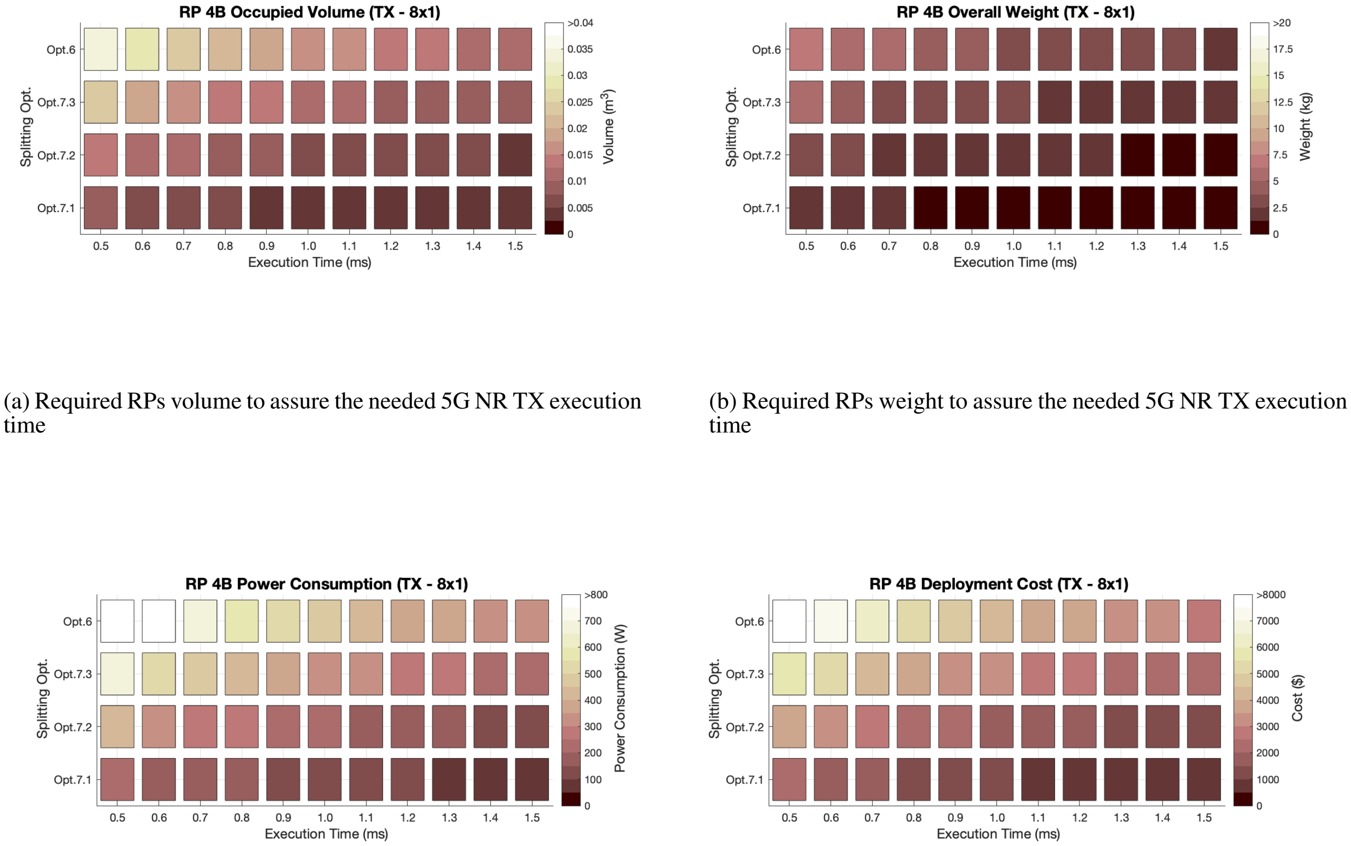
Results recap and design guidelines

In this subsection we aim at providing a recap of the experimental results achieved in the paper, along with some guidelines for the design of 3D networks. The main outcome of the analysis is that baseband processing time and computational load are the keys for the system optimization. We can summarize the recap as follows:

1. Splitting option 7.2 detaches in strongly uneven portions of the computational load both for downlink and uplink.

2. This behavior is particularly noticed for DL MISO 4x1 and 8x1 systems and for UL SIMO 1x2, 1x4 and 1x8.

3. RP 4B performs better in terms of execution time with



(a) Required RPs volume to assure the needed 5G NR TX execution time

(b) Required RPs weight to assure the needed 5G NR TX execution time

(c) Required RPs energy resources to assure the needed 5G NR TX execution time

(d) Required RPs cost to assure the needed 5G NR TX execution time

Figure 9: SWaP-C analysis for Raspberry PI 4B splitted 5G NR TX. (a) Volume required to host RPs 4B on nodes implementing C-RAN to perform splitting opt.6, opt.7.3, opt.7.2 and opt.7.1 at TX side. (b) Weight required to host RPs 4B on nodes implementing C-RAN to perform splitting opt.6, opt.7.3, opt.7.2 and opt.7.1 at TX side. (c) Power consumption to host RPs 4B on nodes implementing C-RAN to perform splitting opt.6, opt.7.3, opt.7.2 and opt.7.1 at TX side. (d) Approximated deployment costs to host RPs 4B on nodes implementing C-RAN to perform splitting opt.6, opt.7.3, opt.7.2 and opt.7.1 at TX side.

respect to RP 3B+, as expected.

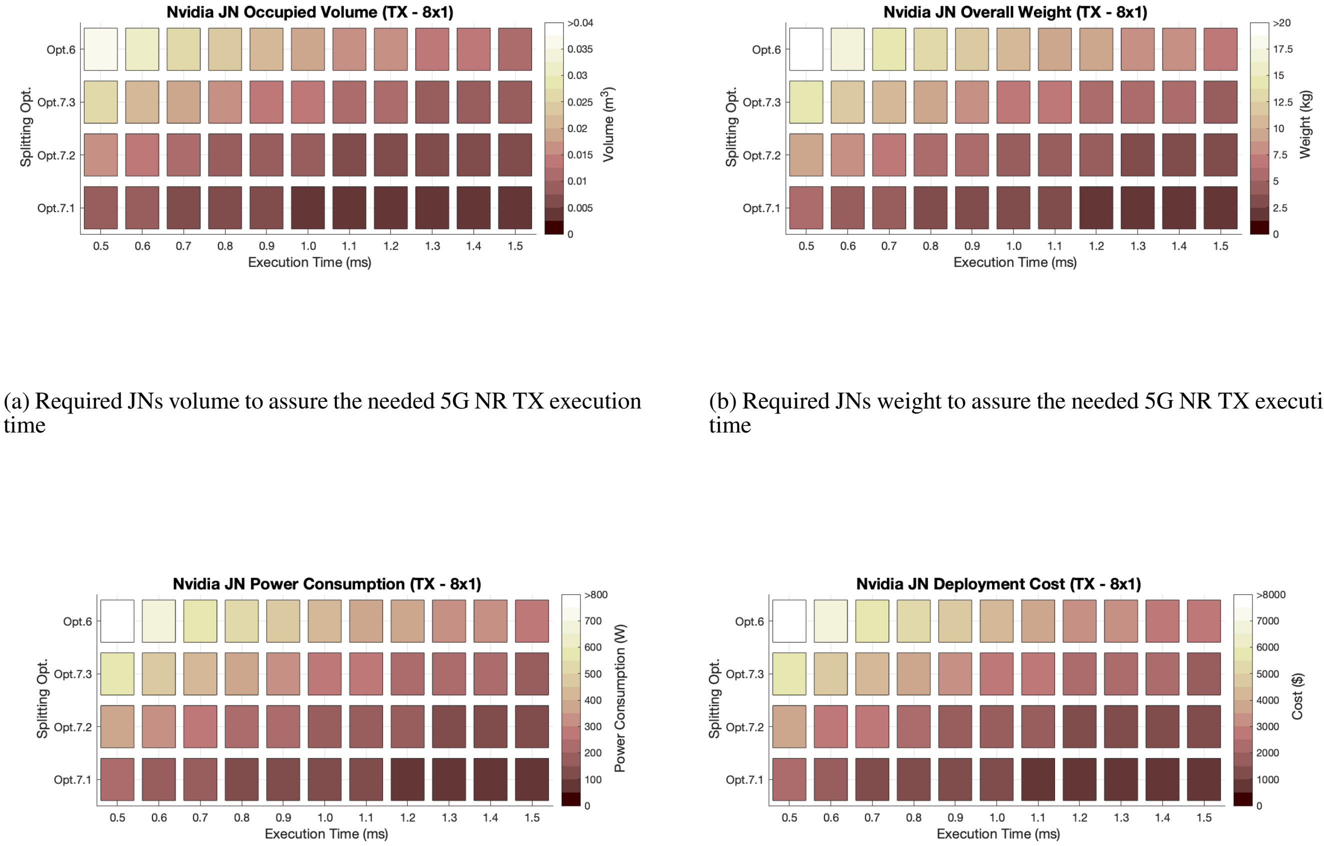
4. NVIDIA Jetson Nano could be more suitable as PU over UAVs than RP 4B+ due to reduced consumed power and deployment costs. On the other hand, RP 4B+ considerably reduces the overall equipment weight.

5. It should be worth framing this work within the Digital Twin concept [43], probably on of the most promising enabler of 6G technologies. The aim of this paper was to dimension the resources on board UAVs in the context of 3D Networks implementing the C-RAN. However, the methodology provided a simple way to design complex systems starting from digitally rendering the impact of processes on small, limited and low-cost hardware.

5. Conclusion

We studied the resource dimensioning of 3D networks implementing C-RAN from the UAV perspective. First, we detailed the impact of baseband processing on the radio transmission delay and E2E delay. Then, we addressed the issue concerning the UAV on-board resources required to execute the baseband functionalities. Later on, we discussed about splitting opportunities to save UAV-based RRH resources, while moving on the CubeSats, or other sky platforms, the major part of the computational load. Finally, we derived a methodology to predict the required resources, i.e. size weight power and cost (SWaP-C), on UAV to perform splitting options in an advantageous time window. Results firmly demonstrated the impact and benefits of baseband splitting operations over system design and performance.

Further works will consider the CubeSat point of view, thus providing guidelines about the design of satellite platform



(a) Required JNs volume to assure the needed 5G NR TX execution time

(b) Required JNs weight to assure the needed 5G NR TX execution time

(c) Required JNs energy resources to assure the needed 5G NR TX execution time

(d) Required JNs cost to assure the needed 5G NR TX execution time

Figure 10: SWaP-C analysis for Nvidia Jetson Nano splitted 5G NR TX. (a) Volume required to host JNs on nodes implementing C-RAN to perform splitting opt.6, opt.7.3, opt.7.2 and opt.7.1 at TX side. (b) Weight required to host JNs on nodes implementing C-RAN to perform splitting opt.6, opt.7.3, opt.7.2 and opt.7.1 at TX side. (c) Power consumption to host JNs on nodes implementing C-RAN to perform splitting opt.6, opt.7.3, opt.7.2 and opt.7.1 at TX side. (d) Approximated deployment costs to host JNs on nodes implementing C-RAN to perform splitting opt.6, opt.7.3, opt.7.2 and opt.7.1 at TX side.

able to efficiently execute all the functions relocated at its side. The literature lacks also of an integrated framework able to characterize 3D networks for 6G connectivity as a whole. Moreover, recent studies were only focused on point-to-point communications. In such a framework, the connectivity will be realistically guaranteed by fleets of UAVs and multi-orbit constellations of CubeSats. Trade-offs and dynamics, in this sense, should be adequately analyzed as, for instance, the handover strategies both in the aerial and space segment.

6. ACKNOWLEDGMENTS

We would like to thank M.SOFT Srl, represented by Michele Ermini, for having provided useful hardware for the tests exposed in this paper. The work is partially funded by the US National Science Foundation under the New Mexico SMART Grid Center - EPSCoR cooperative agreement Grant OIA-1757207 and a Faculty Fellowship awarded by DST NMICPS TiHAN at IIT Hyderabad and SERB Startup Research Grant (SRG-2021-001522).
The research activities presented in this paper fall within the field of interest of the IEEE AESS technical panel on Glue Technologies for Space Systems.

REFERENCES

[1] “5G by Ericsson,” https://www.ericsson.com/en/, February 2022, (Date last accessed on September 26, 2022).

[2] G. Liu, Y. Huang, N. Li, J. Dong, J. Jin, Q. Wang, and N. Li, “Vision, requirements and network architecture of 6g mobile network beyond 2030,” *China Communications*, vol. 17, no. 9, pp. 92–104, 2020.

[3] J. Zhu, M. Zhao, S. Zhang, and W. Zhou, “Exploring the road to 6g: Abc — foundation for intelligent mobile

networks," *China Communications*, vol. 17, no. 6, pp. 51–67, 2020.

[4] S. Chen, S. Sun, and S. Kang, "System integration of terrestrial mobile communication and satellite communication —the trends, challenges and key technologies in b5g and 6g," *China Communications*, vol. 17, no. 12, pp. 156–171, 2020.

[5] C. Sacchi, M. Ruggieri, K.-M. Cheung, M. Marchese, F. Granelli, V. Popescu, M. Rice, M. Murroni, N. Conci, C. Schlegel, T. Rossi, and M. Noble, "Glue technologies for space systems: An introduction to a new aess technical panel," *IEEE Aerospace and Electronic Systems Magazine*, vol. 35, no. 1, pp. 46–54, 2020.

[6] D. Wang, M. Giordani, M.-S. Alouini, and M. Zorzi, "The potential of multilayered hierarchical nonterrestrial networks for 6g: A comparative analysis among networking architectures," *IEEE Vehicular Technology Magazine*, vol. 16, no. 3, pp. 99–107, 2021.

[7] 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 and Tutorials*, vol. 21, no. 1, pp. 146–172, 2019.

[8] "Common Public Radio Interface (CPRI); Interface Specification," CPRI Consortium, Bengaluru, India, Tech. Rep., Jan. 2015.

[9] R. Bassoli, F. Granelli, C. Sacchi, S. Bonafini, and F. H. Fitzek, "Cubesat-based 5g cloud radio access networks: A novel paradigm for on-demand anytime/anywhere connectivity," *IEEE Vehicular Technology Magazine*, vol. 15, no. 2, pp. 39–47, 2020.

[10] S. Bonafini, R. Bassoli, F. Granelli, F. H. Fitzek, and C. Sacchi, "Virtual baseband unit splitting exploiting small satellite platforms," in *2020 IEEE Aerospace Conference*, 2020, pp. 1–14.

[11] S. Bonafini, C. Sacchi, R. Bassoli, F. Granelli, K. Kondepu, and F. H. Fitzek, "An analytical study on functional split in martian 3d networks," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–10, 2022.

[12] S. Bonafini, C. Sacchi, F. Granelli, R. Bassoli, F. H. Fitzek, and K. Kondepu, "3D Cloud-RAN Functional Split to Provide 6G Connectivity on Mars," in *2022 IEEE Aerospace Conference (AERO)*, 2022, pp. 1–13.

[13] S. Bonafini, C. Sacchi, R. Bassoli, K. Kondepu, F. Granelli, and F. H. Fitzek, "End-to-end performance assessment of a 3d network for 6g connectivity on mars surface," *Computer Networks*, vol. 213, p. 109079, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128622002171

[14] J. Yusupov, A. Ksentini, G. Marchetto, and R. Sisto, "Multi-objective function splitting and placement of network slices in 5g mobile networks," in *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*, 2018, pp. 1–6.

[15] B. S. Rawal, G. Manogaran, R. Singh, P. M, and M. Hamdi, "Network augmentation by dynamically splitting the switching function in sdn," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2021, pp. 1–6.

[16] Arzo, Sisay Tadesse, Scotece, Domenico, Bassoli, Riccardo, Barattini, Daniel, Granelli, Fabrizio, Foschini, Luca, and Fitzek, Frank H. P., "Msn: A playground framework for design and evaluation of microservices-based sdn controller," *Journal of Network and Systems Management*, vol. 30, no. 1, pp. 1573–7705, 2021.

[17] F. W. Murti, S. Ali, and M. Latva-aho, "Deep reinforcement based optimization of function splitting in virtualized radio access networks," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2021, pp. 1–6.

[18] S. T. Arzo, R. Bassoli, F. Granelli, and F. H. P. Fitzek, "Multi-agent based autonomic network management architecture," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3595–3618, 2021.

[19] E. Amiri, N. Wang, M. Shojafar, and R. Tafazolli, "Optimizing virtual network function splitting in open-ran environments," in *2022 IEEE 47th Conference on Local Computer Networks (LCN)*, 2022, pp. 422–429.

[20] B. M. Khorsandi, F. Tonini, E. Amato, and C. Raffaelli, "Dedicated path protection for reliable network slice embedding based on functional splitting," in *2019 21st International Conference on Transparent Optical Networks (ICTON)*, 2019, pp. 1–4.

[21] L. M. P. Larsen, A. Pruski, H. L. Christiansen, S. Ruepp, and M. S. Berger, "Xhaul latency dimensioning of 5g drone control," in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2022, pp. 762–771.

[22] A. Guidotti, A. Vanelli-Coralli, V. Schena, N. Chuberre, M. Jaafari, J. Puttonen, and S. Cioni, "The path to 5g-advanced and 6g non-terrestrial network systems," https://www.researchgate.net/publication/363843651_The_path_to_5G-Advanced_and_6G_Non-Terrestrial_Network_systems, 09 2022.

[23] R. Bassoli, C. Sacchi, F. Granelli, and I. Ashkenazi, "A virtualized border control system based on uavs: Design and energy efficiency considerations," in *2019 IEEE Aerospace Conference*, 2019, pp. 1–11.

[24] Geospatial Applications of Unmanned Aerial Systems (UAS), "Classification of the unmanned aerial systems," https://www.e-education.psu.edu/geog892/node/5, 2022.

[25] ITU, "Haps – high-altitude platform systems," https://www.itu.int/en/mediacentre/backgrounders/Pages/High-altitude-platform-systems.aspx, 2022.

[26] Cal Poly, "Cubesat design specification (1u – 12u) rev 14 cp-cds-r14," https://static1.squarespace.com/static/5418c831e4b0fa4ecac1bacd/t/62193b7fc9e72e0053f00910/1645820809779/CDS+REV14_1+2022-02-09.pdf, 2022.

[27] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, "Cloud ran for mobile networks—a technology overview," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 1, pp. 405–426, 2015.

[28] P. K. Agyapong, M. Iwamura, D. Staehle, W. Kiess, and A. Benjebbour, "Design considerations for a 5g network architecture," *IEEE Communications Magazine*, vol. 52, no. 11, pp. 65–75, 2014.

[29] 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 Communications Surveys and Tutorials*, vol. 20, no. 4, pp. 3098–3130, 2018.

[30] H. J. Son, S. M. Shin, "Fronthaul size: Calculation of maximum distance between rrh and bbu,"

https://www.netmanias.com/en/post/blog/6276/c-ran-fronthaul-lte/fronthaul-size-calculation-of-maximum-distance-between-rrh-and-bbu, 2014.

[31] HSE, “Drones with long flight times,” https://www.netmanias.com/en/post/blog/6276/c-ran-fronthaul-lte/fronthaul-size-calculation-of-maximum-distance-between-rrh-and-bbu.

[32] 3GPP, “Study on channel model for frequency spectrum above 6 ghz (3gpp tr 38.900 version 14.2.0 release 14),” https://www.etsi.org/deliver/etsi_tr/138900_138999/138900/14.02.00_60/tr_138900v140200p.pdf, 2017.

[33] Shaban, Muhammad, “Cpu performance evaluation,” http://meseec.ce.rit.edu/eecc550-winter2011/550-12-6-2011.pdf.

[34] L. Hattersley, “Raspberry pi 4 vs raspberry pi 3b+,” https://magpi.raspberrypi.com/articles/raspberry-pi-4-vs-raspberry-pi-3b-plus, 2019.

[35] Raspberry Pi Dramble, “Power consumption benchmarks,” https://www.pidramble.com/wiki/benchmarks/power-consumption.

[36] X. Lin, J. Li, R. Baldemair, J.-F. T. Cheng, S. Parkvall, D. C. Larsson, H. Koorapaty, M. Frenne, S. Falahati, A. Grovlen, and K. Werner, “5g new radio: Unveiling the essentials of the next generation wireless access technology,” *IEEE Communications Standards Magazine*, vol. 3, no. 3, pp. 30–37, 2019.

[37] 3GPP, “Study on new radio access technology: Radio access architecture and interfaces (release 14). r 38.801,” https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3056, 2017, (Date last accessed on July 23, 2021).

[38] R. Longbottom, “Raspberry pi 3b+ 32 bit and 64 bit benchmarks and stress tests,” 09 2018.

[39] R. Graves, “Benchmarking the brand new nvidia jetson nano: 4gb, usb 3, $99!” https://www.sevarg.net/2019/04/07/benchmarking-nvidia-jetson-nano/, 2019.

[40] A. Lele, “Nvidia jetson nano is a $99 raspberry pi rival for ai development,” https://itsfoss.com/nvidia-jetson-nano/#:~:text=NVIDIA%20Jetson%20Nano%20is%20a%20%2499%20Raspberry%20Pi%20Rival%20for%20AI%20Development&text=At%20the%20GPU%20Technology%20Conference,the%20Jetson%20Nano%20Developer%20Kit., 2020.

[41] K. Yurkova, “Make the most of your jetson’s computing power for machine learning inference,” https://deci.ai/blog/jetson-machine-learning-inference/#:~:text=You%20can%20buy%20a%20Jetson,250%20g%20(~9%20ounces)., 2021.

[42] L. Pounder, “Raspberry pi 4 in short supply, being scalped at 400% markup (updated),” https://www.tomshardware.com/news/raspberry-pi-4-supply-issues, 2022.

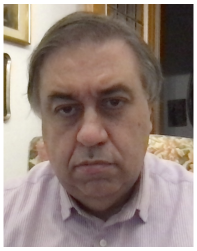
[43] H. X. Nguyen, R. Trestian, D. To, and M. Tatipamula, “Digital twin for 5g and beyond,” *IEEE Communications Magazine*, vol. 59, no. 2, pp. 10–15, 2021.

BIOGRAPHY

Stefano Bonafini *received his B.S. in Electronics and Telecommunications engineering and his M.S. in Information and Communications engineering at the University of Trento, Italy. He worked as a post-graduate researcher at the Department of Information Engineering and Computer Science (DISI) of University of Trento. He is currently enrolled in the Ph.D. program at the ICT School of University of Trento. His studies concern interplanetary and aerospace communication systems with a particular focus on reconfigurable softwarized networking and “Towards 6G” connectivity.*

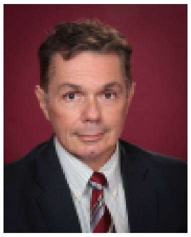
Sisay Tadesse Arzo *is a PostDoc Fellow at the University of New Mexico, USA. He got his Ph.D. in Information and Communication Technology and MSc. in Telecommunication Engineering from the University of Trento, Italy. He received his BSc. from Hawassa University, Ethiopia. He has more than five years of industrial experience in the Telecom industry. He has more than ten journals and papers in cloud computing, network softwarization, and network automation. His research interest included Network Softwarization, Internet of Things(IoT), Network Automation, SmartGrid, Quantum Computing, and Communication and Computing in Space.*

Koteswararao Kondepu *is an Assistant Professor at India Institute of Technology Dharwad, Dharwad, India. He obtained his Ph.D. degree in Computer Science and Engineering from Institute for Advanced Studies Lucca (IMT), Italy in July 2012. His present research is focused on 5G, softwarization and virtualization of mobile network and communication networks reliability.*



Claudio Sacchi *received the “Laurea” Degree in Electronic Engineering, and the Ph.D. in Space Science and Engineering at the University of Genoa (Italy) in 1992 and 2003, respectively. From 1996 to 2002, he has been research cooperator with the University of Genoa, Dept. of Biophysical and Electronic Engineering (DIBE) and with the National Italian Consortium in Telecommunications (CNIT), managing project activities in the field of multimedia surveillance systems and satellite communications. Since December 2020, Dr. Sacchi is associate professor at the Department of Information Engineering and Computer Science (DISI) of the University of Trento (Italy). Claudio Sacchi is author and co-author of more than 110 papers published in international journals and conferences. In 2011, he was guest editor of the special issue of PROCEEDINGS OF THE IEEE: “Aerospace Communications: History, Trends and Future.” Moreover, in 2015, he was guest editor of the featured-topic special issue of IEEE COMMUNICATIONS MAGAZINE: “Toward the Space 2.0 Era.” Since*

May 2019, Dr. Sacchi is coordinator of the IEEE AESS technical panel: "Glue Technologies for Space Systems" (the panel has been awarded by AESS in 2020). The research interests of Dr. Sacchi are mainly focused on wide band mobile and satellite transmission systems based on space, time and frequency diversity; MIMO systems; array processing; multi-rate and multi-access wireless communications; EHF broadband aerospace communications; software radio and cognitive radio; radio communications for emergency recovery applications. Claudio Sacchi is a senior member of IEEE and a member of the IEEE ComSoc, IEEE Broadcast, IEEE VT and IEEE AESS society.



Michael Devetsikiotis (FIEEE) *received the Diploma degree in electrical engineering from the Aristotle University of Thessaloniki, Greece, in 1988, and the M.S. and Ph.D. degrees in electrical engineering from North Carolina State University, Raleigh, NC, USA, in 1990 and 1993, respectively. He joined the University of New Mexico, Albuquerque, NM, USA, in July 2016, as a Professor and the Chair of the ECE Department, School of Engineering. His work has received well over 7,000 citations. In 2017, he was inducted to the NC State ECE Alumni Hall of Fame. His research work has resulted in 50 refereed journal articles, 130 refereed conference papers, and 61 invited presentations, in the area of design and performance evaluation of telecommunication networks, complex socio-technical and cyber–physical systems, efficient simulation, and smart grid communications.*

Fabrizio Granelli *Fabrizio Granelli was a Visiting Professor at the State University of Campinas, São Paulo, Brazil, and at The University of Tokyo, Japan. He is currently an Associate Professor with the Department of Information Engineering and Computer Science (DISI), University of Trento, Italy. From 2012 to 2014, he was an Italian Master School Coordinator in the framework of the European Institute of Innovation and Technology ICT Labs Consortium. He was the ComSoc Director of Online Content from 2016 to 2017. He was the Dean of education of the DISI Department for the period 2015–2017 and a coordinator of the research and didactical activities on computer networks within the degree in telecommunications engineering. From 2018 to 2019, he was the IEEE ComSoc Director for Educational Services. He was an advisor of more than 80 B.Sc. and M.Sc. theses and eight Ph.D. theses. He is the author or coauthor of more than 200 papers published in international journals, books, and conferences in networking, with particular reference to performance modeling, cross-layering, wireless networks, cognitive radios and networks, green networking, and smart grid communications. He was the Chair of the IEEE ComSoc Technical Committee on Communication Systems Integration and Modeling (CSIM TC). He is the Chair of the IEEE Communications Society Technical Committee on Transmission, Access and Optical Systems (TAOS TC). Since January 2017, he has been the Associate Editor-in-Chief of the IEEE Communications Surveys and Tutorials journal (IF=22—the HIGHEST among all computer science and telecommunications publications!). He is an Area Editor of the IEEE Transactions on Green Communications and Networking. He was an IEEE ComSoc Distinguished Lecturer for the period 2012–2015.(Based on document published on 9 August 2021).*