On the impact of novel function mappings, sharing policies, and split settings in network slice design

Wesley da Silva Coelho, Amal Benhamiche, Nancy Perrot, Stefano Secci

▶ **To cite this version:**

Wesley da Silva Coelho, Amal Benhamiche, Nancy Perrot, Stefano Secci. On the impact of novel function mappings, sharing policies, and split settings in network slice design. International Conference on Network and Service Management, Nov 2020, Izmir, Turkey. hal-02942693v1

HAL Id: hal-02942693

https://hal.science/hal-02942693v1

Submitted on 18 Sep 2020 (v1), last revised 8 Jan 2021 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On the impact of novel function mappings, sharing policies, and split settings in network slice design

Wesley da Silva Coelho
Cnam & Orange Labs, France
wesley.dasilvacoelho@orange.com

Amal Benhamiche
Orange Labs, France
amal.benhamiche@orange.com

Nancy Perrot
Orange Labs, France
nancy.perrot@orange.com

Stefano Secci
Cnam, France
stefano.secci@cnam.fr

***Abstract*—In this work, we model the network slice provisioning as an optimization problem including novel mapping and provisioning requirements rising with new 5G radio and core function placement policies. We propose an MILP-based formulation that joins different functional splitting strategies with different network function sharing policies and novel mapping continuity constraints from 5G specifications. We show by numerical simulations the impact of taking into full and partial consideration these peculiar sets of novel technical constraints.**

***Index Terms*—network slicing; functional split; sharing policy.**

I. Introduction

5G mobile systems are deployed in their first phase in 2020 worldwide, covering mostly at this stage the virtualization of core network functions. In the following 5G phase 2 stage [1], the technological evolution will touch the whole network environment, going from the cellular and IoT radio access to the application service architectures. This transition will challenge slice network design since multiple resources and segments, today managed independently from each other, are to be operated with continuity in resource allocation and provisioning. This whole and unique service is called in the 5G specifications Communication Service (CS) and might be associated with different providers running on the same physical network at the access, core, and application segments. Because of different bitrate and latency requirements, policies on radio access function splitting are going to have an impact on the backhauling network dimensioning, and therefore on the placement of core network functions and on the configuration of edge computing application servers. Moreover, different policies for control versus data-plane function sharing and scaling are expected to be applied. Indeed, services can be of three classes of service [2] - enhanced Mobile Broadband (eMBB), Ultra-Reliable Low Latency Communications (URLLC), and massive Machine Type Communications (mMTC) - differing in the requirements, such as maximum latency, minimum availability, and bandwidth; to provide the necessary flexible provisioning, Network Function Virtualization [3], Software Defined Networking [4], and Network Slicing [5] technologies can be adopted to let the CS provider deploy its services on top of logical networks, named Network Slices.

On the state-of-the-art, optimization approaches related to network slicing mostly consider it either as Virtual Network Embedding [6], Function Placement and Routing [7], or Service Function Chaining [8] problems. Addressing end-to-end network slicing, however, requires to consider different physical and virtual network topologies, each with specific technical constraints and particular orchestration rules. Furthermore, an important novelty of 5G specification is the introduction of three novel mapping dimensions influencing the placement and interconnection of slices and network functions: (i) a CS can be delivered by multiple Slices; (ii) Slices can be decomposed into Slice Subnets; (iii) Network Functions can be decomposed into Network Function Services. While the first mapping requirement can simply impact network design hyperparameters only, the second and third ones come with new technical constraints to guarantee a coherent provisioning of each CS. Namely, the capacity to support specific behaviors for all the components of the same slice, such as function splitting, sharing, and scaling policies. In addition to these peculiar constraints, classical network function embedding, routing, and placement with requirements on latency and network and computing capacities hold as well.

In this paper, we formally define the network slice design problem as a comprehensiveframework that (i) takes into consideration the above mentioned new mapping dimensions, and (ii) models the relationship between flexible radio access functional splitting, control-plane and data-plane function isolation, and core network function placement. Also, our model is compliant with technical specifications published by the 3rd Generation Partnership Project (3GPP) [2], [9], [10]. Even though several works partially cover the network slice design problem [11], [12] and related sub-problems, such as functional split mode selection [13]–[15], network slicing with VNF sharing [16]–[18], and network slicing with VNF scaling [19]–[23], no attention has been given to address jointly all aforementioned aspects in order to design network slices and understand the impact of mapping, sharing and split policies.

II. Background

We draw 5G mapping requirements and taxonomy, and present policies appearing in 5G systems in terms of sharing policies and functional splitting in radio access.

A. 5G System Mapping Requirements

The 3GPP specifications [2], [9], [10] present entities of 5G systems; as we describe in [24], they are: User Equipment (UE), Communication Service (CS), Network Slice (NS),

Acknowledgement: This work is funded by the Agence Nationale de la Recherche (ANR) MAESTRO-5G (ANR-18-CE25-0012) project.

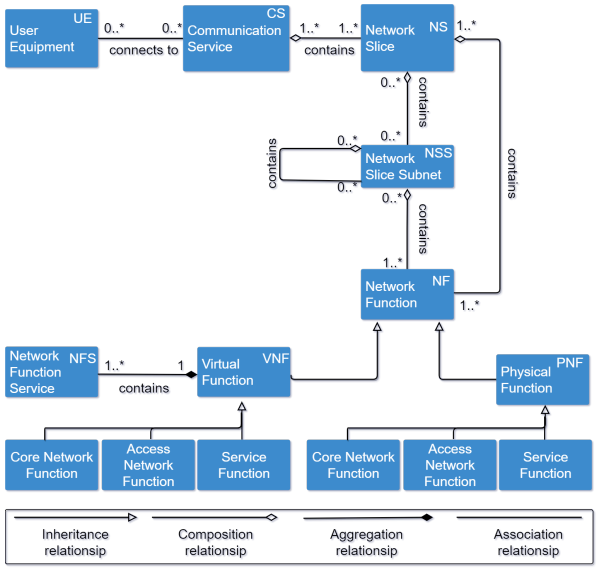

Fig. 1: Interactions between 5G entities: Unified Modeling Language (UML) class diagram. Source: [24].

Network Slice Subnet (NSS), Network Function (NF), NF Service (NFS). Fig. 1 depicts the interactions between these entities. A UE might be connected to several CSs, which might run on one or more customized NSs. Additionally, each NS might be composed of one or more NSSs, which might also be composed of lower-layer NSSs. A simple example of this scenario is given by considering an NS composed of an access NSS and a core NSS, where the latter can, in turn, be composed of a control-plane NSS and a data-plane NSS (data-plane relates to user application traffic while control-plane traffic involves network and service signaling functions). In this nested architecture, each NS or NSS is composed of one or more NFs attached to the Access Network (AN; e.g., Scheduler Function and Connection Mobile Control Function) or to the Core Network (CN; e.g. Session Management Function and Access and Mobility Management Function), or representing a Service Function (e.g., Firewall, Proxy, and Load Balancer). Finally, at the lowest level, each virtual NF is composed of a set of NFSs. This implies that some NFs can directly communicate with each other by request/response and subscribe/notify application-level signaling hitting NFSs. Note that one NF can be virtualized (VNF) or physical (PNF).

As presented in [24], we can distinguish five mapping levels for creating a complete 5G virtual network. Besides the UE to CS mapping (a user can use concurrently multiple CSs hence multiple NSs), we also have:

1) Mapping NFSs to NFs: this is needed to minimize the allocation of resources for each NF. Intuitively, the larger the NF's set of NFSs is, the more physical resources are required to install it. Therefore, a solution to this mapping problem provides the minimum set of NFSs composing each NF.

2) Mapping NFs to Slices and Slice Subnets: This mapping level decides the sub-set of NFs that should be present in each NS as well as the connection between them. Additionally, since each NF has its traffic processing capacity demand, the number of each NF instances by type within a slice should be dimensioned. At this level, NFs are jointly mapped to NSSs.

3) Mapping Slice Subnets to Slices: This level creates NSs from well-defined NSSs. This can be the case when a Core NS is created from two NSSs, e.g. one composed of control-plane NFs and one of data-plane NFs - from the 3GPP's point of view, these two sub-sets of functions are considered as NSSs and the whole virtual environment as an NS.

4) Mapping Slices to CSs: Depending on the heterogeneous needs and the expected data rate throughput in the service, each CS can be mapped into a subset of NSs. In this context, matching techniques can be used to better identify which NSs are the most appropriate to deliver a CS. Also, this level of mapping can be done with active (already deployed) NSs.

It is important to stress that the decomposition of NSs into NSSs and of NFs into NFSs is motivated by scalability and efficiency reasons. Indeed, one NS can be deployed in a scalable manner thanks to the segmentation of a slice into multiple NS subnets; and the overall computing demand can be decreased by allocating resources to NF micro-services rather than to macro NF units.

B. Sharing policies

Given the expected data volume and the treatment capacity of each NF, it is important to predict how many instances of each function type should be installed for each network slice. Moreover, dimensioning strategies have to model how NFs relate to different slices. For instance, isolation is a key aspect for network slicing, and dedicated NFs might be necessary to ensure that each NS operates independently. This approach is important for preventing the incorrect balance of resources between the served NSs. Additionally, security is another crucial point in virtual environments. To ensure security and data routing control, partially or completely isolated network slices with dedicated NFs might be implemented. Hence, isolation constraints might be applied on the virtual layer; NFs installed in the same physical node must be dedicated to a virtual network serving a specific client, thus cannot be shared by two or more NSs. On the other hand, sharing NFs among different NSs can be an interesting strategy to simplify the virtual environment implementation and to reduce redundancies throughout the network [10]. We assume that an NF can treat data from two or more NSs if they allow it.

We depict in Fig. 2 six possible NF sharing policies that, based on our analysis, are possible as of 3GPP specifications; in this illustration, a DP block can refer to data-plane functions for both access and core segments. They are as follows:

1) *Flat Sharing*: all CSs share the same virtual network; it can be an interesting strategy when different slices have no isolation constraints and show similar technical constraints in terms of latency and availability.
2) *Hard Isolation*: the isolation is complete, each CS has its own virtual network.
3) *Shared Control-Plane*: slices share the same Control-Plane (CP) while having their dedicated Data-Planes (D-DPs). It may be a solution for NSs requiring low end-

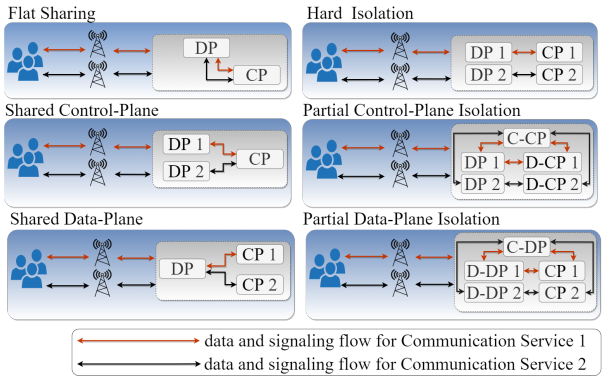

Fig. 2: NF sharing policies.

to-end latency; in this scenario, DP equipment should be deployed as close as possible to UEs, which has, therefore, an impact on the level of functional splitting.

4) *Partial Control-Plane Isolation*: only a part of the CP, called common CP (C-CP), is shared by two CSs; a CP portion and entire DPs of each CS are dedicated.
5) *Shared Data-Plane*: CSs share the same Data-Plane while having their own and dedicated Control-Planes (D-CPs).
6) *Partial Data-Plane Isolation* case: only a part of the DP is shared by two CSs, named common DP (C-DP); a DP portion and entire CPs of each CS are dedicated.

According to 3GPP specifications, these settings are in practice adaptable to multiple CSs. In addition, other configurations might be proposed to guarantee Service Level Agreements.

C. Functional Splitting in the Radio Access Network

Flexible Radio Access Network (RAN) splitting [15] is a technique meant to increase network efficiency leveraging NFV flexibility. To overcome the redundancy throughout the network, centralized RAN (C-RAN) was first introduced in 2011 [25]; pools of Baseband Units with large capacity, now called Centralized Units (CUs), are proposed to treat the data flow from a sub-set of Remote Radio Unit, now named Distributed Units (DUs). Hence, a fundamental task is to define the functionalities enabled locally at the DUs, or centrally at the CUs. Fig. 3 illustrates different functional split options on the 4G stack, as the 5G RAN split options have not yet been specified. Let us take option 3 as an example: all functions from Radio Frequency (RF) to Low Radio Link Control (RLC) blocks are locally installed, while high RLC, Packet Data Convergence Protocol (PDCP) and Radio Resource Control (RRC) functions are centrally installed. Equivalently, with option 7 on the uplink direction, all functionalities after low Physical (PHY) block are installed at a CU, while with option 5 all entities before low Media Access Control (MAC) block are installed at the DUs. Since the functional split was originally meant to be made *a priori* (i.e., before deploying the network) choosing the best split [26] for each scenario is not trivial. Indeed, defining the distributed and centralized functions should consider end-to-end delay and total bandwidth constraints on each physical path connecting DUs to CUs while optimizing the resource allocation.

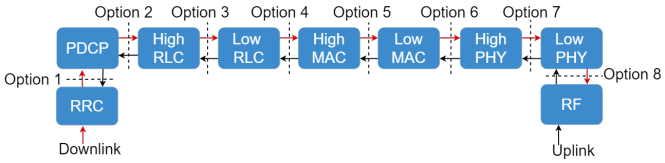

Fig. 3: Different functional split options.

Table I depicts different front-haul (FH) bitrates and latency indicators for each functional split. The bitrates are calculated as in [26] for a scenario using 100 MHz bandwidth and 32 antenna ports, while the maximum accepted one-way latency through FH is proposed by 3GPP [27]. Note first that highest bitrates and lowest latency are imposed by option 8. However, one of the advantages of choosing this split would be in reducing the number of NFs throughout the access network, as they would be installed centrally and shared by different DUs. Contrarily, option 1 requests low bitrates and admits higher latency; the disadvantage of this option is that almost all NFs would be installed locally - this scenario demands higher computational power on each DU, which could be impractical given the number of expected DUs in 5G systems. It is also important to point out the difference between downlink (DL) and uplink (UP) bitrates due to physical layer operations.

SDN and NFV technologies can be used together with C-RAN to offer flexibility to split RAN slice subnets [2], [27]. To this propose, two classes of RAN functions are proposed by [28]: *asynchronous network functions* and *synchronous network functions*; the former refers to network functions that process data asynchronously with the radio interface and demand low data rates. State transition and handover preparation are functionalities from RRC and PDCP blocks that are candidates to be virtualized, centralized into CUs pools, and shared by a sub-set of DUs. However, time-synchronous functions, such as interference coordination, scheduling and power control from PHY and MAC blocks, process data synchronously with the radio interface, requiring low latency and high data rate: these NFs might need some hardware acceleration, which implies that they are good candidates to either be implemented as dedicated machines or installed on a path that assures low latency and high bandwidth. According to [29], strict timing dependency between protocol layers must be avoided, using instead asynchronous NFs as much as possible to grant more flexibility to RAN slicing.

Being consistent with [2], [27], [29], we incorporate flexible RAN splitting to design end-to-end network slices. This approach can better deal with the heterogeneous requirements

TABLE I: Front-haul bitrate and latency in functional split.

Functional Split	DL Bitrate	UP Bitrate	FH Latency
Option 1: RRC-PDCP	4 Gbps	3 Gbps	10 ms
Option 2: PDCP-hRLC	4 Gbps	3 Gbps	1.5-10 ms
Option 3: hRLC-lRLC	4 Gbps	3 Gbps	1.5-10 ms
Option 4: lRLC-hMAC	5.2 Gbps	4.5 Gbps	0.1-1.0 ms
Option 5: hMAC-lMAC	5.6 Gbps	7.1 Gbps	0.1-1.0 ms
Option 6: lMAC-hPHY	5.6 Gbps	7.1 Gbps	0.25 ms
Option 7: hPHY-lPHY	9.2 Gbps	60.4 Gbps	0.25 ms
Option 8: lPHY-RF	157.3 Gbps	157.3 Gbps	0.25 ms

of each NS request while decreasing the redundancy in the network, that is, minimizing the number of virtual AN-based functions installed throughout the physical network. [13]–[15] address the challenges of flexible functional split schemes in order to optimize the allocation of physical and virtual resources. [15], for example, proposes a new architecture which introduces a flexible split of RAN functionalities between the Cloud-RRH, an edge cloud, and the central cloud. [13], in turn, analyze the technical features of the network in order to find the optimal split for different scenarios; the authors considered the configuration of the base stations, the fiber ownership, and the data transmission direction. They demonstrated that lower total cost of ownership can be achieved with optimal functional split compared to classical radio access networks.

III. Problem definition

We define our network slice design problem.

A. Physical layer model

We associate with the physical layer a directed graph $G_p = (V_p, A_p)$ where V_p is the set of nodes and A_p the set of arcs. V_p is composed of disjoint sub-sets, V_p^{du}, V_p^{ac}, and V_p^{ap}, containing the distributed unities, aggregation and core servers, and application nodes, respectively. Every node $u \in V_p$ is associated with a capacity c_u corresponding to the number of available CPUs. Moreover, an arc $a = (u, v) \in A_p$ corresponds to a physical link connecting nodes u and $v \in V_p$. We denote by $\delta^+(u)$ (resp. $\delta^-(u)$) the sub-set of arcs going from (resp. to) node $u \in V_p$. Finally, each arc $a \in A_p$ has a bandwidth capacity denoted b_a, and a latency value d_a expressing the time needed by a flow to traverse a.

B. Virtual layer model

The virtual layer is modeled as a set of directed graphs corresponding to network slices. Every NS is composed of one or more network slice subnets with different network functions, which, in turn, are composed of a specific set of NFSs. In this work, we define an NSS as any sub-set of network functions shared among the same group of NSs

1) Network Function Services: We denote by F the set of different NFS types. F is composed of the sub-set F^d of data-plane NFSs, the sub-set F^c of control-plane NFSs, and an auxiliary dummy function f_0, in such a way that $F^d \cup F^c \cup \{f_0\} = F$ and $F^d \cap F^c \cap \{f_0\} = \emptyset$ hold. Regarding the uplink direction, F^d is an ordered set composed of data-plane NFSs from both access and core networks. Every NFS type $f \in F$ is associated with a capacity denoted c_f corresponding to the minimum number of CPU needed to run one of its copies within an NF. Also, every NFS $f \in F$ is associated with a traffic processing capacity $cap(f)$, expressed in Mbps, and an expected data rate b_f within a physical node given one UE connected to the related slice. We denote by $b_{fg} \geqslant 0$ the total amount of traffic generated between NFSs f and g given one UE connected to the related NS. Additionally, we denote by $d_{fg} \geqslant 0$ the maximum accepted delay[1] between NFSs f and g. For every $f \in F^d$, we denote by λ_f the compression coefficient on the data-plane traffic flow related to the initial volume sent by the origin node of any traffic request. Lastly, all aforementioned parameters related to the auxiliary dummy function f_0 are set to 0, except the compression coefficient λ_{f_0}, which is equal to 1.

2) Network Functions: We denote by N the set of network functions available to pack NFS copies. An NF $n \in N$ might gather several NFS copies, potentially of different types. In our model, NFs are uncapacitated entities with no resource requirements other than those demanded by the hosted NFSs.

3) Network Slice Requests: The set of network slice requests is denoted by S. Each request $s \in S$ is associated with a binary parameter α_f^s that takes value 1 (resp. 0) if an NFS type $f \in F$ is (resp. is not) required to be present in the final associated virtual network. We denote by $G_s = (V_s, A_s)$ the final directed graph associated with $s \in S$, with V_s being the set of virtual nodes representing the sub-set of NFs (and the hosted NFSs) serving the given slice, and A_s being a set of arcs connecting two nodes from V_s. To represent the isolation requirements on the virtual layer, we denote by q_{fg}^{st} the binary parameter that takes value 1 (resp. 0) if slice request $s \in S$ admits (resp. does not admit) packing an NFS of type $f \in F$ with an NFS g from slice request $t \in S$ in the same NF. In addition, every request $s \in S$ is associated with a set $K(s)$ of traffic demands to be routed in the physical layer. Each demand $k \in K(s)$ is defined by a pair (o_k, t_k), being the origin and the destination nodes of k. For any k, $o_k \in V_p^{du}$ and $t_k \in V_p^{ap}$. We denote by $O(s)$ the set of origin nodes of all traffic demand from K(s), by b_k the initial data rate sent by node o_k, in Mbps, and d_s the maximum end-to-end latency for all traffic demands in $K(s)$[2]. Finally, we denote by n_s the expected number of UEs that are to be connected to slice s.

C. Problem Statement

We define our Network Slice Design Problem (NSDP) as follows. Given a directed graph G_p representing the physical network, a set of slice requests S, a directed graph G_s, a set of traffic demands $K(s)$ associated with each request $s \in S$, and a set of available NFS types denoted F, the NSDP consists in determining the number of NFSs to install on the nodes of G_s for each $s \in S$ and the size of NF hosting them, so that:

- $K(s)$ demands can be routed in G_s using these NFs,
- the NFs installed on G_s can be packed into the NFs while satisfying the isolation constraints,
- a path in G_p is associated with each pair of NFs installed,
- the total cost is minimum,
- all technical constraints imposed by both physical and virtual layers are respected.

Fig. 4 depicts an example of solution for an instance with 2 NS requests, 5 demands (e.g., $K(s_2) = \{(u_{23}, u_{16}), (u_2, u_{16})\}$), 7 NFS types , 8 NFs, 3 NSSs, and a

[1]This is important when flexible splitting is applied on the radio access; the selected split must respect the maximum fronthaul latency limitations.

[2]We assume that uplink and downlink flows follow the same physical path and are treated by the same DP NFSs, in a reverse order related to each other. Due to this assumption and for the sake of simplicity, in our model, we take into consideration only the uplink direction on the data-plane flow.

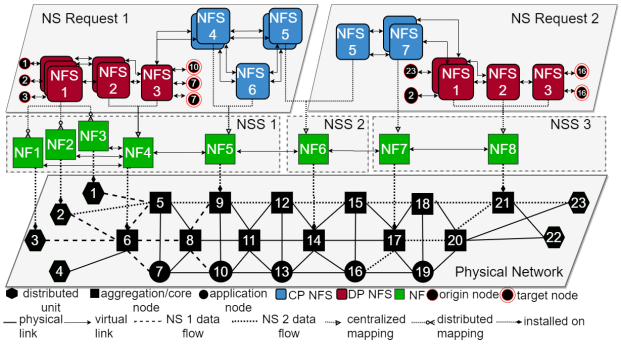

Fig. 4: Example of a solution for a NSDP instance.

physical network. Note that a different number of copies of the same NFS type is required to be installed for each slice (e.g. NFS 2). In addition, copies of NFS 1 from slice 1 are installed locally, while all other NFSs are centralized. Also, copies of NFS 5 are packed into NF6 and shared by both network slices. Finally, the traffic flow from each slice request is routed through the related NSSs and then in the physical network: regarding the traffic demand (u_3, u_7) of slice 1, its virtual DP flow is routed through the virtual link (NF1, NF4), while the related physical path is made on physical links (u_3, u_6) and (u_6, u_7). It is worth mentioning that, since the sub-sets of NFs shared among different slices are not known in advance, each NSS is an abstraction made in post-processing.

IV. Mathematical Programming Formulation

Table II summarizes the model decision variables.

A. Constraints

1) Split Selection: Ineq. (1) set whether a NFS f serving a slice s is distributed or centralized. Since F^d is an ordered set, all related NFSs on the same side of the selected split must be installed accordingly, either locally or centrally.

$$z_f^s \leq z_{f+1}^s \quad ,\forall s \in S, \forall f \in F^d \backslash \{f_{|F^d|}\} \tag{1}$$

2) NFS Placement: Given a set $K(s)$, constraints (2) ensure that distributed NFSs are installed on related origin nodes; we assume that NFSs from CP cannot be installed in a distributed manner. Constraints (3), in turn, ensure that centralized NFS instances are installed in the same physical node.

$$\sum_{n \in N} x_{nu}^{sf} = \begin{cases} 1 - z_f^s & , \text{ if } f \in F^d, u \in O(s); \\ 0 & , \text{ otherwise .} \end{cases} \quad ,s \in S, \forall f \in F, u \in V_p^{du} \tag{2}$$

TABLE II: Decision variables

Variable		Type
z_f^s	1, if function f from slice s is centralized; 0 otherwise.	Binary
x_{nu}^{sf}	1, if NFS f from slice s is packed into NF n and installed on physical node u; 0 otherwise.	Binary
w_{nu}^{sf}	amount of NFS f serving slice s, packed in NF n and installed on physical node u.	Real
y_{nu}^{f}	total number NFSs of type f packed into NF n and installed on physical node u.	Integer
γ_{fg}^{ka}	1, the traffic demand k uses arc a to route the flow between NFSs f and g; 0 otherwise.	Binary

$$\sum_{n \in N} \sum_{u \in V_p \backslash V_p^{du}} x_{nu}^{sf} = \begin{cases} z_f^s & , \text{ if } f \in F^d; \\ \alpha_f^s & , \text{ otherwise .} \end{cases} \quad s \in S, \forall f \in F \tag{3}$$

3) NF dimensioning: (4) calculate the exact amount of distributed centralized NFSs for each NS request. It is important to mention that, to minimize the residual virtual resources from each NFS, this amount might be a fractional value; regarding the sharing possibilities, these values are rounding up with inequalities related to packing and capacity constraints.

$$cap(f) w_{nu}^{sf} = \begin{cases} \lambda_{f-1} b^k x_{nu}^{sf} & , \text{ if } f \in F^d, u \in V_p^{du} \\ n_s b_f x_{nu}^{sf} & , \text{ if } f \in F^c; \\ \sum\limits_{k \in K(s)} \lambda_{f-1} b_k x_{nu}^{sf} & , \text{ otherwise .} \end{cases} \quad ,\forall s \in S, \forall f \in F, \forall n \in N, \forall u \in V_p \tag{4}$$

4) NFS Packing: (5) represent the isolation constraints on virtual layer, ensuring that two different NFSs are not packed into the same NF if it is prohibited; these constraints are responsible of applying different mapping policies (see Fig. 2) imposed by each NS request. (6), in turn, calculate the number of copies of each NFS type packed in each NF and on physical nodes. Finally, constraints (7) ensure that a given NF copy is not enabled at more than one physical node.

$$x_{nu}^{sf} + x_{nu}^{tg} \leq 1 + q_{fg}^{st} q_{gf}^{ts} \quad ,\forall s,t \in S, u \in V_p, n \in N, f,g \in F \tag{5}$$

$$\sum_{s \in S} w_{nu}^{sf} \leq y_{nu}^{f} \ ,\forall n \in N, \forall v \in V_p, \forall f \in F \tag{6}$$

$$x_{nu}^{sf} + x_{nv}^{tg} \leq 1 \ ,\forall s,t \in, f,g \in F, n \in N, u,v \in V_p : v \neq u \tag{7}$$

5) Capacity, routing and latency constraints: Because of space limits, we do not detail in the following the capacity, routing and latency constraints, commonly present in NF placement problems. We provide in [30] the full formulation.

B. Formulation

We minimize the total cost of deploying all network slice requests. To this end, the objective is to share as many NFSs as possible while respecting physical capacity constraints and assuring QoS imposed by each slice request. Being Ω the scaling coefficient related to link utilization, the NSDP is then equivalent to the following formulation:

$$min \sum_{f \in F} \sum_{n \in N} \sum_{u \in V_p} y_{nu}^{f} + \Omega \sum_{a \in A_p} \sum_{s \in S} \sum_{k \in K(s)} \sum_{f,g \in F} \gamma_{fg}^{ka} \tag{8}$$

subject to (1)-(7), capacity, routing, and domain constraints.

While the first term in (8) is related to the number of installed functions, the second one refers to the number of active links. The factor Ω can be used to drive toward the desired outcome, e.g., to emphasize the number of NFSs over the number of links in the design optimization.

V. Numerical Results

We detail the simulation setting and then expose the results.

TABLE III: Simulated slice demand setting

Slice	Service required	Additional CP NFSs	Max E2E latency d_s	UE data rate	UE per DU
1	Broadband access in dense areas	NFS10, NFS11	10ms	300Mbps	600
2	Ultra-low cost broadband	-	10ms	10Mbps	600
3	Real-time communication	NFS11, NFS12, NFS13	1ms	25Mbps	180
4	Video broadcast	NFS10, NFS11	100ms	200Mbps	60

1) Physical topologies: We simulated different physical networks with different features. Inspired by real access networks, we first propose a specific topology called *Mandala* (Fig. 5a) with the following structure: given *n* DU nodes, we have *n/4* aggregation nodes, *n/4* core nodes, and *n/8* application nodes. Each DU node is connected to two aggregation nodes, which, in turn, are connected to two inner-level core nodes. Each core node is additionally connected to two application nodes. Finally, given two different nodes u and v, there exists one arc (v, u) for each arc (u, v). We name this network topology *Mandala*. Fig. 5a shows this topology where *n* is equal to 16. For sake of clarity, each pair of arcs between two nodes is represented by an edge.

In our simulation, while application nodes have no available capacity (considered only as sink nodes), each one of DU, aggregation, and core nodes has 30 16-CPU servers; this capacity corresponds to 12.5% of the global computing demand (with no function sharing) and enables to test all split settings and sharing policies. In addition, fronthaul links (between DUs and aggregation nodes), backhaul links (between aggregation and core nodes), and core links (between core and application nodes) have capacities b_a set to respectively 100%, 200% and 300% of the maximum flow sent by a single DU at the split setting with the highest bitrate. Finally, to simulate a small region, the latency d_a on each arc randomly takes a value between: 50μs and 100μs, 200μs and 300μs, and 400μs and 600μs, for fronthaul, backhaul and core links, respectively.

We also run our tests on two different physical topologies: one binary tree-based structure (hereinafter referred to as *Tree*; Fig. 5b) with 31 nodes and 60 arcs, and *Sun* from SNDlib [31] composed of 27 nodes and 102 arcs (Fig. 5c). We mapped the 16 DUs to all 16 leaves and the nodes composing the external ring path in the former and latter structures, respectively; aggregation, core, and application nodes were randomly mapped in both topologies. While the capacities on physical nodes follow the same parameter values in Mandala, the bandwidth on links from the Tree structure was set to 500% of the maximum flow sent by a single DU at the split setting with the highest bitrate; the latency is between 50μs and 100μs. For the Sun topology, these values were randomly chosen between 50μs and 600μs for the latency whereas the bandwidth values were set between 100% and 300% of the maximum flow sent by a single DU.

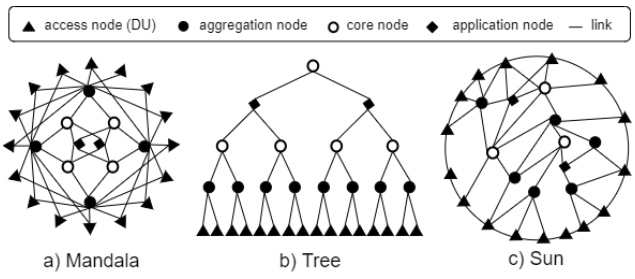

Fig. 5: Physical network structures: examples with 16 DUs.

2) Virtual layer: We set F^d with five DP NFS types: NFS1 represents functions of the MAC bloc; NFS2 corresponds to functions of the RLC block; NFS3 acts as the PDCP bloc; NFS4 represents functions from RRC bloc; NFS5 corresponds DP functions from the core network[3]. In addition, there are four mandatory CP NFS types (labeled NFS6..NFS9) and other four optional CP NFS types (labeled NFS10..NFS13; examples of mandatory and optional 5G core NFs are presented in [24]). Each NFS has a processing capacity set to 100% of the average volume sent by all DUs. Furthermore, the resource c_f required to install each copy of them is set to approximately 5% of the average capacity available on physical nodes. Also, the traffic generated from or to any CP NFS was set to 1 kbps per UE.

According to the 4G functional split levels reported in Table I and considering the uplink direction, we set similar compression coefficients λ_f related to initial volume sent by a traffic demand: 65% for NFS1 and 40% for the other CP NFSs. Additionally, the acceptable latency d_{fg} between two DP NFSs also follows those in Table I, taking the upper bound when an interval is proposed. Finally, the latency d_{fg} involving any CP NFS is set not to exceed 500μs; this value corresponds to 5% of the total CP latency proposed by 3GPP [2].

3) Slice requests: We tested instances with 4 NS requests, each with 8 traffic demands with random origin-destination pairs; for each $k \in K(s)$, origin o_k is a DU while destination t_k is an application node as previously discussed. Additionally, all network slices must contain all DP NFSs, four mandatory CP NFSs, and a different set of additional NFSs that can be required (see Table III). We assume that all CP NFSs are connected to each other. Furthermore, to simulate the communication between data and control planes, there exists an expected traffic volume between CP NFSs and DP ones on each related network slice; we create such traffic from NFS6 (e.g., corresponding to the AMF in 5G core [24]) to all DP NFSs. To also observe the impact of different sharing policies on the number of distributed NFSs, 25% of available DUs are set to be an origin node of all NS requests; application nodes are evenly distributed as target nodes. Finally, each slice request imposes different technical constraints related to end-to-end latency d_s, demands for optional CP NFSs, and expected user experienced data rate. As depicted in Ta-

[3]Since RF and PHY blocs have synchronous network functionalities that pose extremely strict latency requirements, we assume they are integrated to each DU as PNFs. Hence, they are not considered in our virtual DP chains.

ble III, we applied the assumptions proposed by [32] for each aforementioned requirements. In our simulations, slice request 1 represents an eMBB application with an important traffic volume, which impacts both virtual and physical capacities. Slice request 3, in turn, represents an URLLC application, imposing a strict end-to-end DP latency, which restrains the placement possibilities of the related NFSs. The other two slice requests are intermediate regarding both aforementioned parameters.. Finally, each DU is associated with a flow rate equal to the product between the expected number of UE per DU and their related data rate in such NS.

4) Scenarios: Following Fig. 3, each scenario represents one combination of functional split setting and sharing policy applied to all slices. While different sharing policies are those previously presented (see Fig. 2), the split settings impose different sets of distributed and centralized DP NFSs. Table IV summarizes the tested scenarios. We applied additional constraints to impose the desired split setting to all slice demands. Also, sharing policies were imposed by changing the q_{fg}^{st} parameters values used in Ineq. (5).

To scale with the complexity of the formulation[4], all instances were generated using Mandala, Sun, and Tree structures with 16 DUs. Finally, we run 10 tests on each topology varying both traffic demands' origin and destination nodes. We implemented our model in a Julia-JuMP environment using ILO CPLEX 12.8 as the linear solver. We set Ω to an enough small value (i.e., 10^{-3}) in (8) to emphasize the number of NFSs over the number of links in the optimization process. The data-set and the code are available on [33].

5) Results: The goal of the following numerical analysis is to assess the impact of novel mapping, splitting, and sharing policies on the network. Fig. 6 reports the number of distributed and centralized NFSs on different sharing policies and split strategies. While distributed entities are only NFSs from DP, centralized ones also aggregate NFSs from CP; translucent bars show the total number of installed NFSs. Note first that the generated instances' characteristics are such that:

- the minimum (resp. maximum) number of NFSs required to serve all NS requests is equal to 101 (resp. 227);
- split setting 1 requires the largest number of NFS copies in all proposed sharing policies;
- Hard Isolation has the greatest number of NFSs copies on all split settings, including the flexible one.

In our simulations, having isolation constraints on different sets of NFS types led to different impacts on the network slice design. Regarding the five first split settings, Shared DP and Partial DP policies provided a mean decrease (resp. increase) of 28% (resp. 42%) on the number of distributed (resp. centralized) NFSs compared to Shared CP and Partial CP. Also, flexible splitting proves to be an interesting strategy even for scenarios that have strong isolation restrictions. With roughly 56% as overall reduction, this approach has the smallest number of NFSs in all mapping scenarios; regarding each sharing policy, the average reduction was roughly 38% (standard deviation equals to approximately 10%) compared to split setting 1. It is important to note that, since we minimize the total number of NFSs, flexible split always had the same number of NFS copies than split setting 6, which provided the greatest number of centralized NFSs. This behavior might differ if the NS provider is interested in optimizing other parameters, such as the load on physical arcs. As seen in Fig. 7, applying different sharing policies and split settings has also an important impact on the physical network. It is worth mentioning that we excluded from the computation of the average load the unused links and nodes, and for sake of readability, the standard deviations are not depicted; the observed ratio of the standard deviation to the mean were always less than 14%. First, we observe that split settings 6 and Flexible have the worst impact on link load in all mapping approaches, requiring up to 100% of the capacity on the most loaded link (see Fig. 7e); the average load on backhaul and core (resp. fronthaul) links was equal to 52% (resp. 40%) applying Flat (resp. Shared DP) sharing policy and split setting 6 (see Figures 7c and 7d). This behavior is expected since all NFSs are installed centrally and the data volume sent

[4]The time needed to achieve the best solution increases exponentially with the size of the instance. For instance, even with small topologies (13 physical nodes), the problem with 24 functions could not be solved within 3 hours.

TABLE IV: Scenarios: split settings and sharing policies

Split Setting	Description
Setting 1	all DP NFS are installed locally for all NS requests.
Setting 2	for each slice, only NFS5 is installed centrally.
Setting 3	for each slice, only NFS4 and NFS5 are installed centrally. It correspond to 3GPP's split 1 in Fig. 3.
Setting 4	for each slice, only NFS1 and NFS2 are distributed; it corresponds to 3GPP's split 2 in Fig. 3.
Setting 5	for each slice, only NFS1 is installed locally. It corresponds to 3GPP's split 4 in Fig. 3.
Setting 6	all DP NFSs are installed centrally for all NS requests. It corresponds to 3GPP's split 6 in Fig. 3
Flexible	free functional split selection for each NS request.
Policy	**Description**
Hard Isolation	NS requests do not accept sharing any NFS.
Shared DP	only DP NFSs can be shared among slices.
Shared CP	only CP NFSs can be shared among slices.
Partial DP Isol.	only NFS1, NFS2, and NFS3 can be shared.
Partial CP Isol.	only mandatory CP NFSs can be shared among NSs.
Flat Sharing	NS requests do not impose any isolation constraint.

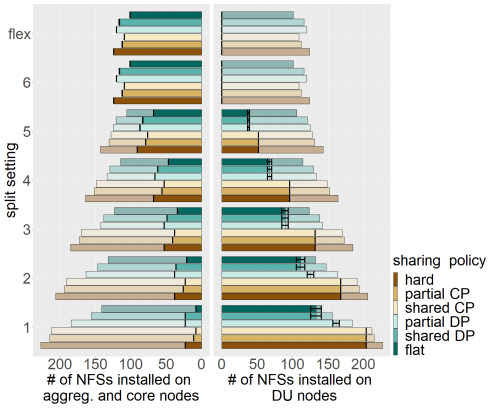

Fig. 6: Number of NFSs on different scenarios.

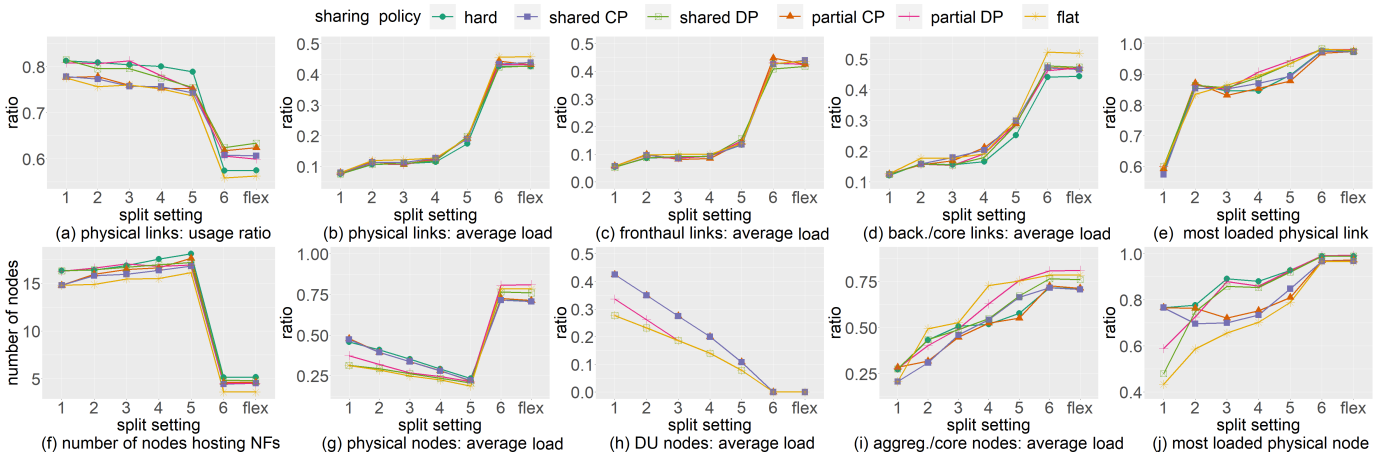

Fig. 7: Impact of different split settings and sharing policies on the physical network.

by each traffic demand is completely decompressed before traversing the fronthaul links. On the other hand, split setting 1 benefits of the impact of the compressed data and demands the least amount of capacity on the links in all mapping approaches, requiring at most 60% of the capacity on the link on average. However, as shown in Fig. 7a, this split requires the largest number of links (between 77% and 82%) since the CP and DP NFSs are far from each other. Besides, CP NFS6 must be connected to all related slice's distributed DP NFSs.

We also note a strong impact of different scenarios on physical nodes. Since there exist at least one NFS type installed locally, the first five functional splits had the largest number of physical nodes hosting at least one NF (see Fig. 7f). We also observe a decrease of the average load on physical nodes (see Fig. 7g), in particular on DU nodes (see Fig. 7h), on all sharing policies. However, due to the completely decompressed data arriving in the centralized DP chain, a shift of behavior is observed when split setting 6 is applied (see Fig. 7g). Unlike physical links and aggregation and core nodes (see Fig. 7b and Fig. 7i, respectively), DU nodes benefit of functional splits where a greater number of NFSs is installed centrally. The average load on DU (resp. aggregation and core) nodes decreased (resp. increased) from roughly 43% (resp. 21%) applying split setting 1 along with Partial CP (resp. Shared CP) sharing policy to approximately 8% (resp. 75%) applying split setting 5 jointly with Flat sharing policy; the most loaded physical node (see Fig. 7j) provided 98% (resp. 43%) on average of its available resource applying split setting 6 (resp. setting 1) and Partial DP (resp. Flat) sharing policy.

Even with a negative impact on the number of installed NFSs, mapped links, and nodes (see Figures 6, 7a, 7f, respectively), Hard Isolation could partially unload the physical network. In fact, due to strong isolation constraints, this sharing policy demanded less physical capacity from links (see Fig. 7b) and from aggregation and core nodes (see Fig. 7i) in some split settings. Consequently, a short physical path for each traffic demand was prioritized, leading to the use of physical nodes and links not mapped to other traffic demands. Also, let us recall that the final solutions prioritized minimizing the number of NFSs, even if this approach harms the load of the physical network; to bring the final solution closer to its economic strategy, the NS provider can simply modify the objective function (8) to a more suitable one. It is also worth mentioning that, to test feasible instances of all functional split settings, we set a low enough latency for each physical link; otherwise, some split settings (e.g. settings 5) could be impossible. Finally, since we imposed the same scenario (see Table IV) to all slice requests, we did not observe a significant difference in the results using distinct physical topologies. For instance, comparing the three topologies, the difference in the number of physical nodes hosting an NF, the ratio of active links, and on the number of NFS copies were always less than 7%, 11%, and 1%, respectively. This behavior might be different in real scenarios since NS requests are likely to impose different isolation constraints and physical networks might not have enough capacity to allow all split settings (due to the relation between the fronthaul capacity and the NFSs' compression coefficient). This, therefore, reinforces the importance of applying flexible functional splitting while considering different sharing policies in virtual environments.

VI. Concluding Remarks

In this paper, we modeled the 5G network slice provisioning as an optimization problem including novel mapping and provisioning requirements. In particular, we considered novel 5G-specific mapping dimensions, modeling the relationship between flexible radio access functional splitting, control-plane and data-plane function separation, and sharing policies. Our model is compliant with 3GPP specifications, and we demonstrated by simulation the impact of taking into full and partial consideration of the peculiar constraints rising with 5G systems. For instance, we reported numerical results showing that flexible splitting appears as a key factor to deal with heterogeneous requirements to deploy distinct CSs, leading to considerable network slice cost decrease. In our simulations, the number of NFSs needed to deploy the virtual networks could be reduced by up to 56% depending on which of the six proposed sharing policies is applied to each network slice.

REFERENCES

[1] "3rd Generation Partnership Project. Release 16: 5G system - phase 2," https://www.3gpp.org/release-16, Accessed: 2020-07-24.

[2] 3rd Generation Partnership Project, "3GPP TR 38.913 V14.3.0; Study on scenarios and requirements for next generation access technologies ," 2017.

[3] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, "Network function virtualization in 5G," *IEEE Comm. Mag.*, vol. 54, no. 4, pp. 84–91, 2016.

[4] T. Chen *et al.*, "Software defined mobile networks: concept, survey, and research directions," *IEEE Comm. Mag.*, vol. 53, no. 11, pp. 126–133.

[5] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5G: Survey and challenges," *IEEE Comm. Mag.*, vol. 55, no. 5, pp. 94–100, 2017.

[6] A. Fischer *et al.*, "Virtual network embedding: A survey," *IEEE Comm. Surveys & Tutorials*, vol. 15, no. 4, pp. 1888–1906, 2013.

[7] B. Addis, D. Belabed, M. Bouet, and S. Secci, "Virtual network functions placement and routing optimization," in *2015 IEEE 4th International Conference on Cloud Networking (CloudNet)*. IEEE, 2015, pp. 171–177.

[8] D. Bhamare, R. Jain, M. Samaka, and A. Erbad, "A survey on service function chaining," *J. of Network and Computer Applications*, vol. 75, pp. 138–155, 2016.

[9] 3rd Generation Partnership Project, "3GPP TS 23.501 V15.4.0: System Architecture for the 5G System ," 2018.

[10] ——, "3GPP TR 28.801 V15.1.0: Study on management and orchestration of network slicing for next generation network ," 2018.

[11] A. Baumgartner, T. Bauschert, A. M. Koster, and V. S. Reddy, "Optimisation models for robust and survivable network slice design: A comparative analysis," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–7.

[12] B. Tan *et al.*, "Analog coded softcast: A network slice design for multimedia broadcast/multicast," *IEEE Transactions on Multimedia*, vol. 19, no. 10, pp. 2293–2306, 2017.

[13] X. Wang *et al.*, "Centralize or distribute? a techno-economic study to design a low-cost cloud radio access network," in *ICC*. IEEE, 2017.

[14] A. Maeder *et al.*, "Towards a flexible functional split for cloud-ran networks," in *2014 EuCNC*. IEEE, 2014, pp. 1–5.

[15] O. Chabbouh, S. B. Rejeb, N. Agoulmine, and Z. Choukair, "Cloud RAN architecture model based upon flexible RAN functionalities split for 5G networks," in *WAINA 2017*.

[16] F. Malandrino and C.-F. Chiasserini, "Getting the most out of your vnfs: Flexible assignment of service priorities in 5G," in *WoWMoM 2019*.

[17] T. Truong-Huu, P. M. Mohan, and M. Gurusamy, "Service chain embedding for diversified 5G slices with virtual network function sharing," *IEEE Comm. Letters*, vol. 23, no. 5, pp. 826–829, 2019.

[18] M. R. Crippa *et al.*, "Resource sharing for a 5G multi-tenant and multi-service architecture," in *European Wireless Conference*. VDE, 2017.

[19] I. Alawe *et al.*, "On the scalability of 5G core network: the AMF case," in *CCNC*. IEEE, 2018, pp. 1–6.

[20] ——, "Smart scaling of the 5G core network: an rnn-based approach," in *2018 GLOBECOM*. IEEE, 2018, pp. 1–6.

[21] X. Wang, C. Wu, F. Le, and F. C. Lau, "Online learning-assisted vnf service chain scaling with network uncertainties," in *2017 CLOUD*. IEEE, 2017, pp. 205–213.

[22] X. Fei, F. Liu, H. Xu, and H. Jin, "Adaptive VNF scaling and flow routing with proactive demand prediction," in *INFOCOM*. IEEE, 2018, pp. 486–494.

[23] X. Zhang, C. Wu, Z. Li, and F. C. Lau, "Proactive VNF provisioning with multi-timescale cloud resources: Fusing online learning and online optimization," in *IEEE INFOCOM 2017*. IEEE, 2017, pp. 1–9.

[24] W. da Silva Coelho, A. Benhamiche, N. Perrot, and S. Secci, "Network function mapping: from 3G entities to 5G service-based functions decomposition," *IEEE Comm. Standards Mag., In press (HAL Id.: hal-02446529)*, 2020.

[25] C. Mobile, "C-RAN: the road towards green RAN," *White paper, ver*, vol. 2, pp. 1–10, 2011.

[26] L. M. Larsen, A. Checko, and H. L. Christiansen, "A survey of the functional splits proposed for 5G mobile crosshaul networks," *IEEE Comm. Surveys & Tutorials*, vol. 21, no. 1, pp. 146–172, 2018.

[27] 3rd Generation Partnership Project, "3GPP TR 38.801 V14.0.0 :Study on new radio access technology ," 2017.

[28] Mobile and wireless communications Enablers for the Twenty-twenty Information Society, "3GPP TR 38.913 V14.3.0 : Final Report on Architecture," *ICT-317669-METIS/D6.4*, 2015.

[29] P. Marsch *et al.*, "5g radio access network architecture: Design guidelines and key considerations," *IEEE Comm. Mag.*, vol. 54, no. 11, 2016.

[30] "W. Da Silva Coelho et al: Mathematical Formulation for the Network Slice Design Problem," https://hal.archives-ouvertes.fr/hal-02448028 (HAL Id.: hal-02448028), 2020.

[31] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, "SNDlib 1.0–Survivable Network Design Library," in *Proceedings of the 3rd International Network Optimization Conference (INOC 2007), Spa, Belgium*, April 2007, http://sndlib.zib.de.

[32] N. G. M. N. Alliance, "5G White Paper," 2015.

[33] W. da Silva Coelho. (2020) NSDP: source code and instances. [Online]. Available: https://github.com/wdscoelho/NSDP