Function splitting, isolation, and placement trade-offs in network slicing

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

Function splitting, isolation, and placement trade-offs in network slicing

Wesley da Silva Coelho, Amal Benhamiche, Nancy Perrot, Stefano Secci, *Senior, IEEE*

***Abstract*—We model the network slice provisioning as an optimization problem including novel mapping and provisioning requirements rising with new radio and core function placement policies. We propose an open-access framework based on an MILP formulation that encompasses flexible functional splitting, with possibly different splitting for different slices and slice subnets, while taking into account different network sharing policies from 5G specifications. We also consider novel mapping and continuity constraints specific to the 5G architectures and beyond. We show by numerical simulations the impact of taking into full and partial consideration these peculiar novel technical constraints.**

***Index Terms*—Network slicing; functional split; sharing policy**

I. Introduction

Telecommunications network infrastructures evolved with 5G [2] and the development of the 'network slice' as a novel virtualized infrastructure model. This technology now not only covers application-level slice abstraction as done with preliminary works on 'slicing', but also physical and switching layers virtualization, with different radio access and link communication technologies. This transition challenges slice network design since multiple resources and segments, historically managed independently from each other, are to be operated with continuity in networking and computing resource allocation and provisioning as a whole and unique service. In this context, different providers can be associated with different communication services 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 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 to be applied. For instance, in 5G a first service classification in three classes is given [3]: enhanced Mobile Broadband (eMBB), Ultra-Reliable Low Latency Communications (URLLC), and massive Machine Type Communications (mMTC). Each of these application categories has its specific requirements, such as maximum latency, minimum availability, and bandwidth capacity and, to provide a flexible environment to support those customized networks, novel infrastructures are supported by Network Function Virtualization [4], Software Defined Networking [5], and Network Slicing [6] technologies. Hence, each CS provider is to be able to deploy its services on top of logical networks, named Network Slices, specifically tailored to its technical requirements.

At the state-of-the-art, optimization approaches related to network slicing mostly considered it either as Virtual Network Embedding [7], Function Placement and Routing [8], or Service Function Chaining [9] problems. Addressing end-to-end network slicing, however, requires considering heterogeneous resources from 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 network slices; (ii) Slices can be decomposed into Network Slice Subnets; and (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, continuity constraints among slice subnets and 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 requirements on latency, availability, and network and computing capacities hold as well.

Taking the 3rd Generation Partnership Project (3GPP) [3], [10], [11] 5G standard as the reference system, our main contribution relies on formally defining the network slice design problem as a comprehensive network function dimensioning, placement, routing, and mapping framework. Firstly, we take into consideration the above-mentioned new mapping dimensions by modeling the relationship between flexible radio access functional splitting, control-plane and data-plane function isolation, and core network function placement. Secondly, we show by numerical simulations the impact of taking into full and partial consideration these peculiar novel technical constraints. Even though several works partially cover the network slice design problem [12], [13] and related sub-problems, such as functional split mode selection [14]–[19], network slicing with VNF sharing [20]–[22], and network slicing with VNF scaling [23]–[27], 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 on both virtual and physical networks.

A preliminary version of this article was presented at IFIP CNSM 2020 [1].

W. da Silva Coelho is with CNAM and Orange Labs, France. Email: wesley.dasilvacoelho@orange.com

A. Benhamiche and N. Perrot are with Orange Labs, France. Email: {firstname.lastname}@orange.com

S. Secci is with CNAM, France. Email: stefano.secci@cnam.fr

This paper is an extension of [1], differing in the following aspects: (i) we model new variants and extensions of the problem, and (ii) we provide additional analyses regarding the impact of each proposed variant on the network. In particular, about (i) we take into consideration new technical constraints in the NSDP mathematical formulation such that flexible splitting can happen within the same slice, hence fully exploiting the capabilities of forthcoming Open-RAN barriers to break in the development of Open-Distributed Units and related Open-Cloud infrastructure extension, described in [28] as Scenarios E and F. Moreover, in the new model, we consider and evaluate novel inter-slice split continuity constraints, and integrate optimized link load aspects.

The remainder is organized as follows. We present in Section II the state of the art, the entities appearing in new generation networks, and the main modeling aspects and technical constraints related to 5G systems and beyond. We formally state the Network Design Problem in Section III and introduce its mathematical formulation in Section IV. We present some variants of the problem in Section V. Section VI presents the experiments and discusses the results. We conclude in Section VII. Tab. I summarizes the abbreviations in this paper.

II. Background

We first draw the new mapping requirements and the related taxonomy. Then, we present the requirements rising with 5G systems in terms of sharing policies and functional splitting in radio access, meant to stay valid with future generations.

A. 5G System Mapping Requirements

The 3GPP specifications [3], [10], [11] present the different entities appearing in 5G systems; as we describe in [29], they are: User Equipment (UE), Communication Service (CS), Network Slice (NS), Network Slice Subnet (NSS), Network Function (NF), NF Service (NFS).

TABLE I: Abreviations

Acronym	Definition
AMF	Access and Mobility Management Function
AN	Access Network
BBU	Baseband Unit
CP	Control-Plane
CN	Core Network
C-RAN	Centralized Radio Access Network
CS	Communication Service
CU	Centralized Unit
DP	Data-Plane
DU	Distributed Unit
MAC	Medium Access Control
NF	Network Function
NFS	Network Function Service
NS	Network Slice
NSS	Network Slice Subnet
PDCP	Packet Data Convergence Protocol
PHY	Physical
PNF	Physical Network Function
RAN	Radio Access Network
RF	Radio Frequency
RLC	Radio Link Control
RRC	Radio Resource Control
RRU	Remote Radio Unit
UE	User Equipment
VNF	Virtual Network Function

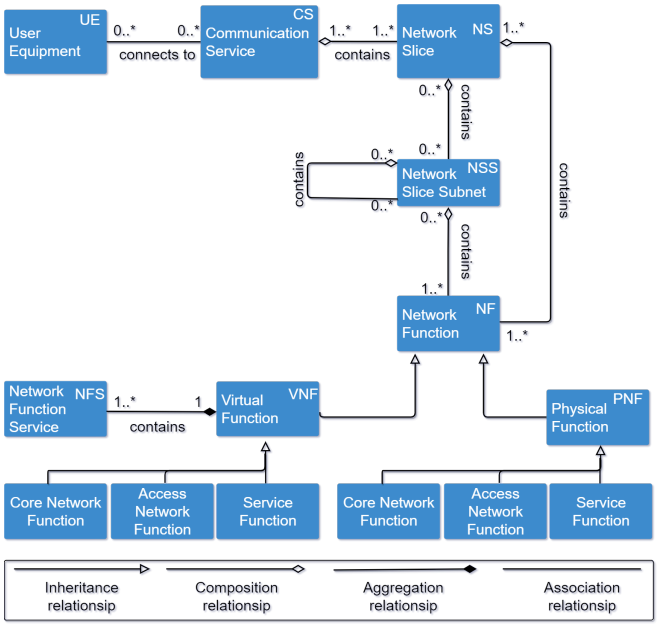

Fig. 1: Relationships between 5G entities. Source: [29].

Fig. 1 depicts the relationships between these entities. A UE can be a smartphone, a robot, or even an autonomous car, that might be connected to several CSs; e.g., a car connected to an Autonomous Car Service while broadcasting movies and music from Streaming Services to its passengers. To better deal with heterogeneous technical constraints of each service, each CS 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). 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 NFS; note this is meant to be a compulsory behavior in our modeling, but only a possible NF behavior for both control-plane and data-plane functions. Example of interactions between NFs are depicted in Fig. 2; one NF might consume NFSs from different NFs, and might also offer NFSs to different NFs. It is also important to notice that an NFS can also provide and consume services to and from other NFS within the same function. Finally, note that one NF can be virtualized (VNF) or physical (PNF).

3GPP technical documents report a non-exhaustive list of possible interactions between different functions and NF ser-

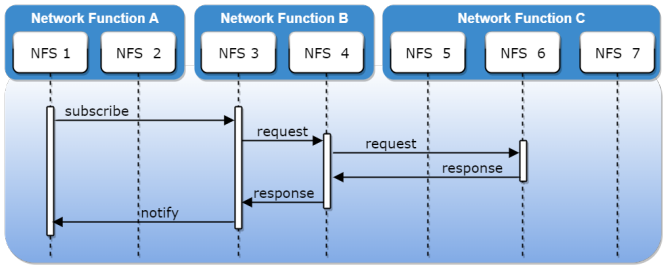

Fig. 2: Service-based interactions between network functions. Source: [29].

vices [10], [30], [31]. Some of those iterations are exemplified in Table II, where UDM is taken as an example with its non-exhaustive list of NF services [31]. Note that UDM might offer the same service to different NF service consumers. In this example, UDM can subscribe data from AMF, SMF, and SMSF. It is also important to notice that the same service might make different operations (e.g. get, subscribe, update, result confirmation) and they are made by request/response and subscribe/notify queries.

In this work, we do not differentiate VM-based network functions (VNFs) from container-based network functions (CNF). It is worth mentioning, however, that the main difference between these two types of network functions is based on how they are built and embedded into the physical network; while VNFs are deployed as virtual machines that run on their own operational systems, CNFs are deployed as containers, which facilitates the scalability and migration of such entities within cloud-native platforms [32]–[34]. Since the scope of this work concentrates on a higher abstraction level for network slice deployment, the actual nature of the virtualized network function, VM-based or container-based, does not affect our modeling and algorithms. Hence, we refer to the generic term VNF as any virtualized network function.

As presented in [29], we can distinguish five mapping levels for creating a complete virtual environment in 5G systems and beyond, each of which has its technical requirements. Besides the UE to CS mapping (a user can use concurrently multiple CSs), 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.

TABLE II: Example of a NF service decomposition: UDM as NF service producer

NF Service	NF Service operations	Signaling semantic	NF Service consumers
Subscriber Data Management	Get	Request/Response	AMF, SMF, SMSF
	Subscribe	Subscribe/Notify	
	Notification		
UE Context Management	Get	Request/Response	NEF, SMSF
	Update		
UE Authentication	Result Confirmation		AUSF
Parameter Provision	Update	Request/Response	NEF

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 instance 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 Communication Services: 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. Note that this level of mapping can also be done with active (already deployed) NSs.

We have to stress that the decomposition of NSs into NSSs and of NFs into NFSs is, on the one hand, motivated by scalability and efficiency reasons and, on the other hand, requires the network slice design process to take into consideration continuity constraints. Indeed, one NS can be geographically deployed in a scalable manner thanks to the segmentation of a slice into multiple NSSs; and the overall computing demand can be decreased by allocating resources to NF micro-services rather than to macro NF units. Moreover, depending on techno-economic and network management policies, continuity constraints on the decomposition may be needed; more precisely, the slice provider might decide whether all NSSs belonging to the same higher level NS should undergo the same functional split setting in the Radio Access Network (RAN).

B. Sharing policies

Given the expected data volume from each UE connected to each antenna 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.

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 [11]. We assume that an NF can treat data from two or more NSs if and only if they have an affinity for each other. By affinity,

we mean allowing a network slice to share one or more NFs with another NS. It is important to mention that an NS request might impose isolation constraints only on a specific subset of network functions that cannot be shared with a specific subset of NSs; this might be the case for critical NFs or network slices belonging to the different tenants, for example.

We depict in Fig. 3 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 own and dedicated user Data-Planes (D-DPs); it may be a solution for NSs requiring low end-to-end latency, and 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 [10]). For instance, Unstructured Data Storage Function (UDSF) can be shared by any function from the same Public Land Mobile Network (PLMN) or even be dedicated to a specific NF. Also, for example, in non-roaming 5G System architecture for concurrent access to two network slice subnets, a Session Management Function (SMF) might potentially be shared by their respective User Plane Function (UPFs). It is interesting to notice that, for this SMF example, while the SMF may be shared among slices and related functions based on the 5G systems standard, current

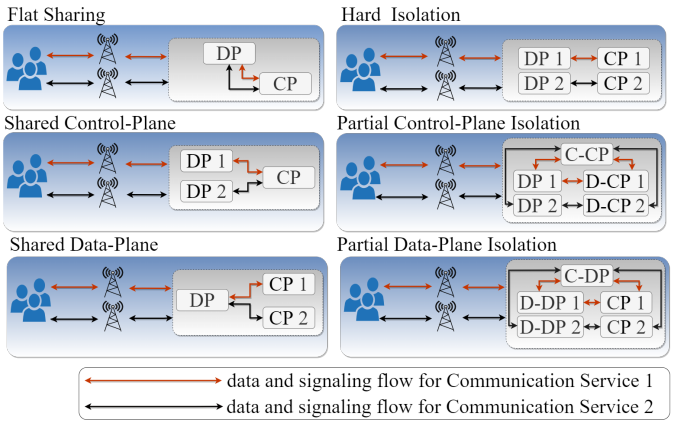

Fig. 3: NF sharing policies.

implementations by major vendors do not encompass SMF sharing. In addition, regarding the orchestration complexity inherent to each sharing policy (e.g., security and route control), other configurations might be proposed to guarantee Service Level Agreements.

C. Resource scaling and allocation

Let us describe resource management, scaling, and allocation practices in network slicing that we model in our work.

1) NF scaling: Scalability is a crucial point in dynamic environments, such as mobile networks. Authors in [23] propose an algorithm based on Control Theory in order to balance the load on instances of a specific core-based NF, called Access Management Function (AMF). Their algorithm scales out or in the AMF instance depending on the network load in order to save both virtual and physical resources. In the same context, authors in [24] propose another solution to scale dynamically the 5G NFs; the proposed approach prevents the latency and avoids overloading the core network. Authors in [25] propose an online algorithm to minimize the cost for provisioning NF instances while minimizing the congestion in a data-center network. Authors in [26], [27] propose different proactive approaches in order to estimate the upcoming traffic and adjust NF deployment *a priori*. While [27] combine an online learning method with a multi-period online optimization algorithm, authors in [26] aim to minimize the error in predicting the service chain demands for new instance assignment and service chain rerouting. Moreover, authors in [20] address an NF scaling and sharing problem in order to minimize the redundancy throughout the virtual networks; they propose FlexShare, a near-optimal NF-sharing algorithm capable of ensuring priority and NF sharing decisions in polynomial time. Authors in [21] propose a mathematical formulation and a heuristic based on a goodness function in order to address large-sized network instances; they show that sharing NFs among network slices can use up to 30% less bandwidth and 45% fewer NF instances, compared to dedicated-NF approaches. It is important to mention that, differently from these works, our model also encompasses the control-plane and data-plane separation and considers different functional split options for designing NSs; these aspects are discussed in the following subsections of this paper.

2) Resource allocation constraints: In terms of multi-resource constraints rising in network slicing [35], we apply the *knee model* [36], which gives a linear relationship between allocated resources and the function processing bitrate: it is such that by multiplying by a factor the amount of CPU allocated to an NFS, the traffic processing capacity for this NFS is also multiplied by the same factor, and this behavior is restricted by the maximum amount of available CPU. In this context, applying the knee model, the maximum NFS traffic processing rate is constrained by the dominant resource proportionally most demanded by the NFS [35]. Figure 4 depicts an example of resource allocation for an NFS. Since the initial allocation already took 50% of the available amount of resource 2 and only 40% and 10% of resources 1 and 3, respectively, the maximum scaling factor for this NF is 2. In

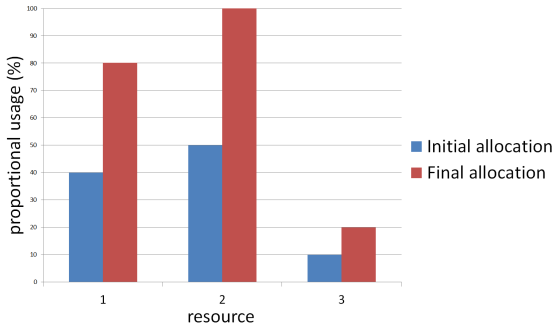

Fig. 4: Example of a resource allocation distribution for a NFS.

this example, resource 2 is the dominant one and the final scenario will allow the NF to treat two times more traffic compared to the initial scenario. In this work, we consider the CPU as the critical resource required by all NFS types and available in any physical node.

D. Routing with bandwidth and latency constraints

On top of mapping NFSs into network functions, we should provide a physical path connecting each pair of NFs that must be connected. In this context, each virtual link between two or more NFs has specific rules that must be ensured on the physical layer, such as ordering constraints, minimum bandwidth, and maximum end-to-end delay values. The objective might be to optimize the length of physical paths carrying NSs' flow while respecting the imposed technical constraints. These restrictions are the combination of technical constraints imposed by each network slice served by the given virtual structure. This sub-problem can be seen as a variant of the well-known multi-commodity flow problem [37] with additional latency and ordering constraints. Authors from [38], [39] address different aspects of this problem and propose mathematical models applied to 5G networks. In [38], the authors propose a framework that exploits the traffic information and topology of both backhaul and core networks for 5G systems; they propose a linear programming relaxation method and a heuristic method in order to better manage network load balancing, achieving close-optimal solutions with low computational complexity. Authors in [39], in turn, aim at integrating backhaul and fronthaul traffic over the same transport layer; a routing optimization framework is proposed, taking into account delay and path constraints, as well as a heuristic to reduce the computational complexity and apply it to production-level networks.

E. Functional Splitting in the Radio Access Network

Flexible Radio Access Network splitting [19] is a technique meant to increase network efficiency leveraging NFV flexibility. In 1G and 2G RANs, all entities responsible for radio and baseband processing were distributed and integrated into each base station. To minimize costs and facilitate network deployment, it was proposed to split the base station into Remote Radio Unit (RRU, also called Remote Radio Head and Radio Unit) and Baseband Unit (BBU) (also called Data Unit). In this context, the RRU is responsible for Physical Layer functionalities, while the BBU is responsible for Data Link Layer functionalities [40]; the distance between these two entities could be up to 40 kilometers. However, there is still redundancy in the network: all RAN functionalities are replicated for each pair of BBU and RRU. To overcome this, centralized RAN (C-RAN) was first introduced in 2011 [41]; pools of BBUs with large capacity, now called Centralized Units (CUs), are proposed to treat traffic from a sub-set of RRUs, now named Distributed Units (DUs). Hence, one of the first tasks is to define the functionalities enabled locally at the DU, and those installed centrally at the CU and thus shared among a subset of DUs. Figure 5 illustrates different functional split options on the 4G stack, as the 5G RAN split options have not yet been specified. Let's 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 the low Physical (PHY) block are installed at a CU, while with option 5 all entities before the low Media Access Control (MAC) block are installed at DUs.

Since the functional split was originally meant to be made *a priori* (i.e., before deploying the network) choosing the best split [42] for each scenario is not trivial. Defining the distributed and centralized functionalities should take into account end-to-end delay and total bandwidth constraints on each physical path connecting DUs and CUs while optimizing the resource allocation. It is important to mention that all distributed functionalities should be installed in each DU to support any type of split. Complementary, centralized functionalities have few instances that are installed in CUs and are shared by a specific sub-set of DUs. The dependency factors such as varying network latency and capacity has recently motivated experimenting *dynamic* functional splitting, where the split decision can be reconfigured on a short time scale for one or a few split options [43].

Table III depicts different fronthaul (FH) bitrates and latency indicators for each functional split. The bitrates are calculated as in [42] for a scenario using 100 MHz bandwidth and 32 antenna ports, while the maximum accepted one-way latency through FH is proposed by 3GPP [40]. 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

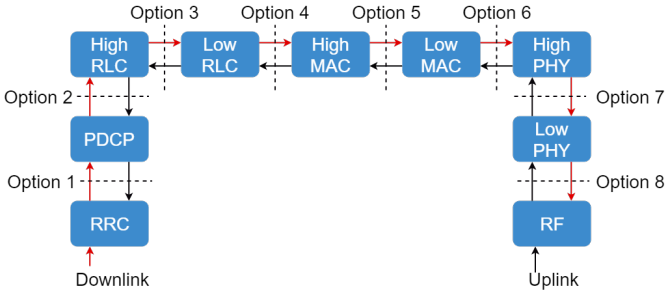

Fig. 5: Different functional split options.

TABLE III: 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

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 mobile systems. It is also important to point out the difference between downlink (DL) and uplink (UP) bitrates, due to physical layer operations (e.g., transformations between transport blocks and in-phase and quadrature symbols in each direction of the data flow).

Figure 6 represents a scenario with different split options with two operators and a RAN function chain composed of four NFs. In this example, the split between NF 1 and NF 2 is applied to treat the flow from DU 3 and DU 4. These two DUs have only NF 1 installed locally and send their flow to CU 2, which has NF 2, NF 3, and NF 4. Note that these functions in CU 2 are shared by both DU 3 and DU 4.

SDN and NFV technologies can be used together with C-RAN to offer flexibility to split RAN slice subnets [3], [40]. To this propose, two classes of RAN functions are proposed by [44]: *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. Hence, the related 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 [45], 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 [3], [40], [45], we incorporate flexible RAN splitting in order to design end-to-end network slices.

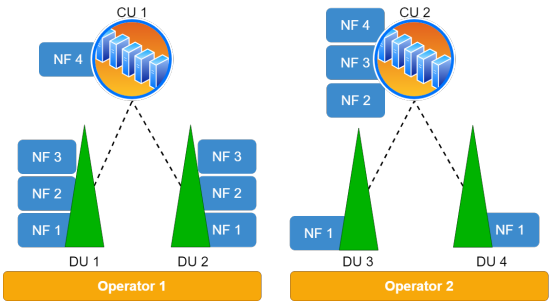

Fig. 6: Functional split example with four RAN NFs.

This approach can better deal with the heterogeneous requirements 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. [17]–[19] address the challenges of flexible functional split schemes in order to optimize the allocation of physical and virtual resources. [19], for example, proposes a new architecture that introduces a flexible split of RAN functionalities between the Cloud-RRH, an edge cloud, and the central cloud. [18], 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 a lower total cost of ownership can be achieved with optimal functional split compared to classical radio access networks. The required backhaul capacities for uplink traffic in terms of minimum bandwidth and maximum latency for different split options are analyzed in [17].

Even though there are several works in the literature addressing functional split mode selection and network slicing problems with NF sharing and NF scaling, no attention has been given to jointly address the complete problem in order to design end-to-end network slices including functional splitting for the radio access functions and different schemes for dimensioning and sharing NFs.

III. Problem Definition

We provide a network slice design problem statement, taking into account the presented requirements. Table IV summarizes the used notations.

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, in such a way that $V_p^{du} \cup V_p^{ac} \cup V_p^{ap} = V_p$ and $V_p^{du} \cap V_p^{ac} = V_p^{du} \cap V_p^{ap} = V_p^{ac} \cap V_p^{ap} = \emptyset$ hold. Every node $u \in V_p$ is associated with a number of available CPU c_u. 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 slices.

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

TABLE IV: Notation

Set	
V_p	Set of all nodes.
V_p^{du}	Set of all access nodes.
V_p^{ac}	Set of all non-access nodes.
V_p^{ap}	Set of all applications server nodes.
A_p	Set of all arcs.
$\delta^+(u)$	Set of all arcs going from node u.
$\delta^-(u)$	Set of all arcs going to node u.
F	Set of all NFS types.
F^d	Set of all data-plane NFS types.
F^c	Set of all control-plane NFS types.
S	Set of all network slice requests.
$F(s)$	Set of all CP NFS pairs that must be connected in slice s.
$G(s)$	Set of all pairs of NFSs from different type sets that must be connected to each other in slice s.
$K(s)$	Set of all demands of slice request s.
$O(s)$	Set of origin nodes of all traffic demand from slice s.
N	Set of all NFs.
Parameter	
c_u	number of available CPUs on node u.
b_a	bandwidth value on arc a.
d_a	delay value on arc a.
c_f	number of CPU required by NFS f.
$cap(f)$	traffic processing capacity of NFS f.
b_{fg}	total amount of traffic generated between NFSs f and g by an UE.
b_f	expected data rate of NFS f given one UE.
d_{fg}	the maximum accepted delay between NFSs f and g.
λ_f	compression coefficient of NFS f.
α_f^s	equals to 1 if a NFS type f must be present in slice s; 0 otherwise.
q_{fg}^{st}	equals to 1 if slice request s admits sharing a NFS of type f with a NFS of type g of slice t; 0 otherwise.
η_s	expected number of UEs connected to slice s
d_s	maximum accepted delay on data plane of slice s.
o_k	origin node of demand k
t_k	target node of demand k
b_k	expected volume of data between sent by origin node of demand k.

$F^d \cup F^c \cup \{f_0\} = F$ and $F^d \cap F^c \cap \{f_0\} = \emptyset$ hold[1]. Regarding the uplink direction, F^d is an ordered set composed of data-plane NFSs from both access and core networks. Every network function service $f \in F$ requires the minimum number of CPUs c_f needed to be packed into a 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} the maximum accepted delay[2] between NFSs f and g. Finally, for every $f \in F^d$, we denote by λ_f the compression coefficient on the DP traffic flow related to the initial volume sent by any traffic request's origin node. 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[3], potentially of different types, and 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 the set of arcs connecting two nodes from V_s. For the control plane, we denote by $F(s) \subseteq A_s$ the set of arcs between CP NFSs such that for any pair $(f, g) \in F(s)$, $(f \in F^c) \wedge (g \in F^c)$ holds. Additionally, we denote by $G(s) \subseteq A_s$ the set of arcs between NFSs from different sub-sets of NFS types such that for any pair $(f, g) \in G(s)$, $(f \in F^c) \oplus (g \in F^c)$ holds. 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 also 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 identifiers of the origin and the destination physical nodes of traffic demand 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). Also, we denote 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 $K(s)$. Finally, we denote by n_s the expected number of UEs that are to be connected to slice s. In this work, 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, we take into consideration only the uplink direction on the data-plane flow.

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:

- $K(s)$ traffic demands can be routed in G_s using these NFs and respecting the selected functional split setting on the data-plane,
- the NFSs 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.

D. Example

Fig. 7 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 (3 for data-plane and 4

[1]Note we do not consider any service function (e.g. Firewall and Proxy), which can be easily added in model extensions.

[2]This is important, for example, when flexible functional splitting is applied on the radio access; the selected split must respect the maximum fronthaul latency proposed by standards organizations.

[3]We assume that every NF already contains an intelligent entity responsible for directing the incoming flow to the right hosted NFS copy.

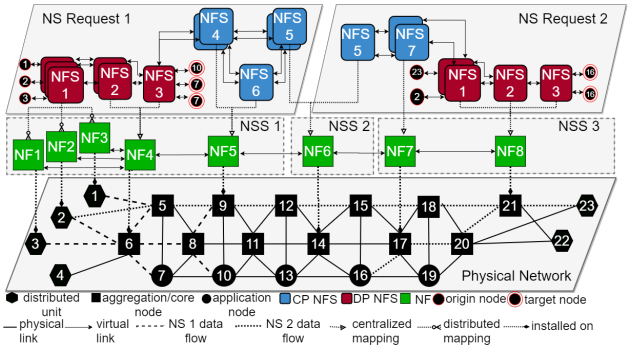

Fig. 7: Example of a solution for a NSDP instance.

for control-plane), 8 NFs, 3 NSSs, and a physical network with 23 nodes (6 DUs, 12 aggregation/core nodes, and 5 application nodes). Note first 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

We now introduce the mathematical model addressing the Network Slice Design Problem.

A. Decision variables

The binary variable z_f^s takes value 1 if NFS f is centralized, and 0 otherwise. x_{nu}^{sf} is a binary variable that takes value 1 if NFS f, installed on node u, is packed into NF n serving slice s, and 0 otherwise. The variable w_{nu}^{sf} is the amount of NFS f serving slice s packed in NF n and installed on node u. The variable y_{nu}^{f} is the total number of NFSs of type f packed into NF n and installed on node u. γ_{fg}^{ka} is a binary variable that takes value 1 if arc a is used to route the flow between NFSs f and g for demand k, and 0 otherwise. Table V summarizes all decision variables used in this model.

B. Constraints

1) Split Selection: Inequalities (1) decide whether a NFS f serving a slice s is installed locally or centrally. Since the RAN NFSs are chained in a specific order, all NFSs on the same side of the selected split must be installed in the same way, that is, either locally or centrally. This ordering constraint is also represented by inequalities (1). Note that we need to consider only the uplink direction of the flow.

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

ensure that all distributed NFSs will be installed on all related origin nodes; we assume that NFSs from CP cannot be installed in a distributed manner. Constraints (3), in turn, ensure that all copies of the same centralized NFSs type will be installed in the same physical node. Note, however, that two NFs of different types (e.g., AMF and SMF) with their own sub-sets of NFSs might still potentially be installed on different physical nodes.

2) NFS Placement: Given a set $K(s)$, constraints (2) ensure that all distributed NFSs will be installed on all related origin nodes; we assume that NFSs from CP cannot be installed in a distributed manner. Constraints (3), ensure that all copies of the same centralized NFSs type will be installed in the same physical node. Note, however, that two NFs of different types (e.g., AMF and SMF) with their own sub-sets of NFSs might still potentially be installed on different physical nodes.

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

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

3) NF dimensioning: (4) calculate the exact amount of distributed and 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.

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

4) NFS Packing: (5) represent the isolation constraints on the virtual layer. These constraints are responsible for applying different sharing policies imposed by each NS demand type. Constraints (6), in turn, ensure that an NF will not be present in more than one physical node.

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

$$x_{nu}^{sf} + x_{nv}^{tg} \leq 1, \forall s, t \in S, f, g \in F, n \in N, u, v \in V_p : v \neq u \qquad (6)$$

$$\sum_{s \in S} w_{nu}^{sf} \leq y_{nu}^{f} \qquad ,\forall n \in N, \forall u \in V_p, \forall f \in F \qquad (7)$$

TABLE V: 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

Let us explain in detail the inequalities (7) with some examples. Suppose that NFSs of type f from s and t cannot be packed together ($\forall n \in N, x^{sf}_{nu} \oplus x^{tf}_{nu}$). Hence, all copies of f installed on node u and serving s are not shared with t. In this way, if (4) set w^{sf}_{mu} to 4.60 and w^{tf}_{nu} to 1.25, for example, we must install at least seven ($\lceil 4.60 \rceil + \lceil 1.25 \rceil$) NFSs of type f on the node u using two different NFs. Now, let s and t be two slices with no isolation constraints and using the same NF for a given NFS f ($x^{sf}_{nu} \wedge x^{tf}_{nu}$). Suppose that (4) have set w^{sf}_{bu} and w^{tf}_{nu} equal to 4.60 and 1.25, respectively. Since both s and t accept NFS sharing with each other ($q^{st}_{ff} \wedge q^{ts}_{ff}$), we need to install only six ($\lceil 4.60 + 1.25 \rceil$) NFSs of type f on node u instead of seven of them. Using this approach on residual capacities, this saving can be even greater if we have a bigger sub-set of slices having $q^{st}_{fg} = 1$ for a given tuple (s, t, f, g).

5) Physical node capacity: (8) ensure that there will not be more installed NFs than a node can support. Note that the capacity constraints on physical nodes are considered at the NF level, which dependents directly on the physical capacity required by the inner NFSs of each NF.

$$\sum_{n \in N} \sum_{f \in F} c_f y^f_{nu} \leq c_u \qquad , \forall u \in V_p \qquad (8)$$

6) Routing: Constraints (9) represent the conservation flow constraints on both control-plane and data-plane traffics. Note that, since there can be only one virtual control-plane for each slice request, γ variables related to the set $F(s)$ can be indexed to only one k; we chose the first traffic demand to represent the whole control-plane on each slice. These constraints also represent the conservation flow constraints on the data-plane for each traffic demand k; they provide a path between each pair NFSs from DP, from the origin node of each traffic demand k to the first related data-plane NFS, and between the last data-plane NFS and the target node for each traffic demand k. Note that we use the dummy function f_0 in order to find a physical path between it and the data-plane chain if and only if its first NFS is installed centrally.

$$\sum_{a \in \delta^+(u)} \gamma^{ka}_{fg} - \sum_{a \in \delta^-(u)} \gamma^{ka}_{fg} =$$
$$\begin{cases} z^s_f - 1 & , \text{ if } (f,g) \in G(s), f \in F^c, u = o_k, \\ 1 - z^s_f & , \text{ if }, u = o_k, (f,g) \in G(s), f \in F^d \\ & \text{ or if } u = o_k, f = f_{|F^d|}, g = f_0; \\ -\sum_{n \in N} x^{sg}_{nu} & , \text{ if } u \in V \backslash V^{du}, f = f_0, g = f_1 \\ z^s_g & , \text{ if } u = o_k, f = f_0, g = f_1 \\ -1 & , \text{ if } u = t_k, f = f_{|F^d|}, g = f_0 \\ \sum_{n \in N} x^{sf}_{nu} & , \text{ if } u \in V \backslash V^{du}, f = f_{|F^d|}, g = f_0 \\ z^s_g - z^s_f & , \text{ if } u = o_k, \forall f, g \in F^d | g = f + 1 \\ \sum_{n \in N} x^{sf}_{nu} - \sum_{m \in N} x^{sg}_{mu} & , \text{otherwise.} \end{cases}$$
$$\forall k \in K(s) : s \in S, \forall f, g \in F, \forall u \in V \qquad (9)$$

7) Latency: Inequalities (10) ensure that the maximum end-to-end latency imposed by each slice request s will be respected on the path between o_k and t_k for every traffic demand k. Note that these technical constraints are applied only on the data-plane and only on the uplink direction as discussed before; we assume an arc has the same latency constant value d_a in both directions. Inequalities (11) ensure that the maximum latency between NFSs f and g will be respected on both data and control planes, these constraints are applied in both control and data planes.

$$\sum_{a \in A_p} d_a (\gamma^{ka}_{f_{|F^d|} f_0} + \sum_{f \in \{f_0\} \cup F^d \backslash \{f_{|F^d|}\}} \gamma^{ka}_{ff+1}) \leq d_s$$
$$, \forall k \in K(s) : s \in S \qquad (10)$$

$$\sum_{a \in A_p} d_a \gamma^{ka}_{fg} \leq d_{fg} \quad , \forall k \in K(s) : s \in S, \forall f, g \in F \qquad (11)$$

8) Physical link capacity: Inequalities (12) ensure that an arc will not carry more data than it can support. Note that comprehension coefficients are considered in these constraints.

$$\sum_{s \in S} \sum_{k \in K(s)} b^k (\lambda_{f_{|F^d|}} \gamma^{ka}_{f_{|F^d|} f_0} + \sum_{f \in \{f_0\} \cup F^d \backslash \{f_{|F^d|}\}} \lambda_f \gamma^{ka}_{ff+1})]$$
$$+ \sum_{s \in S} n_s (\sum_{(f,g) \in F(s)} b_{fg} \gamma^{k_s a}_{fg} + \sum_{(f,g) \in G(s)} \sum_{k \in K(s)} \frac{b_{fg} \gamma^{ka}_{fg}}{|K(s)|}) \leq b_a$$
$$, \forall a \in A_p \qquad (12)$$

C. 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:

$$min \sum_{f \in F} \sum_{n \in N} \sum_{u \in V_p} y^f_{nu} + \Omega \sum_{a \in A_p} \sum_{s \in S} \sum_{k \in K(s)} \sum_{f,g \in F} \gamma^{ka}_{fg} \qquad (13)$$

subject to (1)-(12) and

$$y^f_{nu} \geq 0 \qquad \in \mathbb{Z} \qquad (14)$$
$$x^{sf}_{mu}, z^s_f, \gamma^{ka}_{fg} \qquad \in \{0, 1\} \qquad (15)$$
$$w^s_{uf} \geq 0 \qquad \in \mathbb{R} \qquad (16)$$

While the first term in (13) is related to the number of installed functions, the second term in (13) multiplied by Ω is inserted to mechanically avoid loops, with a negligible qualitative impact on the network solution. An alternative way to the second term would be to add loop avoidance constraints, however increasing the complexity. Network designers may want to tune the factor Ω to drive the solution toward the desired outcome (e.g., to emphasize the number of NFSs over the number of links).

V. Variants and extensions for the NSDP

In the following, we present a few relevant variants of the Network Slice Design Problem.

A. NDSP with intra-slice flexible splitting

With this variant of the problem, different split settings can be selected within the same slice. In other words, flexible splitting is applied independently to each DU related to a given slice. For this purpose, we apply a pre-processing to transform each traffic demand into a slice request[4]. Hence, any NS request is now composed of only one traffic demand (i.e., representing a unique traffic demand of the initial NS request). In order to impose a shared control-plane to all DU related to the same initial NS (i.e., before the pre-processing), we introduce β_{st}, a binary parameter generated during the pre-processing: it holds 1 if the new slices s and t come from the same initial NS request (i.e., before the decomposition); 0 otherwise. Finally, in order to reduce the management complexity, we add the new constraints (17): they impose that requests from the same NS must share the same control-plane NFSs. Note that the single requests from the same original slice can have their own data-plane. We refer to this variant by NSDP with intra-slice flexible splitting (NSDP-ISFS).

$$\beta_{st} - 1 \leq x^{sf}_{nu} - x^{tf}_{nu} \leq 1 - \beta_{st} \quad ,\forall s,t \in S, f \in F^c, n \in N, u \in V_{ac} \quad (17)$$

Note that, constraints (17) impose the binary variables x to have the same value (i.e., either 1 or 0) if and only if the related parameter β holds 1; otherwise, these inequalities are implicitly relaxed. After applying the described pre-processing on the initial input, the original formulation of NSDP (1)-(16) can be directly applied along with the new constraints (17).

B. NDSP with inter-slice split continuity

We propose this variant in order to represent the scenarios with strict split setting constraints on each DU. In fact, imposing the same split selection for any traffic demand traversing a given DU might be necessary to reduce the management complexity. Complementary to Ineq. (17), we add the new constraints (18), where ρ_{st} is a binary parameter generated during the pre-processing; it holds 1 if the new slices s and t have the same origin DU node as their traffic demands; 0 otherwise.

$$\rho_{st} - 1 \leq z^s_f - z^t_f \leq 1 - \rho_{st} \quad ,\forall s,t \in S, \forall f \in F^d \quad (18)$$

Note that these inequalities can only be applied to instances whose slice requests have only one traffic demand (i.e., after pre-processing). We refer to this variant by NSDP with inter-slice split continuity (NSDP-ISSC).

C. NSDP with optimized link load

In order to minimize the traffic volume throughout the network, we introduce U, a continuous variable that represents the maximal load among physical links. We then replace constraints (12) by Ineq. (19) and (20) and add the new constraints (21) in order to impose upper bounds to y variables; these inequalities are important to this NSDP variant since we no longer have the related component within the new objective function (22).

[4]Following the taxonomy presented in this work, these new pos-processed requests can be seen as network slice subnets.

$$\sum_{s\in S}\sum_{k\in K(s)} b^k(\lambda_{f_{|F^d|}}\gamma^{ka}_{f_{|F^d|}f_0} + \sum_{f\in\{f_0\}\cup F^d\setminus\{f_{|F^d|}\}} \lambda_f\gamma^{ka}_{ff+1})] + \sum_{s\in S} n_s(\sum_{(f,g)\in F(s)} b_{fg}\gamma^{k_s a}_{fg} + \sum_{(f,g)\in G(s)}\sum_{k\in K(s)} \frac{b_{fg}\gamma^{ka}_{fg}}{|K(s)|}) \leq b_a U \quad ,\forall a \in A_p \quad (19)$$

$$0 \leq U \leq 1 \quad (20)$$

$$y^f_{nu} < 1 + \sum_{s\in S} w^{sf}_{nu} \quad ,\forall n \in N, \forall v \in V_p, \forall f \in F \quad (21)$$

The new objective function is then formulated as following:

$$min\ U \quad (22)$$

This formulation can be applied to any NSDP variant and a similar model can be generated in order to minimize the maximal load on physical nodes.

VI. Numerical results

We detail the simulation setting and then expose the results.

A. Test setup

The simulated scenarios represent realistic areas, such as small cities and dense zones to scale with the complexity of the formulation while stressing the impact of functional splitting on the placement of network functions services. Each simulated parameter follows those proposed in related technical documents [3], [46] in order to provide scenarios that are as realistic as possible.

1) Physical topologies: We simulated different physical networks with different features. Inspired by common access networks structure, we first propose a specific topology called *Mandala* (Fig. 8a) with the following structure: given *n* DU nodes, we have *n/4* aggregation nodes, *n/4* core nodes, and *n/8* application nodes. Note that n must be equal or multiple of 8. 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, where demands are served. Finally, given two different nodes u and v, there exists one arc (v,u) for each arc (u,v). Fig. 8a 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 capacity constraint (they are considered as sink nodes), each one of DU, aggregation, and core nodes provides 30 servers, each of which with 16 CPUs; this capacity corresponds to 12.5%

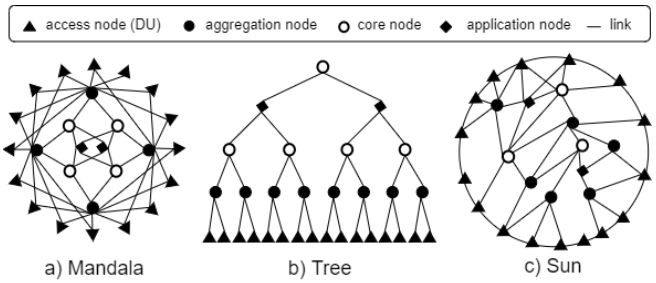

Fig. 8: Physical network structures: examples with 16 DUs.

TABLE VI: Simulated slice demand setting

Slice	Service required	Aditional 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

of the global generated CPU computing demand (i.e., with no function sharing) and enables to test all split settings and sharing policies. In addition, fronthaul links (i.e., between DUs and aggregation nodes), backhaul links (i.e., between aggregation and core nodes), and core links (i.e., between core and application nodes) have link 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 in each arc randomly takes a value between: 50μs and 100μs for fronthaul links, 200μs and 300μs for backhaul links, and 400μs and 600μs for core links[5].

We also run our tests on two different physical topologies: one binary tree-based structure (hereinafter referred to as *Tree*; Fig. 8b) with 31 nodes and 60 arcs, and *Sun* from SNDlib [47] composed of 27 nodes and 102 arcs (Fig. 8c). 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.

In order to represent the virtual RAN for each slice, we set F^d with five data-plane NFS types: NFS1 represents functions of the MAC bloc; NFS2 represents functions of the RLC block; NFS3 represents functions from PDCP block; NFS4 represents functions from RRC block; NFS5 represents DP functions from the core network.

2) Virtual layer: In order to represent the virtual RAN for each slice, we set F^d with five data-plane NFS types: NFS1 represents functions of the MAC bloc; NFS2 represents functions of the RLC block; NFS3 represents functions from PDCP block; NFS4 represents functions from RRC block; NFS5 represents DP functions from the core network[6]. In addition, there are four mandatory control-plane 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 [29]). Each NFS has a processing capacity $cap(f)$ 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 roughly 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 III 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 DP NFSs. Additionally, the acceptable latency d_{fg} between two DP NFSs from F^d also follows those in Table III, taking the upper bound when an interval is proposed. Finally, the latency d_{fg} involving any CP NFS is set to 500μs; this value corresponds to 5% of the total CP latency proposed by 3GPP [3].

TABLE VII: Scenarions: 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. 5.
Setting 4	for each slice, only NFS1 and NFS2 are distributed; it corresponds to 3GPP's split 2 in Fig. 5.
Setting 5	for each slice, only NFS1 is installed locally. It corresponds to 3GPP's split 4 in Fig. 5.
Setting 6	all DP NFSs are installed centrally for all NS requests. It corresponds to 3GPP's split 6 in Fig. 5
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.

3) Slice requests: We tested instances with four NS requests, each with four 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 data-plane NFSs, four mandatory control-plane NFSs, and a different set of additional NFSs that can be required (see Table VI). 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 CP NFS6 only (e.g., corresponding to the Access and Mobility Management Function, AMF, in 5G core [29]) to all DP NFSs (NFS1..NFS5). 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

[5]Note that the end-to-end latency along the shortest path between any DU and application node is at most 1ms. This value is commonly used as a threshold to strict latency constrained 5G services [46].

[6]Since RF and PHY blocs have synchronous network functionalities that pose extremely strict latency requirements, we assume they are PNFs integrated to each DU. Hence, they are not considered in our virtual DP chains. In addition, We take into consideration only splits between the macro-blocs (e.g., PHY-MAC and RLC-PDCP). However, our model is flexible enough to consider any sub-split proposed by 3GPP. In fact, it depends only on how the input instance is defined: since our model considers the data-plane functions individually, one can describe the DP chain as a set of ordered inner functionalities from each bloc.

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 Table VI, we applied the assumptions proposed by [46] 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 latency on the data-plane, which restrains the placement possibilities of the related NFSs. The other two slice requests are intermediate regarding both aforementioned parameters; request 2 can be seen as an mMTC application. Finally, being an origin of one of some slice's traffic demands, 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. 5, 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. 3), the split settings impose different sets of distributed and centralized DP NFSs. Table VII summarizes the tested scenarios.

B. Numerical results

The analyses made in the following sub-sections 1 and 2 are related to the formulation (1)-(26) and present the numerical experiments of each variant of the problem on sub-section 3. We implemented our model in a Julia-JuMP environment using ILOG CPLEX 12.8 as the linear solver. We set Ω to an enough small value (i.e., 10^{-3}) on the objective function (13) only to prioritize elementary paths to carry traffic demands and to emphasize the number of NFSs over the number of links in the optimization process. Finally, our tests were run on a Linux server with an Intel Xeon E5-2650 CPU and 256GB RAM. The data-set and the source code are available on [48].

1) Execution time: Before discussing the results related to the simulation setting as previously detailed, we present the performance of our model on different instance sizes: we varied the number of NFSs available (from 3 up to 24), the number of traffic demands per slice request (from 1 up to 16), and the physical topology size (13, 26, and 52 physical nodes with random connection and average degree equals to 10). Finally, all instances had 4 NS requests. We run 30 tests of each instance size, varying both origin and target nodes of each traffic demand. Finally, we set the maximal number of parallel threads that could be invoked by the solver to 1 and the time limit to 10 800 seconds (three hours).

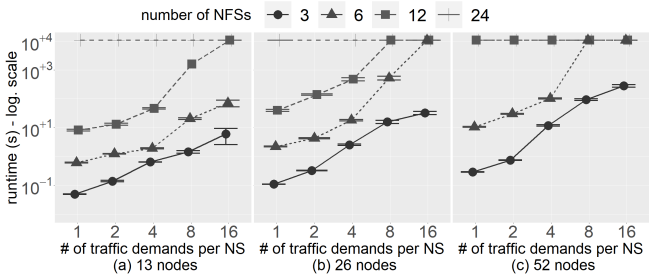

Fig. 9: Runtime on different NSDP instance sizes.

As shown in Fig. 9, the time needed to achieve the best solution increases exponentially with the size of the instance. In particular, the number of NFSs available has the worst impact on the model. Even with small topologies (13 nodes), the problem with 24 NFSs could not be solved within 3 hours. Due to the different levels of packing problems within the model, this time limit is also reached with the biggest topologies (52 nodes) and only 12 NFSs. The execution time also increases as the number of traffic demands increases. Due to the related routing sub-problems, the limit of three hours is also reached with any number of demands and 12 NFSs on large topologies. It is worthwhile to mention that, for those instances with 13 (resp. 52) nodes that reached the time limit and could not be solved to optimality by the solver, the average relative gap was roughly 7% (resp. 43%) with standard deviation equals to approximately 2% (resp. 8%).

2) Functional split and NF sharing: We now discuss the results depending on the presented network settings. We applied additional constraints to impose the desired split setting to all slice demands. Additionally, sharing policies were imposed by changing the q_{fg}^{st} parameters values used in Ineq. (5). All instances were generated using Mandala, Sun and Tree topologies with 16 DUs (see Fig. 8). Finally, we run 10 tests on each physical network varying both traffic demands' origin and destination nodes. The goal of the following numerical analysis is to assess the impact of novel mapping, splitting, and sharing policies on the network design.

Fig. 10 reports the average number of distributed and centralized NFSs on different sharing policies and split strategies for the three aforementioned physical topologies merged together (i.e., in the same results set here). 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);
- Since all NFSs are installed on all DUs related to each slice request, split setting 1 requires the largest number of NFS copies in all proposed sharing policies;
- Since each (resp. no) NFS copy is dedicated to a single NS, Hard (resp. Flat) Isolation has the greatest (resp. smallest) 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; the total number of NFSs when using shared policies (i.e., CP Shared and DP Shared) was always smaller than when using partial policies (i.e., Partial CP and Partial DP). 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

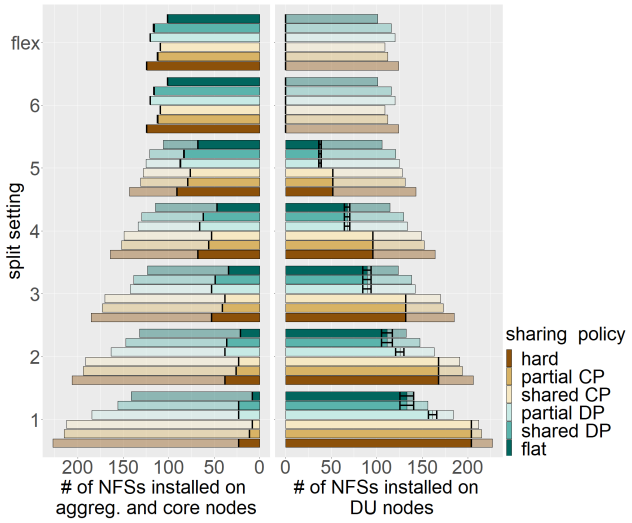

Fig. 10: Number of NFSs on different scenarios.

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 as 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.

In Fig. 11 we show that 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 ratios 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 sharing policies, requiring up to 100% of the capacity on the most loaded link (see Fig. 11e); 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 11c and 11d). This behavior is expected since all NFSs are installed centrally and the data volume sent by each traffic demand is completely decompressed before traversing the fronthaul links. Conversely, split setting 1 benefits from 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 (see Fig. 11e). However, as shown in Fig. 11a, this split is one of the settings that require the largest number of links (between 77% and 82%) since the NFSs from CPs and DPs are further from each other. Besides, CP NFS6 must be connected to all distributed DP NFSs of the related NS.

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. 11f). We also observe a decrease of the average load on physical nodes (see Fig. 11g), in particular on DU nodes (see Fig. 11h), 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. 11g). Unlike physical links and aggregation and core nodes (see Fig. 11b and Fig. 11i, respectively), DU nodes benefit from 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. 11j) provided 98% (resp. 43%) on average of its available resource applying split setting 6 (resp. setting 1) and Partial DP (resp. Flat) sharing policy. Note that, applying Hard, Shared CP, and Partial CP isolation policies, all distributed NFSs can serve only one slice. Hence, they will always demand the same capacity from DU nodes when the same split setting is applied (see

Even with a negative impact on the number of installed NFSs, mapped links, and nodes (see Figures 10, 11a, 11f, 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. 11b) and from aggregation and core nodes (see Fig. 11i) 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 (13) to a more suitable one. It is also worth mentioning that, in order 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 VII) to all slice requests, we did not observe a significant difference in the results using distinct physical topologies. For instance, comparing the three proposed topologies, the difference in the number of physical nodes hosting an NF, the ratio of active links, and 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.

3) NSDP and variants: We now present the impact of the proposed variants of the presented problem on the physical network. Henceforth, we refer by *NSDP* the original formulation (1)-(16); the other two variants refer to the proposed formulations as previously discussed. Since we set Ω to 10^{-3} in (13), we refer by *minNFS* this objective function while *minLinkLoad* refers to (22); as aforesaid, this objective function is implemented along with inequalities (19)-(21). Also, the sharing policy was randomly chosen for each pair of slices while we applied the Flexible Split setting along with the

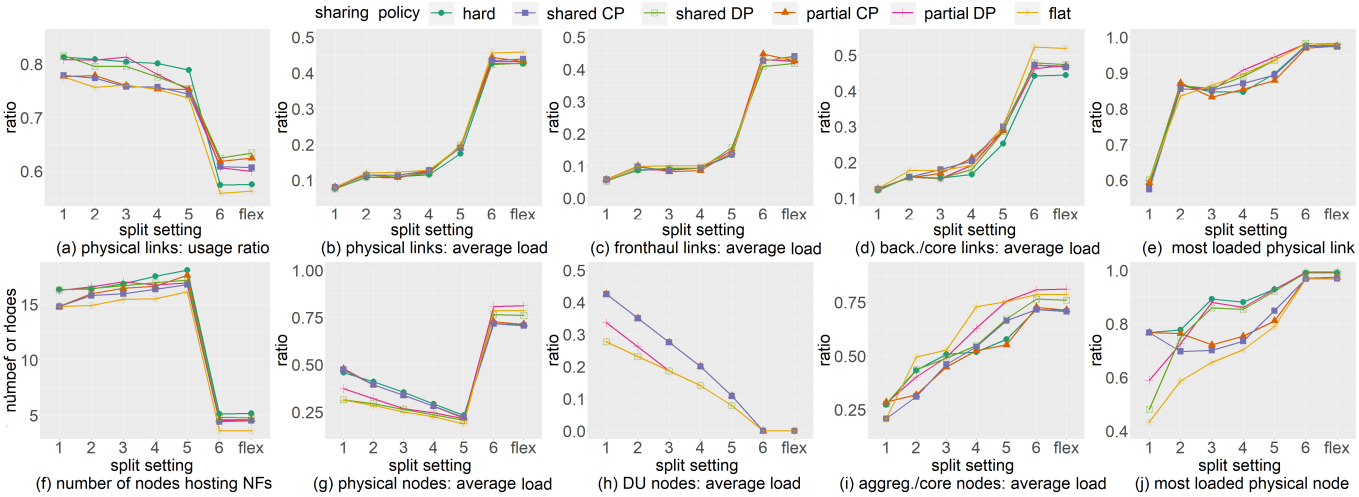

Fig. 11: Impact of different split settings and sharing policies on the physical network.

same parameters related to both physical and virtual layers as previously presented in this section. Finally, we run 10 tests on each combination of objective function, variant, and physical topology, varying both origin and target nodes of each traffic demand; for NSDP-ISFS and NSDP-ISSC variants, the pre-processing described in the previous section was applied on the NSDP instances.

Figure 12 shows the mean and the standard deviation of each depicted parameter. Minimizing the load on physical links generally increased the number of NFSs installed throughout the network (see Figures 12a and 12b) and the number of physical nodes hosting an NF (see Fig. 12h); regarding all variants, the average increase in terms of both numbers of NFSs and hosting nodes was roughly 400%. This is due to the impact of the compression coefficient related to data-plane NFSs; when installed locally, they can compress the data before leaving the origin node of each traffic demand. However, as seen in Fig. 12j, this strategy is limited by the available resources on DU nodes, which impose to install some DP NFSs centrally whenever the related physical capacities are reached (see Fig. 12k and Fig. 12l). Moreover, since the data flow is spread over as many physical links as possible, minimizing the load on physical links also increased the end-to-end DP latency (i.e., the latency between traffic requests' origin and target physical nodes); this increase was approximately 100% on all NSDP variants (see Fig. 12c).

Fig. 11h).

Applying different variants has also an important impact on the physical network. Since NSDP-ISFS and NSDP-ISSC variants allow sharing data-plane NFSs with other slices rather than the ones from the same initial slice request (which gives more flexibility to NFS placement decisions), different objective functions had opposite impacts on the physical network. Indeed, the NSDP approach demanded up to 40% more (resp. 50% less) bandwidth compared to the other variants when the number of NFSs (resp. load on the physical links) is minimized (see Figures 12d,12e, and 12f). Also, comparing the two objective functions, the load on physical links could be reduced by a factor of 3, and the most loaded physical link provided roughly 97% (resp. 34%) of its capacity when NSDP-ISSC (resp. NSDP) is applied along with minNFS (resp. minLinkLoad) objective function (see Fig. 12g). This behavior is explained by the concentration of NFSs on few physical nodes when the minNFS objective function is applied, hence stressing the related incoming links; applying the split setting 6 whenever is possible, this concentration is due to the greater number of centralized DP functions to be installed. Moreover, the NSDP-ISSC variant had a relevant impact on the physical nodes (see Fig. 12i); comparing this variant to NSDP and regarding the minNFS formulation, the load on physical nodes could be decreased from 85% to 65% on average.

NFSs' compression coefficients also play an important role in the final solution. Indeed, depending on the parameter to be optimized, different split settings are prioritized. For instance, when the load on the physical links is minimized, split setting 1 is always selected when it is admissible by DUs' capacity (see Fig. 12j); this functional split places all DP NFSs locally and therefore completely compresses the traffic demands' flow before sending it through fronthaul links (see Fig. 12e).

It is worth mentioning that, regarding the three proposed physical topologies, we observed an important difference only on the end-to-end DP latency and on the fronthaul links' load. In our simulations, while the average end-to-end DP latency was 2.50 ms on the Tree structure, these values increased to 4.20 ms and 6.60 ms on Sun and Mandala topologies, respectively. Let us recall that, since there is only one possible elementary path to connect any pair of physical nodes on the Tree topology and slice request 3 imposes a strict DP latency (see Tab. VI), the latency on the related physical links is lower than those found on Sun and Mandala structures; otherwise, some instances would be infeasible. For the same reason and in order to carry the flow from slice request 1, the links' capacity on Tree topology is greater than on the other two structures. Hence, when the number of NFS copies was minimized, the average load on fronthaul links on Tree, Sun, and Mandala was respectively 12.50%, 30%, and 40%. However, running the same instance on each of these topologies, we observed no difference in the selected split setting on the final solutions. However, this behavior might not be observed in real scenarios since physical networks are

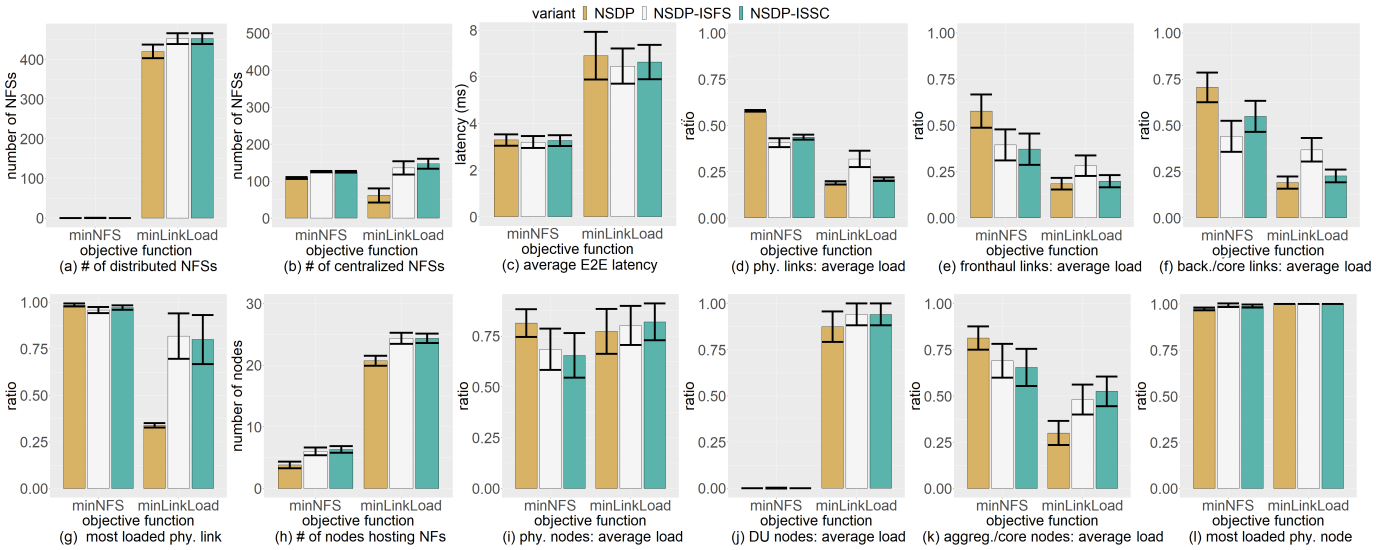

Fig. 12: Impact of different NSDP variants on the physical network.

unlikely to have enough capacity to allow any split setting and hence be able to allocate resources to all slice requests. This, therefore, reinforces the importance of applying flexible functional splitting in future 5G systems and beyond.

VII. Concluding Remarks

In this paper, we modeled the network slice provisioning as an optimization problem including novel mapping and provisioning requirements. In particular, we considered novel mapping dimensions appearing with 5G systems, modeling the relationship between flexible radio access functional splitting, control-plane and data-plane function separation, and sharing policies. Different variants of the problem were also proposed and the related models are compliant with running standards. We demonstrated by simulation the impact of taking into full and partial consideration of the peculiar constraints rising from the standards. For instance, we reported numerical results showing that flexible splitting appears as a key factor to deal with heterogeneous requirements to deploy distinct communication services, 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. We also observed that different variants related to the flexible splitting have an important impact on the physical network; depending on the selected approach, the average load on physical links could be reduced by a factor of 3.

Regarding the execution time performance on large instances and to attain (near-)optimal solutions in a competitive runtime, future works can focus on different exact and heuristic approaches applied to the problem addressed in this work. Also, other variants of the problem can be studied (e.g., applying Multi-access Edge Computing) and different parameters (e.g., latency) might alternatively be optimized.

Acknowledgement

Work funded by the Agence Nationale de la Recherche (ANR) MAESTRO-5G (ANR-18-CE25-0012) project.

References

[1] W. da Silva Coelho, A. Benhamiche, N. Perrot, and S. Secci, "On the impact of novel function mappings, sharing policies, and split settings in network slice design," in *International Conference on Network and Service Management*, 2020.

[2] "3rd Generation Partnership Project. Release 16: 5G system - phase 2," https://www.3gpp.org/release-16, accessed: 2020-11-13.

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

[4] 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.

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

[6] 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.

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

[8] 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.

[9] 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.

[10] 3rd Generation Partnership Project, "3GPP TS 23.501 V15.4.0: System Architecture for the 5G System ," 2018.

[11] ——, "3GPP TR 28.801 V15.1.0: Study on management and orchestration of network slicing for next generation network ," 2018.

[12] 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.

[13] 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.

[14] I. Koutsopoulos, "Optimal functional split selection and scheduling policies in 5g radio access networks," in *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2017, pp. 993–998.

[15] D. Harutyunyan and R. Riggio, "Flexible functional split in 5g networks," in *2017 13th International Conference on Network and Service Management (CNSM)*. IEEE, 2017, pp. 1–9.

[16] S. Matoussi *et al.*, "5g ran: Functional split orchestration optimization," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 7, pp. 1448–1463, 2020.

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

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

[19] 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*.

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

[21] 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.

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

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

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

[25] 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.

[26] 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.

[27] 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.

[28] O. R. A. N. Alliance, "O-RAN-WG6.CAD-V01.00.00: Cloud Architecture and Deployment Scenarios for O-RAN Virtualized RAN 1.0," 2019.

[29] W. d. S. Coelho, A. Benhamiche, N. Perrot, and S. Secci, "Network function mapping: From 3G entities to 5G service-based functions decomposition," *IEEE Communications Standards Magazine*, vol. 4, no. 3, pp. 46–52, 2020.

[30] 3rd Generation Partnership Project, "3GPP TS 23.502 V15.4.0 : Procedures for the 5G System ," 2018.

[31] ——, "3GPP TS 29.503 V16.0.0 : 5G System; Unified Data Management Services (Release 16)," 2019.

[32] G.-P. S. N. W. Group, "Cloud-Native and Verticals' services." 2018.

[33] M. Jayakumar, "White Paper: understanding how the core network is evolving to cloud-native." December 2016.

[34] G. Association, "Cloud Infrastructure Reference Model, Version 1.0 ," November 2020.

[35] F. Fossati, S. Moretti, P. Perny, and S. Secci, "Multi-resource allocation for network slicing," *IEEE/ACM Transactions on Networking*, pp. 1–14, 2020.

[36] Y. Etsion, D. Tsafrir, and D. G. Feitelson, "Process prioritization using output production: scheduling for multimedia," *ACM TOMM*, vol. 2, no. 4, pp. 318–342, 2006.

[37] T. C. Hu, "Multi-commodity network flows," *Operations research*, vol. 11, no. 3, pp. 344–360, 1963.

[38] G. Wang, G. Feng, S. Qin, and R. Wen, "Efficient traffic engineering for 5G core and backhaul networks," *J. of Comm. and Networks*, vol. 19, no. 1, pp. 80–92, 2017.

[39] N. Molner, A. De la Oliva, I. Stavrakakis, and A. Azcorra, "Optimization of an integrated fronthaul/backhaul network under path and delay constraints," *Ad Hoc Networks*, vol. 83, 2019.

[40] 3rd Generation Partnership Project, "3GPP TR 38.801 V14.0.0 :Study on new radio access technology ," 2017.

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

[42] 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.

[43] K. Katsalis, N. Nikaein, and A. Huang, "Jox: An event-driven orchestrator for 5g network slicing," in *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2018, pp. 1–9.

[44] 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.

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

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

[47] 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.

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

BIOGRAPHIES

Wesley da Silva Coelho (wesley.dasilvacoelho@orange.com) received B.E. degrees in Computer Engineering from Université Clermont-Auvergne, France, in 2017, and in Industrial Engineering from Universidade Federal de Minas Gerais, Brazil, in 2018. He also holds a M.Sc. degree in Models and Algorithms for Decision Support delivered by Université Clermont-Auvergne, France, in 2017. He obtained his Ph.D. in Computer Science and Communications at Conservatoire national des arts et métiers (Cnam), France, in 2021. His current interests cover network design, modeling, and optimization.

Amal Benhamiche (amal.benhamiche@orange.com) obtained a Ph.D. degree in Combinatorial Optimization from Paris-Dauphine University in 2013. She held a post-doctoral position at the French Alternative Energies and Atomic Energy Commission (CEA). She joined Orange Group in 2016 and is currently part of the Mathematical Models for Optimization and Performance Evaluation research department at Orange Labs, France, where she works on optimization problems for future networks.

Nancy Perrot (nancy.perrot@orange.com) received her Ph.D. degree in Applied Mathematics with a specialization in Operations Research from the University of Bordeaux1 in 2005. She joined Orange in 2005 to work as a researcher on several network optimization problems. Her current research interests are related to network virtualization, slicing in 5G networks, and security in complex systems. She supervised several M. Sc. and Ph.D. students. She is currently a member of the Orange expert community Future Networks, the manager of the Mathematical Optimization research activities at Orange, and the leader of the ANR MAESTRO-5G project on the management of slices in 5G networks.

Stefano Secci (stefano.secci@cnam.fr) is professor at Cnam, France, head of the Networks and IoT Systems team (https://roc.cnam.fr). He received a telecommunications engineering and a Ph.D. degree from Politecnico di Milano, and a dual Ph.D. degree from Telecom ParisTech. He was an associate professor at UPMC from 2010 to 2018, and previously worked for CNIT, Fastweb and Polytechnique Montréal. His current interests cover network automation and cybersecurity.