Network Slice Robustness with Function Sets

Nour-El-Houda Yellas, Jeongku Choi, Prosper Chemouil, Stefano Secci, Deep Medhi

Network Slice Robustness with Function Sets

Nour-El-Houda Yellas, Jeongku Choi, Prosper Chemouil and Stefano Secci
Cedric, Cnam
Paris, France
Email: firstname.lastname@cnam.fr

Deep Medhi
National Science Foundation
Alexandria, VA, USA
Email: dmedhi@nsf.gov

***Abstract*—Network slicing allows leveraging virtualization techniques for the creation of multiple, logically-isolated network instances over a shared infrastructure. In general, it is composed of a set of unique network functions with a specific physical capacity request. In order to improve network and service robustness, ETSI has introduced the concept of *Network Function Set* where network functions are replicated and deployed in different physical nodes. In this paper, we integrate this concept of the network function set to implement load-balancing and efficient resource distribution in 5G networks by leveraging on an existing network slice design formulation without the network function set. This helps relieve the burden on the nodes and links and prevent QoS degradation, in particular during failures. We describe how the baseline approach is impacted for the placement of network functions. We then show that our approach improves load balancing and latency with respect to the baseline solution.**

***Index Terms*—NSDP, Network Function (NF), Network Slicing (NS), NF set, load-balancing, resource distribution.**

I. Introduction

In the last decade, Network Functions Virtualization (NFV) and Software-Defined Networking (SDN) paradigms have been introduced to efficiently manage available resources, pushing back the barriers to the design of virtualized infrastructures on both access and infrastructure networks. This led to the emergence of network slicing, allowing to create multiple unique logical networks over a common multi-domain infrastructure [1]. In particular, network slicing is a key technology in the 5G context and beyond, as this helps meeting specific Service Level Agreements (SLAs) according to the nature of services, while ensuring different security levels [2].

A network slice is usually considered to serve different categories of services. Generally, the 5G core is isolated for each network slice while the radio-access part is shared among users. Note that other deployment scenarios are also possible, such as isolating both the radio-access subsystems and core data (DP) and control plane (CP) functions.

A Network Slice (NS) is commonly composed of one or several Network Functions (NFs) which are in turn composed of a set of Network Function Services (NFS). While most of the implementations considered so far are based on a composition of single Network Functions, Release 16 of ETSI Technical Specifications for the System architecture for the 5G System (5GS) [3] has opened the possibility to replicate network functions in different physical resources in order to achieve higher resilience. The aim is to better serve network slices in terms of performance, as well as to enhance reliability. A particular feature in this direction is the *Network Functions set*.

Our contribution here stems considering this feature, namely the Network Functions set, to replicate individual network functions in different locations. Our goal is to focus on the design of end-to-end network slices while considering the 5G systems requirements imposed by 3GPP. More specifically, we aim at analyzing how the overall Design Problem is impacted and at evaluating the benefits of using such a new capability to better serve network slices in terms of latency where each mobile user can be assigned to the closest NF, bandwidth where the load can be balanced amongst the physical entities, and reliability support thanks to NF redundancy. To do so, we extend an existing research work [4] where the problem is modeled as a Mixed Integer Linear Programming (MILP) in which we propose to enhance the NS provisioning considering network functions sets.

The rest of the paper is organized as follows. First we give the background regarding network slicing and the Network Slice Design Problem (NSDP). Next, we briefly present the initial design problem formulation and focus on how this model is extended to account for the existence of network function redundancy though the Network Functions sets. Then we analyze a variant of the initial solution in which we also aim at minimizing node loads. We thus consider a network topology and service requirements to evaluate the cost and performance of our approach with respect to the initial study. We also show that our extension of the design problem results in better load sharing of the overall network.

II. Background

In this Section, we provide the necessary background on the 5G system evolution towards supporting network slicing, resilience and robustness in network slice design.

A. 5G systems specificity

5G mapping entities: Virtualization paradigms such as NFV and SDN were proposed to efficiently manage available resources [5], pushing the barriers to the design of virtualized infrastructures at both the radio access and core networks, with a decoupling of the control plane and data plane. This led to the emergence of network slicing concept, designed for meeting the specific and heterogeneous requirements of applications [6], [7], [8]. In fact, it consists in partitioning the physical infrastructure into customized end-to-end logical

networks where a communication service can be mapped into one or multiple Network slices (NSs). Each network slice is a chaining of a set of network functions. These NFs are in charge of processing the traffic of one or multiple NSs. More specifically, network slices require the mapping of NFs to NFSs[1] where each NFS is a Virtual Network Function (VNF) component and implemented using virtualization solutions such as containers or virtual machines. A NFS can be a function from the radio-access network, the core network or any NF (e.g., proxy, firewall, etc.). Network slicing has attracted a lot of interest from both academia and industry where several challenges are addressed considering the management architecture [9], network design [4], [10], [11] and resource management [12], [13], [14], among others. In [9] the authors propose 6GLEGO which is an end-to-end orchestration framework for beyond-5G and 6G networks to overcome the limitations of the ETSI-NFV MANO orchestrator [15]. In this paper, our work focuses on the optimization of the allocation decision only without delving into the implementation of the orchestration solution.

Note that control plane (CP) NFs can communicate with each other, if authorized, to expose/consume a service from other CP NFs [3].

Flexible RAN (Radio Access Network) disaggregation: The physical base station is split into three units: the centralized unit (CU) in charge of packet processing functions and virtualized on commodity hardware, the distributed unit (DU) implementing base-band processing functions that can also be virtualized and the radio unit (RU) which ensures radio functions on specialized hardware. The decision on the placement and belonging of functions to either the CU or the DU is known as functional split [16]. Note that the 3GPP proposed different split options to adapt to various use cases in terms of latency and throughput. This solution can leverage on NFV paradigm to increase network efficiency and reduce deployment costs.

Sharing policy: NS isolation is an important challenge to tackle in 5G systems to ensure both security control and resource isolation to guarantee the Service Level Agreement (SLA) within each NS request. From the 3GPP specifications, different settings exist to address isolation between network slices. For example, isolation constraints can be (i) complete, where the NSs do (respectively, do not) share any network function, or (ii) partial, where NSs can share a subset of NFs.

Reliability support with NF sets: Several virtualized deployment scenarios supported by the 5G core have been introduced by ETSI in release 16 [17] which includes a ' *Network Function Set*'option. This concept allows several network function instances to be present within a NF set, in a distributed, scalable and redundant manner. In such a way, NF sets are introduced as a key enabler to support network reliability where alternative NFs can be used in case of failure or load balancing/re-balancing. To that aim, the NF service consumer is notified (i.e., by the Network Repository Function) when the NF instance producer is not available anymore, another NF producer is selected in that case. Moreover, the NFs within the same set may be deployed in different geographical locations while sharing the same context of data; they should be connected to the same data plane NF (i.e., User Plane Function) to allow the service context transfer.

[1]Note that in this paper, NFS refers to Network Function Service. No acronym is used for Network Function Set.

B. Resilient and robust network slice design

Performance of network slices is important from a mobile network operator point of view [18]. A broad range of research works addresses the resilience and robustness problems for network slice design [19], [20], [21]. For instance, authors in [21] propose a polynomial-time framework for end-to-end network slice orchestration while considering VNF placement and traffic routing problems. They propose to re-use already deployed VNFs for new NS requests and they consider availability and reliability metrics to evaluate the proposed solution. In [22], authors take into consideration the geographical dimension in solving the placement and chaining of VNFs where the placement decision of VNFs is based on node and link capacity as well as the maximum end-to-end delay imposed by each slice request. Authors in [14] tackle the NS resource management with a dynamic selection of NS requests in an edge environment. The main goal is to minimize the completion time while dynamically assigning computation and radio resources for NS requests. The authors in [23] tackles the VNF placement problem for slice allocation in 5G core networks while considering reliability issues such as physical isolation between VNFs within each slice and end-to-end delay constraints. The slice allocation at the RAN is not covered in this work. In [24], the authors propose a mathematical model for the network slice design problem. The nominal problem is extended to include traffic robustness in order to cope with traffic variation, in addition to a survivability aspect against single node or link failures. In contrast to traditional approaches where network slice allocation is formulated as an optimization problem, in [25] the authors present data-centric solutions using AI techniques where they discussed the different challenges related to the use of deep reinforcement learning-based approaches for NFS placement and scaling.

III. Network Slice Design

We consider the problem of designing network slices in 5G and beyond-5G systems. Our main objective is to ensure both resilience and load-balancing by design. More precisely, we extend the network slice design proposed in [4] to integrate the novel concept of '**Network Function Sets**', recently presented in [17]. We aim at finding the best placement of deployed NFs to serve a set of network slice requests while defining the necessary number of NFSs copies to be packed into NFs. Note that this work considers network functions from both radio-access and core networks.

In the following, we first present the problem of network slice design. Then, we introduce the NF sets concept.

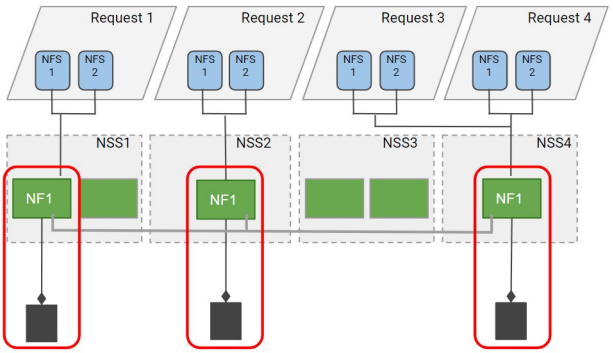

Fig. 1: Diagram of NF duplication with network function sets.

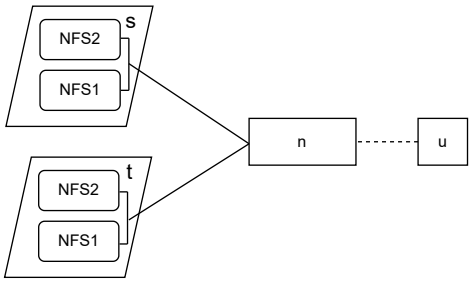

Fig. 2: Two slices s and t sharing the same NF n and installed on node u.

A. Problem Definition

In the reminder, we consider a directed graph G_p to represent the physical network. Let V_p be the set of nodes and A_p the set of arcs. We consider three types of nodes: access nodes where distributed network functions can be deployed, denoted by V_p^{du}, non-access nodes where centralized (i.e., shared) NFs can be deployed, denoted by V_p^{acc} and application nodes where application servers are installed, denoted by V_p^{app}. We assume that physical nodes have heterogeneous CPU resources. Let C_u be the available CPU resources on node $u \in V$. Also, we denote by $a \in A$ the arc connecting the pair of nodes u and v with $(u, v) \in A$. Lastly, b_a is the bandwidth capacity on arc a, while d_a is the latency on arc a.

In contrast, the virtual layer refers to the deployment of network slice requests and it is represented by a directed graph G_v. We denote by S the set of network requests. We consider that each network slice can be composed of one or more network slice subnets (NSS). Each NSS is composed of one or more network functions with N as the set of available NFs. Each NF $n \in N$ is, in turn, composed of a set of NFSs. We denote by F the set of NFS types, F^d the set of data-plane NFSs and F^c the set of control-plane NFSs. Note that the data-plane NFSs are chained in a specific order; each NFS $f \in F$ requires certain physical resources (e.g., CPU) with a processing capacity. In this work, we only consider CPU resources where c_f is the number of CPU needed by NFS f. Furthermore, the processing capacity of NFS f is denoted by $cap(f)$ with b_f is the expected data rate received by a physical node from one User Equipment (UE). A maximum delay d_{fg} should not be exceeded between NFSs f and g. Finally, we denote by λ_f the compression coefficient applied by the data-plane NFS f.

Moreover, each NS request s is associated with a graph $G_s = (V_s, A_s)$ where V_s is the subset of NFs serving the slices and A_s are the arcs connecting two nodes from V_s. Also, each slice request s is associated with a set $K(s)$ of traffic demands defined by a source node $o_k \in V_p^{du}$ and a target node $t_k \in V_p^{acc}$. Given a slice request s and its corresponding traffic demands $K(s)$, we define b_k as the initial data rate sent by the origin node o_k, d_s as the maximum accepted end-to-end latency to be guaranteed within the slice s and n_s the expected number of users served by the slice s.

Another important characteristic is the isolation between NS requests. We denote by q_{fg}^{st} the possibility of packing two NFSs $f, g \in F$ from respectively, NS request s and t into the same NF. In other words, q_{fg}^{st} is equal to 1 if NS request s accepts packing NFS f with NFS g from NS request t into the same network function.

We describe the network slice design optimization problem as a NFS orchestration and packing problem that aims at finding the optimal placement, mapping and routing of network slices onto NFs embedded in the network while finding a trade-off between load balancing over physical nodes and NF replication to ensure resilience in case of failures, and the computation cost minimization.

B. NF Set in network slice design

In contrast to [4] where each NF is present on only one physical node, in this work we integrate the possibility to duplicate a NF, that is, the same NF type can be installed on different physical nodes as shown in Figure 1. The red rectangles in the figure refer to the network functions that belong to the same NF set. Note that the NF sets are not uniform, that is, the number of NF copies can differ from one NF type to another.

The difference between NF duplication and NF scaling is that the former allows NFs to be installed on multiple physical nodes and it can multiply the capacity of a whole NF whereas scaling is limited by the capacity of NFS and is controlled by the scaling factor based on the initial allocation capacity.

We assume that a NF n can serve one or multiple slices. In Figure 2, we show an example of two NS requests sharing the same NF n where each NS request has its own NFSs; this case is also possible in the original network slice design problem presented in III-A. However, figure 3 shows that two NS requests s and t are using different instances from equivalent NFs (i.e., same NF type n) which is not ensured by the original NS design problem. Note that the sharing policies among network slices are defined using the parameter q_{fg}^{st} that is equal to 1 if NFS f from slice request s can be packed with NFS g from slice request t in the same NF.

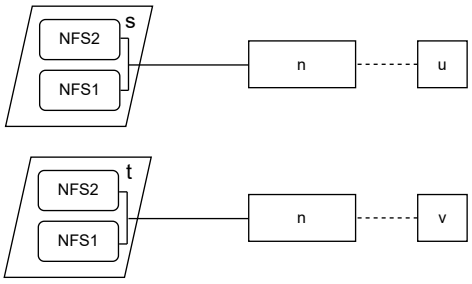

Fig. 3: NF n duplicated and used by different slices s and t. Each NF copy is installed on a different node (u and v).

1) Flow Scenario: In figure 4, we model three network slice requests. Each network slice may accept or not sharing network functions with the two other network slices.

More precisely, subscriber 1 shares NF1 and NF3 with subscribers 2 and 3. In contrast, subscribers 2 and 3 share NF1, NF2 and NF3. In this example, we model two NF sets containing equivalent NFs. The first one is composed of two NF2 instances (rectangle in red) and the second one is composed of three instances of NF4 (rectangle in blue).

Figure 5 depicts an example of solution for the above instance with 3 NS requests, 3 demands, 5 NFs and 5 NFSs. NF1 packs the NFS f1 from each slice request and it is deployed on one physical node. Moreover, two NF2 instances are deployed: the first one is packing two NFSs of type f2, one serving the network slice request from subscriber 2 while the other are serving the network slice requests from subscriber 3, and the second one is composed of one NFS of type f2 and serves the demand from subscriber 1. NF3 packs NFS f3 from the three NS request. Each instance of NF4 is serving only one slice request. Finally, NF5 is used only in NS request 3.

Copies of the same NF can communicate with each other: the dotted line shows this communication. This communication allows the service context transfer.

Note that for the sake of simplicity, we do not model the connections between equivalent NFs in our problem formulation. We consider that all NF copies from the same type can communicate with each other [17]. Also, we suppose that NFs from both data-plane and control-plane can be duplicated.

IV. Mathematical Model

In the following, we present the mathematical model that corresponds to the original network slice design problem, then we introduce the mathematical formulation of the NF set integration.

Table I defines the decision variables used in the core mathematical model of NSDP.

We use two binary variables to represent the placement and packing related decision: z_f^s assumes value 1 if NFS f from slice request s is centralized, and x_{nu}^{sf} is equal to 1, if the NFS f from slice request s is packed into NF n and installed on node u. The real variables w_{nu}^{sf} are introduced to model the amount of NFSs of type f that are serving the slice request s, packed into NF n and installed on node u. Finally, the integer variable y_{nu}^{f} represents the total number of NFSs of type f that are packed into NF n and installed on node u

A. NSDP core model constraints

Our mathematical model is based on the one presented in [4]. For the sake of space, we only present here the most relevant constraints related to, for instance, NF placement and NFS packing and dimensioning. Constraints related to split selection, connectivity, latency and link can be found in [4].

1) Split Selection: As already mentioned, Data-Plane (DP) NFSs are chained in a specific order. Constraints 1 represent whether NFS f is installed locally or centrally as the placement of a DP NFS impacts the placement of the next ones. Note that if the NFS f is centralized, $f+1$ should be centralized too. These constraints result in the decision of the functional split option [16].

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

2) NFS Placement: Constraints 2 ensure that all distributed NFSs of type f that are packed into NF n should be installed on the origin node of the NS request. Note that a control-plane (CP) NFS can not be installed in an origin node, as we consider that only data-plane NFSs are installed in the distributed nodes (i.e., CP NFS are centralized).

Constraints 3 ensure that all copies of the same centralized NFSs type will be installed in the same physical node.

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

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

	NSDP variables
z_f^s	binary, equal to 1, if NFS f from slice request s is centralized; 0 otherwise
x_{nu}^{sf}	binary, equal to 1, if the NFS f from slice s is packed into NF n and installed on physical node u; 0 otherwise
w_{nu}^{sf}	real, represents the amount of NFS f serving slice s, packed in NF n and installed on physical node u
y_{nu}^{f}	integer, represents the total number NFSs of type f packed into NF n and installed on physical node u
γ_{fg}^{ka}	binary, equal to 1 if the traffic of demand k uses arc a to route the flow between NFSs f and g; 0 otherwise
	NF set integration variables
ϕ_n^{st}	binary, equal to 1, if the NF n is admitted to share from slice s and slice t
ψ_{nu}^{s}	binary, equal to 1, if the NF n from slice s is installed on physical node u
ξ_n^{st}	binary, equal to 1, if slice requests s and t admits to share NF n

TABLE I: Decision variables.

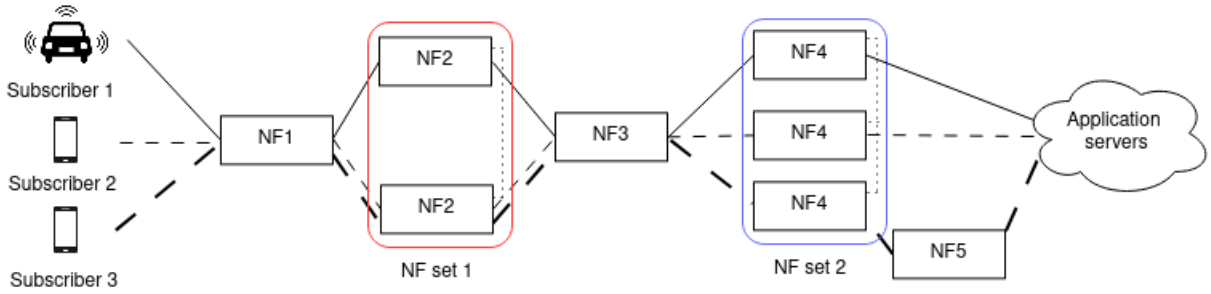

Fig. 4: Flow example of the network slice design problem with NF sets integration.

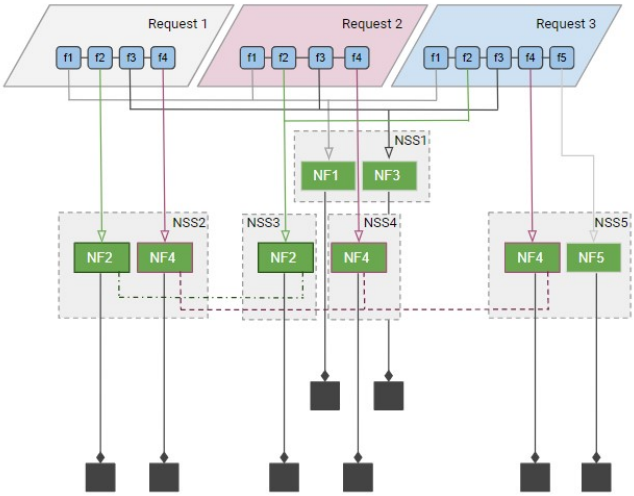

Fig. 5: Example of solution of the network slice design problem with NF sets integration.

3) NF Dimensioning: The quantity (4) calculates the exact amount of NFSs for each NS request. This value can be a fractional value to be considered when the sharing between slices is possible.

$$cap(f)w_{nu}^{sf} = \begin{cases} \sum\limits_{k \in K(s)|u=o_k} \lambda_{f-1} b_k x_{nu}^{sf} & \text{, if } f \in F^d \\ \eta_s b_f x_{nu}^{sf} & \text{, if } f \in F^c \\ \sum\limits_{k \in K(s)} \lambda_{f-1} b_k x_{nu}^{sf} & \text{, if } f \in F^d, u \in V_p^{ac}. \end{cases} \quad (4)$$

$$\forall s \in S, \forall f \in F, \forall n \in N, \forall u \in V_p$$

4) NFS Packing: Constraints (5) define the sharing policy among network slices. Note that theses isolation constraints are imposed by each NS request. For instance, when two slice requests s and t allow sharing NFSs f and g (i.e., f and g are packed into the same NF n), these NFSs are installed on the same physical node.

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

Constraints (6) define the amount of NFS f used by all NS requests based on (4). More precisely, when two NS requests s and t admit sharing the same NF n (i.e., their NFSs can be packed into n), then the total number of NFSs of type f required to serve the two slices is equal to $\lceil (w_{nu}^{sf} + w_{nu}^{tf}) \rceil$. However, when s and t do not admit sharing the same NF n, the total number of installed NFSs of type f is equal to $\lceil w_{nu}^{sf} \rceil + \lceil w_{nu}^{tf} \rceil$

$$\sum_{s \in S} w_{nu}^{sf} \leq y_{nu}^{f} \qquad \forall n \in N, \forall u \in V, \forall f \in F \quad (6)$$

Based on 6, we can calculate the total amount of NF n as follows:

$$The\ amount\ of\ n = \sum_{u \in V_p} \sum_{f \in F} y_{nu}^{f} \quad , \forall n \in N \quad (7)$$

5) Physical node capacity: Constraints 8 ensure that the total amount of NFSs installed on node u do not exceed its physical capacity. This capacity constraints on physical nodes are considered at the NF level, which depends directly on the physical capacity required by the NFSs of each NF.

$$\sum_{n \in N} \sum_{f \in F} c_f y_{nu}^{f} \leq C_u \quad , \forall u \in V_p \quad (8)$$

6) Physical link capacity: Constraints 9 ensure that a link can not deliver more data than its capacity. We consider the comprehension coefficient λ in these constraints.

$$\sum_{s \in S} \sum_{k \in K(s)} b_k (\lambda_{f_{|F^d|}} \gamma_{f_{|F^d|} f_0}^{ka} + \sum_{f \in \{f_0\} \cup F^d \setminus \{f_{|F^d|}\}} \lambda_f \gamma_{ff+1}^{ka}) \quad (9)$$

$$+ \sum_{s \in S} \eta_s \Big(\sum_{(f,g) \in F(s)} b_{fg} \gamma_{fg}^{1a} + \sum_{(f,g) \in G(s)} \sum_{k \in K(s)} \frac{b_{fg} \gamma_{fg}^{ka}}{|K(s)|} \Big) \leq b_a \quad , \forall a \in A_p$$

7) Latency: Constraints 10 ensure that the end-to-end latency of each traffic demand k of NS request s is lower or equal to the maximum accepted end-to-end latency d_s. This latency includes only the data-plane exchanges on the uplink direction. Constraints 11 ensure that the data between NFSs f and g should not exceed its capacity, for both data and control planes.

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

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

B. NF set integration

In this section, we present the set of constraints and objectives that we proposed to ensure NF duplication. Indeed, the following constraints are integrated within the original mathematical formulation presented in IV.

For enabling the NF set in the network slice design model presented in IV-A, we introduce the following binary variables. Variable ϕ^{st}_n assumes value 1, if the NF n is shared among slice requests s and t. ϕ^{st}_n assumes value 1 if a NF n packs at least one NFS that can be shared between network slices s and t. The difference between variable ϕ^{st}_n and parameter q^{st}_{fg} is that the former refers to NFs that are already deployed (i.e., x^{sf}_{nu} = 1 and x^{tg}_{nu} = 1).

Variable ψ^{s}_{nu} assumes value 1, if the NF n from slice s is installed on the physical node u. This variable allows us to determine if NF n is present (respectively, not present) in slice s . Finally, variable ξ^{st}_n assumes value 1, if the network slice requests s and t admit sharing the NF n.

$$\phi^{st}_n = \begin{cases} 0 & \text{, if } \sum_{f \in F} \sum_{g \in F} q^{st}_{fg} x^{sf}_{nu} = 0 \\ 1 & \text{, if } \sum_{f \in F} \sum_{g \in F} q^{st}_{fg} x^{sf}_{nu} \neq 0 \end{cases} \quad \forall s, t \in S, \forall n \in N, \forall u \in V_p \tag{12}$$

$$\psi^{s}_{nu} = \begin{cases} 0 & \text{, if } \sum_{f \in F} x^{sf}_{nu} = 0 \\ 1 & \text{, if } \sum_{f \in F} x^{sf}_{nu} \neq 0 \end{cases} \quad \forall s \in S, \forall n \in N, \forall u \in V_p \tag{13}$$

ξ^{st}_n is used to relate the two variables ϕ^{st}_n and ϕ^{ts}_n. In other words, ξ^{st}_n assumes value 1 if NF n packs NFSs that can be shared among NS requests s and t. It can be expressed as follows.

$$\xi^{st}_n = \phi^{st}_n \cdot \phi^{ts}_n \qquad \forall n \in N, s, t \in S \tag{14}$$

As constraints (14) are not linear, we replace them with the following linear constraints.

$$\xi^{st}_n \leq \phi^{st}_n \qquad \forall n \in N, s, t \in S \tag{15}$$

$$\xi^{st}_n \leq \phi^{ts}_n \qquad \forall n \in N, s, t \in S \tag{16}$$

$$\xi^{st}_n \geq \phi^{st}_n + \phi^{ts}_n - 1 \qquad \forall n \in N, s, t \in S \tag{17}$$

We add the two following constraints to express the possibility of duplicating NFs.

$$\psi^{s}_{nu} + \psi^{t}_{nu} \leq 1 + \phi^{st}_n \phi^{ts}_n \qquad , \forall s, t \in S, \forall n \in N, \forall u \in V_p \tag{18}$$

$$\psi^{s}_{nu} + \psi^{t}_{nv} \leq 2 \qquad , \forall s, t \in S, \forall n \in N, \forall u, v \in V_p : u \neq v \tag{19}$$

Constraint (18) allows network slice requests s and t to use the same NF n if they both allow sharing at least one NFS packed within n (see figure 2).

Constraint 19 allows to duplicate a given NF n on different physical nodes even when serving the same slice. Currently, there is no limitation on NF duplication. However, this limitation can be easily added to the current model (see figure 3).

C. Objective formulation

The objective in (21) is to reach a trade-off between the minimization of the computation cost and the load on the physical nodes and links. The first term is considered to minimize the computational cost and the second term is introduced to minimize the maximal load on the physical nodes while the third term aims at reducing the physical link utilization. Parameters α and β are set between 0 and 1 for not affecting the minimization of the computation cost.

In order to minimize the load on the physical nodes, we also introduce the variable $maxC$ in constraint (20) that allows to compute the maximum load on physical nodes. Note that the term $maxC$ is added to enforce the NF duplication: adding the NF set-related constraints to the current model does not allow the duplication of NFs as the original objective aims to minimize the computational costs.

$$maxC \geq \frac{\sum_{f \in F} \sum_{n \in N} c_f y^f_{nu}}{C_u}, \qquad \forall u \in V_p \tag{20}$$

The goal of this model is:

$$\min(\sum_{f \in F} \sum_{n \in N} \sum_{u \in V} y^f_{nu} + \alpha \times maxC + \beta \sum_{a \in A} \sum_{s \in S} \sum_{k \in K(s)} \sum_{f,g \in F} \gamma^{ka}_{fg}) \tag{21}$$

In contrast to the original network slice design problem proposed in [4], our mathematical model includes the possibility to deploy several instances of the same network function to ensure load-balancing and reliability in case of node failures.

V. Numerical Evaluation

We solve the network slice design problem using the following approaches:

- NSDP: using the original model from [4] (equations (2) to (11) in addition to connectivity-related constraints

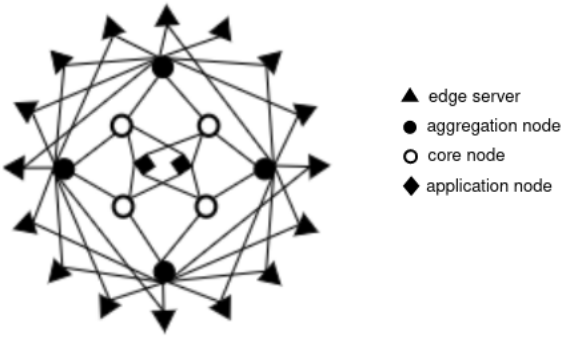

Fig. 6: Mandala topology [4].

in [4]) where the term regarding the load balancing is removed from the objective;

- NSDP-set: our proposal, represented by equations (2) to (21), in addition to connectivity-related constraints from [4];

We use the Mandala topology proposed in [4]: it consists of connecting access nodes through three tiers, i.e., aggregation, core and application nodes. It corresponds to a metropolitan area network topology (see figure 6). This topology was chosen in order to compare the results in the same setting as the baseline approach. The total number of nodes is equal to 26 with 96 unidirectional links.

Each physical node is equipped with 480 CPU units. The delay on the links is randomly generated: between 50μs and 100μs for fronthaul links, 200μs and 300μs for backhaul links, and 400μs and 600μs for core links [4]. The bandwidth capacity is equal to 500% of the maximum data volume sent by one access node.

Regarding the virtual layer, we consider 1 NFS to deploy the radio access network (only DP NFSs) and 4 NFSs for the core network. We consider 4 mandatory NFSs that should be present in each slice and 1 optional NFS. We generate 4 NS requests with the following requirements: an end-to-end latency of 100 ms, 4800 connected UEs with 3 Gbps per UE. Also, we consider 8 demands per each NS request. In addition, we consider that each NFS requires 5% of the physical node CPU capacity. The compression coefficient of the first DP NFS is equal to 65% and 40% for the second DP NFS. The latency between DP NFSs is randomly generated between 100μs and 200μs. Finally, we consider that all slices can share NFSs among each other. We generated 10 different instances while randomly generating the delays on the physical links.

We implemented our model in python and we used Cplex 22.11 as a linear solver. Other software platforms like Open-MANO, Open Source MANO (OSM) and OPNFV could be used to implement this solution; meanwhile our main goal is to optimize the provisioning of the NFs rather than focusing on the implementation challenges.

In the following, we analyze the quality of the solution based on the following aspects: (i) the average latency, (ii) the load on the physical nodes and (iii) the load on links.

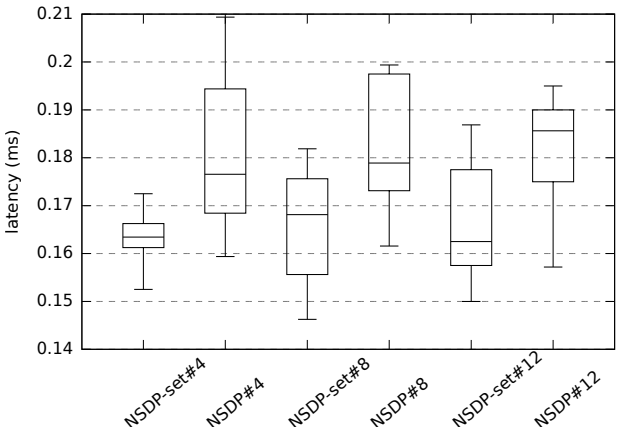

Fig. 7: Average latency.

A. Latency results

Figure 7 presents the distribution of the average latency on the links using different number of NFSs (i.e., available NFS copies of the same type). We observed that:

- Our approach reduces the average latency for all NFS number cases when compared to the original NSDP. Having multiple NF instances allows to connect a set of NFs that are close to each other. For example, when two NF instances are present in two different nodes, the traffic from a given UE can flow through the NF installed on the closest physical node.
- The difference in latency for a given case (e.g., NSDP-set4) is also related to the delays on arcs as we change this parameter for each instance.
- With our approach, the latency increases with the number of available NFS copies. This can be explained by the fact that, when the number of copies is low (i.e., 4), our approach tends to increase the number of NF instances to a certain extent, either for load balancing (i.e., see objective in formulation (21)) or for scaling. The more we increase the NF instances, the lower the delay is between chained NFs composing a network slice and consequently the end-to-end delay is lower.

B. Traffic load results

Figure 8 depicts the distribution of the total load on the physical nodes and links, w.r.t the node capacity. Regarding the nodes load, we observed that:

- As expected, our approach reduces the total load on the physical nodes as the main objective is to balance the load on the physical infrastructure. This is made possible thanks to the possibility of activating more than one instance of a NF type and consequently, splitting the traffic on different nodes. Meanwhile, the total load node with NSDP is higher: this can be explained by the fact that a NF is only present on one node where it should process all the traffic. Note that for NSDP, only the

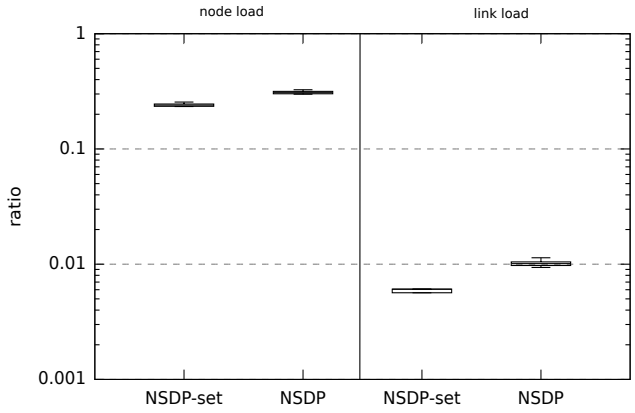

Fig. 8: Distribution of the total load on physical nodes and links.

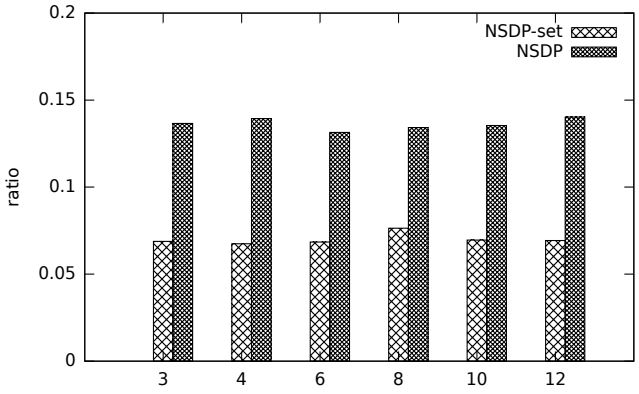

Fig. 9: Average of most loaded physical links.

scaling out operation is considered (i.e., increasing the number of NFS copies).

- Despite the increased costs that can be generated by our approach (i.e., the number of used nodes gets higher), these results show its robustness where it helps to recover in case of node failure as a copy of the NF installed on the broken node is already installed on another node. For simplicity reasons, communications between NF instances are not modeled in this work.

Regarding the link load, we observed that:

- The load on the links was reduced by up to 50% when compared to the original NSDP. This can be explained by the fact that the number of used arcs in our approach is reduced. In other words, as NSDP does not allow NF duplication and where the main goal is to reduce the computation cost (i.e., number of NFS copies that are used), the placement decision is related to the NFS minimization. In that case, NFs can be installed on distant nodes and consequently, the number of arcs that are used is higher which increases the total load on the links.
- Nevertheless, our approach allows the deployment of the same NF copy on several nodes where the placement of chained NFs on nodes that are close to each other is more likely to happen.

Figure 9 shows the average of the highest load on the physical links for different numbers of NFSs. We observed that our approach reduces the highest load on the links by up to 50% when compared to NSDP. We also notice a slight difference when increasing the number of available NFSs for the two approaches.

Regarding our approach (i.e., NSDP-set), we observed that when the number of NFSs is decreased (i.e., 4) or increased (i.e., 12), the most loaded links have lower load when compared to the case with 8 NFSs. This can be explained as our approach aims at reaching a trade-off between the number of used NFSs and the number of deployed NF instances to reduce the computation cost. In other words, when the number of NFSs is low, our approach increases the number of deployed NF instances. When the number of NFSs increases, the number of deployed NF instances is decreased to a certain extent which can generate a higher load on the links as the amount of traffic exchanged is higher (i.e., see the case with 8 NFSs). A higher increase of the number of NFS does not necessarily decrease the number of NF instances as for some NFs the duplication is not activated and the number of NFS copies is increased. Increasing the number of NF instances for the other NFs may be applied to reduce the total cost which increases with the NFS copies and the NF instances.

Finally, though the proposed approach is based on an offline solution for the network slice design provisioning and placement, increased robustness is exhibited in the case of node failure since traffic can be re-routed to another node where an instance of the same NF is installed. As already explained, the NF (service) consumer can detect the non availability of a NF producer allowing it to redirect its traffic to another NF producer (i.e., a duplicated NF).

VI. Conclusion

In this paper, we tackled the robustness problem in network slice provisioning. We relied on the concept of 'Network Function Sets'that was recently proposed by ETSI for reliability support in 5GC virtualized environments. We integrated the NF sets option to a network slice design problem from the state-of-the-art where we explicitly showed the integration of NF sets to the existing formulation. Despite that the proposed approach entails an additional deployment cost due to a higher number of deployed NFs, numerical results for the cases we studied show its capability to reduce the average latency by more than 50% in addition to the load minimization on both physical nodes and links.

Future works may focus on post-failure analysis while emphasizing the resilience aspect. Data-centric solutions using AI techniques as suggested in [25] where deep reinforcement learning-based approaches may be considered for more realistic scenarios.

ACKNOWLEDGEMENTS

This work was partly funded by the ANR HEIDIS (contract nb: ANR-21-CE25-0019; https://heidis.roc.cnam.fr) and the ENE5AI projects.

D. Medhi's work was supported by the National Science Foundation (NSF), a United States Federal Government Agency, while employed by NSF under NSF's Independent Research/Development (IRD) program. Thus, this copyrightable work is subject to a royalty-free, irrevocable, nontransferable, nonexclusive license for or on behalf of the United States Government to reproduce, perform, translate and otherwise use and authorize others to use such materials for governmental purposes. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Description of Network Slicing Concept. NGMN Alliance, V1.0, Jan. 2016. Available: https://www.ngmn.org/wp-content/uploads/160113_NGMN_Network_Slicing_v1_0.pdf.

[2] 3GPP TR 23.700-40 V17.0.0, "Study on enhancement of network slicing 5G (Release 17)"; Tech. Spec. Group Services and System Aspects, 2021.

[3] 3GPP TS 29.503 V16.0.0, 5G System; Unified Data Management Services (Release 16), Tech. Spec. Group Services and System Aspects, 2019.

[4] W. da Silva Coelho, A. Benhamiche, N. Perrot and S. Secci, "Function Splitting, Isolation, and Placement Trade-Offs in Network Slicing," in *IEEE Transactions on Network and Service Management*, vol. 19, 2022.

[5] 3GPP TR 28.801 V15.1.0, Telecommunication Management; Study on Management and Orchestration of Network Slicing for Next Generation (Release 15), Tech. Spec.Group Services and System Aspects, 2018.

[6] A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, 5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges, Computer Networks, vol. 167, p. 106984, Feb. 2020.

[7] A. Kaloxylos, A Survey and an Analysis of Network Slicing in 5G Networks, IEEE Commun. Stand. Mag., vol. 2, no. 1, pp. 6065, Mar. 2018.

[8] L. M. Contreras, Slicing challenges for operators, in Emerging Automation Techniques for the Future Internet, M. Boucadair and C. Jacquenet (Eds.), IGI Global, 2019.

[9] S. Kuklinski, L. Tomaszewski, R. Kolakowski and P. Chemouil, "6G-LEGO: A framework for 6G network slices," in Journal of Communications and Networks, vol. 23, no. 6, pp. 442-453, Dec. 2021.

[10] G. Wang, G. Feng, S. Qin, R. Wen and S. Sun, "Optimizing Network Slice Dimensioning via Resource Pricing," in IEEE Access, vol. 7, pp. 30331-30343, 2019.

[11] H. Li et al., "Slice-Based Service Function Chain Embedding for End-to-End Network Slice Deployment," in *IEEE Transactions on Network and Service Management*, vol. 20, no. 3, pp. 3652-3672, Sept. 2023.

[12] Q.-T. Luu, S. Kerboeuf and M. Kieffer, "Admission Control and Resource Reservation for Prioritized Slice Requests With Guaranteed SLA Under Uncertainties," in *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 3136-3153, Sept. 2022.

[13] A. Thantharate and C. Beard, "ADAPTIVE6G: Adaptive Resource Management for Network Slicing Architectures," in Current 5G and Future 6G Systems. J. Netw. Syst. Manage. 31, 1, Jan 2023.

[14] S. Josino and G. Dan. "Joint wireless and edge computing resource management with dynamic network slice selection,". *IEEE/ACM Transactions on Networking*, vol. 30, no 4, p. 1865-1878, 2022.

[15] Management and orchestration; architecture framework, v16.7.0, 3GPP TS 28.533, Apr. 2021.

[16] L. M. Larsen, A. Checko, and H. L. Christiansen, "A survey of the functional splits proposed for 5G mobile crosshaul networks," *IEEEComm. Surveys Tutorials*, vol. 21, no. 1, pp. 146172, 2018.

[17] ETSI TS 123 501 V16.7.0((3GPP TS 23.501 version 16.7.0 Release 16)), 2021.

[18] S. Kuklinski, and L. Tomaszewski, "Key Performance Indicators for 5G network slicing," in NetSoft 2019, pp. 464-471, Paris, France, 2019.

[19] J. Zheng, A. Banchs and G. d. Veciana, "Constrained Network Slicing Games: Achieving Service Guarantees and Network Efficiency," in *IEEE/ACM Transactions on Networking*, vol. 31, no. 6, pp. 2698-2713, Dec. 2023.

[20] R. Wen et al., "On Robustness of Network Slicing for Next-Generation Mobile Networks," in *IEEE Transactions on Communications*, vol. 67, no. 1, pp. 430-444, Jan. 2019.

[21] J. Martin-Perez, F. Malandrino, C.-F. Chiasserini, et al. "OKpi: All-KPI network slicing through efficient resource allocation". In : IEEE INFOCOM 2020.

[22] J. J. A. Esteves, A. Boubendir, F. Guillemin and P. Sens, "Location-based Data Model for Optimized Network Slice Placement," IEEE NetSoft 2020, pp. 404-412, Ghent, Belgium, 2020.

[23] D. Sattar and A. Matrawy, "Optimal Slice Allocation in 5G Core Networks," in IEEE Networking Letters, vol. 1, no. 2, pp. 48-51, June 2019.

[24] A. Baumgartner, T. Bauschert, A. M. C. A. Koster and V. S. Reddy, "Optimisation Models for Robust and Survivable Network Slice Design: A Comparative Analysis," GLOBECOM 2017 - 2017 IEEE Global Communications Conference, pp. 1-7, Singapore, 2017.

[25] N. Saha, M. Zangooei, M. Golkarifard and R. Boutaba, "Deep Reinforcement Learning Approaches to Network Slice Scaling and Placement: A Survey," IEEE Communications Magazine, 61(2), 82-87, Feb.2023.