A Two-Stage Reconfiguration in Network Function Virtualization: Toward Service Function Chain Optimization

Karcius D. R. Assis, Raul C. Almeida Jr., Hojjat Baghban, *Member, IEEE*, Alex F. Santos, and Raouf Boutaba, *Fellow, IEEE*

***Abstract*—Network Function Virtualization (NFV), as a promising paradigm, speeds up the service deployment by separating network functions from proprietary devices and deploying them on common servers in the form of software. Any service in NFV-enabled networks is achieved as a Service Function Chain (SFC) which consists of a series of ordered Virtual Network Functions (VNFs). However, migration of VNFs for more flexible services within a dynamic NFV-enabled network is a key challenge to be addressed. Current VNF migration studies mainly focus on single VNF migration decisions without considering the sharing and concurrent migration of VNF instances. In this paper, we assume that each deployed VNF is used by multiple SFCs and deal with the optimal placement for the contemporaneous migration of VNFs based on the actual network situation. We formalize the VNF migration and SFC reconfiguration problem as a mathematical model, which aims to minimize the VNF migration between nodes or the total number of core changes per node. The approach is a two-stage MILP based on optimal order to solve the reconfiguration. Extensive evaluation shows that the proposed approach can reduce the change in terms of location or number of cores per node in a 6-node and 14-node networks while ensuring network latency compared with the model without reconfiguration.**

***Index Terms*—Service function chain, reconfiguration, optimization, network function virtualization, VNF migration.**

Received 6 June 2024; revised 21 November 2024 and 13 February 2025; accepted 20 April 2025. Date of publication 9 May 2025; date of current version 7 August 2025. The authors would like to express their gratitude to the financial support from Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) - Brasil. This study was financed in part by the Fundação de Amparo a Ciência e Tecnologia do Estado de Pernambuco (FACEPE). This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001. We also thank the institutional support of the Federal University of Bahia (UFBA), the Federal University of Pernambuco (UFPE), the State University of Feira de Santana (UEFS), and the National Science and Technology Council (NSTC), Taiwan, project No. 113-2222-E-182-001. The associate editor coordinating the review of this article and approving it for publication was P. Papadimitriou. *(Corresponding author: Karcius D. R. Assis.)*

Karcius D. R. Assis is with the Electrical and Computer Engineering Department, Federal University of Bahia, Salvador 40210-630, Brazil (e-mail: karcius.assis@ufba.br).

Raul C. Almeida Jr. is with the David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada, and also with the Department of Electronics and Systems, Universidade Federal de Pernambuco, Recife 50670-901, Brazil.

Hojjat Baghban is with the Department of Artificial Intelligence, Chang Gung University, Taoyuan 33302, Taiwan, and also with the College of Intelligent Computing, Chang Gung University, Taoyuan 33302, Taiwan (e-mail: hojjat.baghban@cgu.edu.tw).

Alex F. Santos is with the Center of Science and Technology in Energy and Sustainability, Federal University of Reconcavo of Bahia, Cruz das Almas 44380, Brazil, and also with the Computer Science, State University of Feira de Santana, Feira de Santana 44036-900, Brazil (e-mail: alex.ferreira@ufrb.edu.br).

Raouf Boutaba is with the David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada.

Digital Object Identifier 10.1109/TNSM.2025.3567906

I. Introduction

IN MODERN networking, network function virtualization (NFV) has transformed how network services are provisioned and managed by decoupling network functions from proprietary hardware, allowing them to be deployed in software-based environments. Central to the NFV paradigm are the service function chains (SFCs), which specify the sequence of network functions traversed by data packets to fulfill a particular service request [1], [2].

Service function chain reconfiguration (SFC-R) is essential in adapting to evolve service requirements, fluctuating network conditions, and variable resource availability [3]. SFC-R involves optimizing resource allocation, virtual function migration, and traffic routing to maintain network efficiency and service quality. By dynamically reconfiguring SFCs, operators can improve agility and respond to traffic fluctuations and security threats without service interruption [4].

Despite its advantages, SFC-R faces practical challenges. Traditional static configurations are inadequate for today's dynamic environments, where applications, ranging from cloud services to IoT, constantly demand adaptable infrastructure. VNF sharing, where virtualized network functions (VNFs) are allocated across multiple services, is particularly beneficial but introduces further complexity to ensure efficient, scalable, and resilient reconfiguration processes.

Dynamic SFC-R and VNF sharing are critical in network services orchestration. For instance, in media streaming Services [5], sudden spikes in traffic during live events or viral content dissemination require the network to dynamically scale up VNFs such as load balancers or transcoding services. Without SFC-R, these spikes can overwhelm static resource allocations, causing buffering and poor user experiences. In addition, in IoT-enabled Smart Cities [6], [7], [8], applications such as traffic management or energy distribution rely on real-time data processing. As the number of connected devices grows, the network must dynamically reconfigure SFCs to maintain low-latency responses and optimize resource utilization, ensuring that virtualized functions like traffic monitors

or data analyzers are efficiently shared. On the other hand, in Enterprise Security Services, VNFs related to cybersecurity services like firewalls and intrusion detection systems must quickly adapt to emerging threats [9]. The sharing of VNFs between multiple SFCs allows operators to allocate security resources on-demand, dynamically reconfiguring based on real-time traffic patterns and attack detection [7]. The referenced use cases, including but not limited to those mentioned, highlight the practical necessity of an efficient SFC reconfiguration process to ensure service continuity, scalability, and optimal resource utilization.

Conventional static approaches to SFC-R are manual and reactive, relying on network operators to update configurations based on observed changes. This process is time-consuming and error-prone, leading to service disruptions. Additionally, static configurations often result in suboptimal resource utilization, with over-provisioning during low-demand periods and under-provisioning during high traffic hours [10], [11]. In contrast, automated and optimized SFC-R solutions can address these inefficiencies. For example, dynamic reconfiguration, informed by automated network analytics, enables operators to adapt SFCs to changing conditions dynamically. However, the existing solutions face significant limitations, especially in complex, distributed environments like multi-cloud or hybrid-cloud deployments, where network orchestration and coordination of numerous VNFs are increasingly challenging [12], [13].

Optimization techniques are essential to improve SFC-R efficiency, focusing on goals such as reducing end-to-end (E2E) latency, minimizing VNF migration costs, and enhancing scalability [14]. By leveraging mathematical models, algorithms, and heuristics, network operators can dynamically place VNFs closer to end-users, improving response times for latency-sensitive applications such as real-time communications and financial transactions [15], [16]. Moreover, optimization in SFC-R plays a critical role in ensuring SFC's scalability [17], [18]. As the user base and application demands grow, optimization techniques allow network operators to scale SFCs efficiently, reallocating resources such as computing power and bandwidth to ensure continuous service delivery without over-provisioning. Despite advancements, SFC-R optimization remains an open research problem. Existing works focus on specific challenges such as deployment and scaling but often overlook comprehensive approaches that holistically address performance, reliability, and resource utilization [19], [20], [21].

This research focuses on enhancing the optimization and reconfiguration of SFCs, which is critical in next-generation networks. The main contributions of this paper can be summarized as follows.

- We proposed a mixed integer linear programming (MILP) problem with predefined paths that block the problem's computational complexity. We conduct extensive simulations in a 6-node network and a 14-node network. The results validate that our model can find values of the objective functions for different scale networks while ensuring a low running time.
- With the benefits of the model, we design a two-stage SFC-R, optimization and re-optimization, which considers E2E latency and VNF migration minimization. Furthermore, the approach seeks to remedy the lack of strategies to calculate the latency limit between two sequential functions. Previous latency studies have not addressed this problem or treated it with non-linear constraints. We introduce a set of constraints that represent the limit of flow latencies.
- Our proposed approach guarantees to address the SFC's reliability concerns.
- The proposed SFC-R approach provides a load-balanced physical server across the substrate network hosts the several running SFCs.

The rest of this paper is organized as follows. The main related works are discussed in Section II. Section III describes the motivation and Section IV describes the methodology used in the paper. Section V defines the network and reconfiguration models. The SFC-R problem is formulated in Section VI. We use extensive numerical simulations for performance evaluation in Section VII. Finally, we summarize our paper and some open issues in the future work are discussed in Section VIII.

II. Background

Two critical aspects of SFC optimization are SFC migration and SFC resource allocation. Works related to our paper are categorized into these aspects, and without loss of generality, we can say that they are related to SFC-R. The research presented in [19] explores heuristic algorithms for SFC migration, focusing on optimizing the average latency of deployed SFCs in edge-core networks while meeting specified Service Level Agreement (SLA) requirements. The proposed solution aims to schedule SFC migrations in edge-core networks to address the latency-aware migration problem. In addition, Pham [20] introduced an Integer Linear Programming (ILP) model for SFC migration, which utilizes both heuristic and reinforcement learning algorithms. The proposed model aims to enhance the cost-efficiency of network function virtualization by addressing fluctuations in service demands and dynamic routing. The authors in [22] identified critical challenges in SFC, including the maintenance of state consistency and the management of dependencies between VNFs. This research underscores the importance of SFC orchestration mechanisms in managing variations in traffic distribution over time and responding to real-time changes. In several scenarios, such as unmanned aerial vehicles (UAV) or vehicular Internet (IoV), the network topology varies due to the mobile edge servers. To address changes in the routing path between adjacent VNFs in SFCs, the authors in [21] investigate the SFC migration problem with long-term budget constraints. They propose an SFC migration method that balances latency and cost of migration. However, it does not consider the reliability and scalability requirements in SFC.

Resource management for SFCs focuses on efficiently distributing network and computational resources to meet the requirements of VNFs [23]. This involves optimizing resource utilization, reducing latency, and ensuring scalability. The authors in [24] explored resource management solutions that optimize the placement of virtual network functions (VNFs)

within SFCs. Their work presents an integer linear programming (ILP) model that minimizes overall network operational expenditure and physical resource fragmentation. A jointly optimized resource allocation in NFV is investigated in [25]. The authors formulated the resource allocation problem as mixed integer linear programming. However, their proposed heuristic solution does not consider the service chain path planning with respect to incoming demands.

On the other hand, ensuring QoS and adhering to SLAs are paramount in SFC resource management. The research in [26] addresses the challenge of multi-traffic scheduling in VNF-based service orchestration. It introduces a dynamic approach to ensure Quality of Service (QoS) across multiple services. The proposed method aims to minimize data coupling among services and prevent communication resource type preemption.

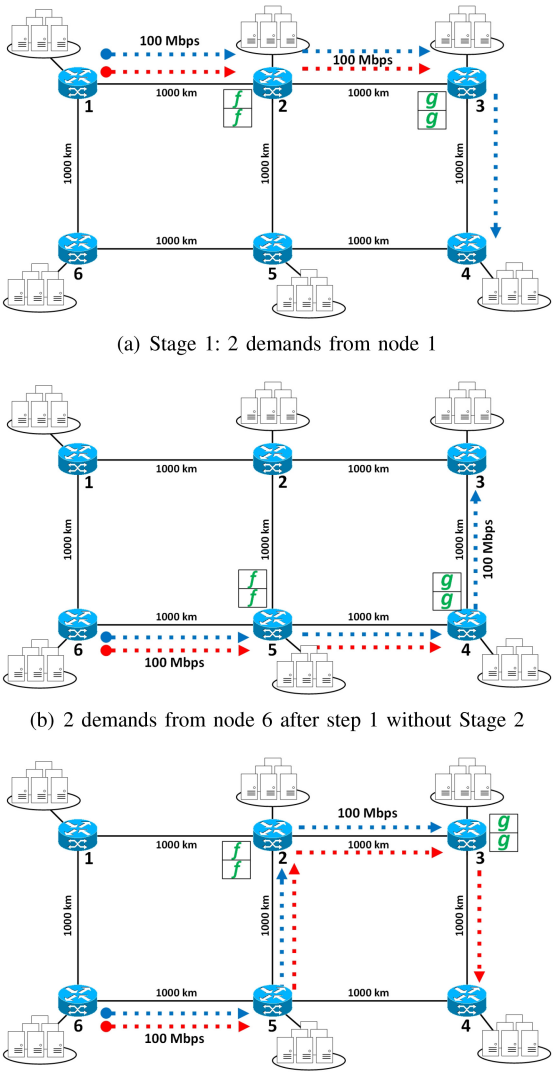
While significant progress has been made in SFC optimization, several areas require further research. Several already deployed SFCs may be running on the substrate network, and new incoming SFC demands with heterogeneous SLA requirements have been submitted. The already deployed SFC may not be able to serve these incoming demands through a conventional SFC optimization approach. The SFC optimization approach in the network orchestrator is expected to meet the SLA requirements of new SFCs in the light of guaranteeing the SFC's reliability and providing servers' load balancing across the substrate network.

III. Motivation

There are several studies about the reconfiguration of SFC and the migration of VNF in NFV-enabled networks. However, there are still some problems remaining to be solved. Firstly, the VNF sharing phenomenon, across the SFCs, is common in NFV-enabled networks. As it is illustrated in Fig. 1, let $G = (N, E)$ be a substrate network, where $N = \{n_1, n_2, \ldots, n_{|N|}\}$ and $E = \{l_1, l_2, \ldots, l_{|L|}\}$ denote, respectively, the set of existing physical nodes and physical links between the nodes.

We assume several service function chains (SFCs) are already deployed across the substrate network, serving as the dynamic network services. For example, in Fig. 1(a), let the service function chains SFC_1 (red dashed lines) and SFC_2 (blue dashed lines) already be deployed on the substrate network. SFC_1 is running across the physical nodes n_1, n_2 and n_3, while SFC_2 is serviced across the physical nodes n_1, n_2, n_3 and n_4. We assume VNF instances type f and g are deployed at nodes n_2 and n_3, respectively, for the service function chains SFC_1 and SFC_2. While most previous studies assumed that a VNF is only used by one SFC, without considering the sharing of VNF, Fig. 1(a) shows a real-world situation.

Secondly, dynamic network services can lead to multiple links' or nodes' capacity overload simultaneously. Therefore, the related VNF migration is often performed concurrently. For example, in Figure 1(b), there is reconfiguration in the network and the services demand d associated to SFC_1 and SFC_2 are being served across the physical nodes n_6, n_5, and n_6, n_5, n_4, n_3 respectively. Therefore, the VNF instance types f and g must be migrated from n_2 and n_3 to physical nodes n_5 and n_4, respectively. However, a few of the literatures have



(a) Stage 1: 2 demands from node 1

(b) 2 demands from node 6 after step 1 without Stage 2

(c) 2 demands from node 6 after Stage 1 with Stage 2 - Reconfiguration Approach

Fig. 1. Different demand and VNF assignment for a 6-node network.

focused on this problem. Finally, although the conventional virtual machines (VMs) migration technology has matured in data centers, VNF cooperates to provide services for the SFC reconfiguration problem. The conventional VM migration algorithms fail to consider multiple service flows' states and perform poorly.

In this paper, we study the VNF migration in light of the SFC reconfiguration problem considering VNF sharing and concurrent VNF migration. VNF migration tends to be more lightweight and faster than conventional VM migration [15]. Therefore, we mainly focus on the influence of VNFs migration on network services and their status. The idea is to minimize the possible number of VNFs migration to fulfill an efficient SFC reconfiguration. For example, as shown in Fig. 1(c), the reconfiguration associated to SFC_1 and SFC_2 is decided without the migration of VNF instances.

Alternatively, the idea is to minimize the number of scaling up/down the allocated computation capacities to the deployed VNF instances on the related physical server, through changing

the computation capacity (i.e., number of cores) in each node. Through this idea, the operator does not need to go to a node to add or to remove cores manually. If there needs to be any update in terms of the VNF types associated with running SFC, it will be done dynamically.

Without loss of generality, the physical topology is assumed to be generic, i.e., the nodes can be connected in arbitrary topologies using even multiple fibers between them. Note that the analysis presented in the following sections applies only to a network that is used to carry packet traffic service from a source to a destination, perhaps over several virtual links.

IV. Methodology

Multi-period or Multi-stage design refers to network design problems spanning a time horizon. In the multi-stage case, the demand volume for each period is new, meaning that this volume is in addition to the demand volume that was present in prior periods; this newness of demand in a subsequent period is sometimes better understood as incremental demand volume.

When we consider multiple stages, there are a few important issues to consider (i.e., besides the incremental demand volume): 1) the cost structure may change over time (e.g., due to economic discounting), 2) demand that is routed during a one-time window may incur maintenance costs in a later time window, 3) the capacity expansion can happen over the entire time horizon, 4) VNFs may need to be migrated from one node to another (e.g., from one server to another, or from one data center to another) as network conditions evolve. The need for VNF migration arises for various reasons, such as better balancing the load across the network and avoiding resource exhaustion. However, the migration process consumes resource types (i.e., bandwidth, CPU, and memory), which may interfere with normal operations. If large VNFs or numerous VNFs are migrated simultaneously, this can lead to congestion and degraded performance.

In this paper, each traffic demand matrix is associated with a stage, with the first stage consisting of an optimization process aimed at minimizing the total network latency. However, for each of the subsequent *D* traffic demand matrices, a re-optimization is performed, and the strategy solves the problem based on the SFC placement while determining the number of VNF outputs. The objective in subsequent stages is to minimize VNF migration between *D* in *t* and *D* in $t + \Delta$, where *t* represents the stage under consideration. The output of each re-optimization stage contains the established VNFs, and the next stage takes the already established VNFs from the previous one as input and tries to minimize the migration of already used VNFs. The same procedure is followed for every stage throughout the time horizon under consideration. This paper considers only two stages to explain the idea better. Therefore, in detail:

1) Stage 1: Initial Network Planning With Minimum Latency: Given an initial set of traffic demands, the objective is to optimize the allocation of SFC functions across network nodes to achieve the lowest possible latency. This stage involves defining the network structure, the required service functions, and the traffic flows that must traverse specific sequences of functions. The outcome of this stage is an optimized network plan that specifies the allocation of functions and the routing of traffic flows, ensuring minimal latency across all demands.

TABLE I
List of Input Notations

Notation	Description
D	Set of demands, $D = \{d_1, d_2, \ldots, d_{\lvert D \rvert}\}$
N	Set of existing nodes in the network
E	Set of existing physical links (u, v) in the network
P_d	Candidate paths for demand d: $P_d = \{p_1, p_2, \ldots, p_{\lvert P_d \rvert}\}$
$q_{uv}^{d,p}$	Indicates whether link (u, v) is hosting path p_d of demand d
$\xi_u^{d,p}$	Indicates if node u is hosting path p_d of demand d
h^d	Traffic intensity of demand $d \in$D
s_d, d_d	Source and destination nodes of demand d
c_u	Computational capacity of node u. It indicates the number of cores that node u supports
m_f	Maximum rate for function f
$l_{u,v}$	Latency between nodes u and v of a physical link $\in$E
t_{max}^d	Maximum tolerated latency of demand d from a source s_d until the destination d_d
$L_{max}^{d,f,g}$	Maximum tolerated latency between VNFs f and g for demand d. It means the maximum tolerated latency between two sequential functions deployed on distinct nodes in the network
F_d	Virtualized functions requested by d: $F_d \subset F$. This set represents the quantity and types of functions requested by a demand d
$\delta_{f,g}^d$	Binary indicator of an anti-affinity rule between f and g. It specifies that these two functions must not be deployed on the same physical node
$\hat{D}$	Future demands to be allocated: $\hat{D} = \{d_1, d_2, d_{\lvert \hat{D} \rvert}\}$ (Stage 2)
$x_{f,u}^{old}$	Established instantiations of function f on node u from Stage 1 (Stage 2)
$\delta_{f,u}^{old}$	Binary indicator associated with $x_{f,u}^{old}$ (Stage 2)
M	A large constant (big-M notation). It is a sufficiently large positive number that activates or deactivates constraints based on binary decision variables of the formulation

2) Stage 2: Adaptation to Traffic Change With Minimal Reconfiguration: Over time, new traffic demands may emerge. However, the total latency constraint established in Stage 1 can be kept as a requirement. Therefore, Stage 2 involves a re-optimization process that seeks to accommodate new demands while minimizing disruption to the initial configuration. Two alternative objectives for this stage can be considered: to minimize the migration of functions to other nodes and to minimize the change in the number of function cores over the network nodes. We will explain in detail later, the possible objectives for this re-optimization stage.

The approach with re-optimization ensures that latency constraints remain within predefined limits, while also addressing the complexities of VNF migration. This method is particularly relevant in NFV-enabled networks, where flexible service deployment and dynamic adaptation are essential. Using MILP models, the strategy effectively balances performance optimization with reconfiguration costs, ensuring network stability even in highly dynamic environments.

V. Model

In this section, we formulate the network model to clarify the optimization and re-optimization approaches for deploying SFC-R over backbone networks. Tables I and II list the notations and their definitions used in the network and proposed model.

TABLE II
LIST OF DECISION VARIABLES NOTATION

Notation	Description
$B^{d,p}$	It represents the traffic flow allocated to path p_d for demand d in the network. It quantifies the amount of traffic assigned to a specific path among the set of candidate paths available for a given demand
$b^{d,p}$	It is a binary decision variable that indicates whether traffic is allocated to path p_d for demand d. It takes a value of 1 if any traffic is assigned to $B^{d,p}$, and 0 otherwise, ensuring that the activation of a path is properly represented in the optimization model
t^d	It indicates the end-to-end latency of demand d from source s_d until the destination d_d
$t^{d,p,f}$	It indicates the latency of demand d (from the source) by the path p until the function f
$x_{f,u}$	It represents the number of instances of Virtual Network Function (VNF) f that need to be deployed on physical node u to meet the service requirements of the network
$\delta_{f,u}$	It is a binary indicator that determines whether at least one instance of VNF f is deployed on physical node u
$y_u^{d,p,f}$	It is a binary variable that takes the value 1 if the function f handles demand d on node u of path p
$Y_u^{d,p,f}$	It is a binary variable that takes the value 1 if It is a binary variable that takes the value 1 if d meets its assigned f before or on node u of path p
w_u	It indicates the status of node u regarding function activation. A value of 1 signifies that node u is actively performing its designated function. Conversely, a value of 0 means that node u is inactive or does not have an assigned function
Γ_u	It is a binary variable that indicates whether a physical node u has updated its number of cores. If Γ_u = 1, it means that node u has performed an update to its core count. If Γ_u = 0, it means that no update has been made to the number of cores at node u
B_{uv}	It is the traffic load over a physical link (u, v) in a network. It refers to the total amount of traffic passing through the link connecting node u and v
C	It indicates the maximum traffic capacity of any link refers to the highest amount of traffic that a single physical link in the network can handle before experiencing congestion. This issue is relevant to network planning. For now, we will designate C as a variable, allowing it to be used as a parameter once the network has been fully planned
C_T	It represents the sum of the traffic load B_{uv} <= C utilized by all connections in the network, ensuring that the system can handle the required traffic efficiently

Among the main definitions, we denote by P_d the set of all elementary (s_d, d_d) paths p_d associated with demand d. For each demand d, the number of feasible provided paths must not be too large; otherwise, it may incur a considerable processing burden, generally not providing corresponding performance improvements. The parameter $q_{uv}^{d,p}$ is the availability of the physical link between nodes u and v to host path p_d of demand d; it means if the path is or is not passing through link uv; $\xi_u^{d,p}$ is the existence of physical node u across the path p_d with the demand d.

The required number of instantiations of VNF f on node u is represented by the variable $x_{f,u}$ in Stage I, and this value will be used as input $(x_{f,u}^{old})$ in Stage II. Evidently, the re-optimization will also generate a new $x_{f,u}$ output for the Stage II.

A key advantage of path formulation (i.e., MILP with predefined paths) is its ability to streamline the solution space by preselecting a limited number of feasible paths, thereby reducing the complexity of the optimization problem, [27]. An excessive number of paths may introduce computational burdens without necessarily improving the quality of solution. By defining the parameter $q_{uv}^{d,p}$, which indicates whether a specific path traverses a given physical link (u, v), and the parameter $\xi_u^{d,p}$, which identifies whether a node u is present in a particular path, the model can efficiently enforce flow conservation and capacity constraints (see B_{uv}, C and C_T in Table II) while maintaining a tractable formulation.

VI. PROBLEM FORMULATION

In this section, we develop a MILP to address the NFV reconfiguration problem. The MILP consists of two stages: optimization and re-optimization. For clarity, we can view these as two separate MILP models, where the outputs from the first model serve as inputs to the second. Using the notations outlined in Tables I and II, we present our formulation strategy, which focuses on minimizing total latency in Stage I and reducing the migration of VNFs in Stage II, as follows.

A. *Stage 1: Current Demand Optimization*

The Stage 1 aims at reducing the sum of the end-to-end delay of all demands from D. The output will give the number of VNFs per node $(x_{f,u})$ in order to attend the objective. Therefore, the objective function and the set of constraints of the Stage 1 optimization problem are defined as follows.

- *Objective function:*

$$Minimize: \sum_{d \in D} t^d \tag{1}$$

- *Traffic routing constraints:*

$$\sum_{p \in P_d} B^{d,p} = h^d \quad \forall d \in D \tag{2}$$

$$B^{d,p}/M \leq b^{d,p} \quad \forall d \in D, p \in P_d \tag{3}$$

$$\sum_{p \in P_d} b^{d,p} \geq \frac{h^d}{M} \quad \forall\, d \in D \tag{4}$$

$$\sum_{p \in P_d} b^{d,p} \leq M\, h^d \quad \forall\, d \in D \tag{5}$$

$$\sum_{p \in P_d} b^{d,p} \leq 1 \quad \forall\, d \in D \tag{6}$$

Eqs. 2 assigns the total amount of the requested traffic to the provided routes, whereas Eqs. 4-6 create the binary indicator of which of the provided routes is used.

$$y_u^{d,p,f} \leq b^{d,p} \quad \forall\ d \in D, p \in P_d, f \in F_d, u \in N \tag{7}$$

Eq. (7) guarantees that function assignment may only occur in some nodes of the selected route.

- *Latency constraints:*

$$t^{d,p,f} = \sum_{u,v \in N} q_{u,v}^{d,p} \left(1 - Y_u^{d,p,f}\right) l_{u,v} \quad \forall\, d \in D, p \in P_d, f, g \in F_d \tag{8}$$

$$|t^{d,p,f} - t^{d,p,g}| \leq L_{max}^{d,f,g} \quad \forall\, d \in D, p \in P_d, f, g \in F_d \tag{9}$$

$$t^d = \sum_{u,v \in N} q_{u,v}^{d,p}\, b^{d,p}\, l_{u,v} \leq t_{max}^d \quad \forall\, d \in D \tag{10}$$

Eq. (8) calculates the time delay from the origin node until the node that executes the function f for each demand d and its selected path, $p \in P_d$; Eq. (9) states that the delay between functions f and g of a given demand cannot exceed a maximum established value between them. Eq. (10) computes the end-to-end SFC delay of demand d once path p is assigned to it. Note that the time (t^d_{max}) to execute the SFC of demand d is already guaranteed in the supply of the incoming routes.

- *Bandwidth constraints:*

$$\sum_{d\in D, p\in P_d} q^{d,p}_{u,v} B^{d,p} = B_{u,v} \qquad \forall\, u, v \in N \quad (11)$$

$$B_{u,v} \le C \qquad \forall\, u, v \in N \quad (12)$$

$$\sum_{u,v\in N} B_{u,v} = C_T \quad (13)$$

The left part of Eq. (11) adds the traffic of those demands with selected path passing through link *u*-*v* in order to account for required capacity in each link of the network. In Eq. (12), it is asserted that the compounded routed traffic through a link is limited to a pre-defined capacity, whereas in Eq. (13), the total traffic in the network is accounted for.

- *Function capacity constraints:*

Each core instantiated with a function f has a limited capacity and needs to be duplicated enough to handle all the commodities assigned to it on the corresponding node.

$$\sum_{d\in D}\sum_{p\in P_d} y^{d,p,f}_u B^{d,p} \le [t]\ m_f x_{f,u} \quad (14)$$

$$\sum_{d\in D}\sum_{p\in P_d} y^{d,p,f}_u B^{d,p} - \frac{1}{M} \ge [t] m_f \left(x_{f,u} - 1\right) \quad (15)$$

$$\forall\, u \in N, f \in F^d$$

This is a non-linear equation, which can be linearized assuming $Z^{d,p,f}_u = y^{d,p,f}_u B^{d,p}$ and using the fact that $B^{d,p}$ is bound limited, which can be used h^d as a possible limit. With such proceeding, Eq. (15) and Eq. (15) may be replaced by

$$\sum_{d\in D}\sum_{p\in P_d} Z^{d,p,f}_u \le m_f x_{f,u} \quad \forall\, u \in N\ \forall f \in F^d \quad (16)$$

$$\sum_{d\in D}\sum_{p\in P_d} Z^{d,p,f}_u - \frac{1}{M} \ge m_f \left(x_{f,u} - 1\right), \quad \forall\, u \in N, f \in F^d \quad (17)$$

and utilizing the following constraints.

$$Z^{d,p,f}_u \ge 0 \quad (18)$$

$$Z^{d,p,f}_u \le h^d\, y^{d,p,f}_u \quad (19)$$

$$Z^{d,p,f}_u \le B^{d,p} \quad (20)$$

$$Z^{d,p,f}_u \ge B^{d,p} - h^d \left(1 - y^{d,p,f}_u\right) \quad (21)$$

A node cannot instantiate more cores with whatever function f than its number of available cores; therefore, all instantiated functions in a node must be limited to the node core capacity:

$$\sum_{f\in F} x_{f,u} \le c_u w_u, \qquad \forall\, u \in N \quad (22)$$

- *Anti-affinity constraints:*

$$y^{d,p,f}_u + y^{d,p,g}_u \quad [t] \le 1 + \delta^d_{f,g} \quad (23)$$

$$\forall\, d \in D, p \in P_d, f \ne g \in F_d,\ u \in N$$

The rule of anti-affinity when present for functions in a demand d implies that they cannot be assigned at the same location. Whenever f and g hold anti-affinity rule in a demand d, $\delta^d_{f,g} = 0$ and Eq. (23) guarantees that either f or g may be assigned to node u in these cases.

- **Function ordering constraints:**

$$y^{d,p,f}_u \le \xi^{d,p}_u \quad (24)$$

$$\sum_u y^{d,p,f}_u \xi^{d,p}_u \le 1 \quad (25)$$

$$\sum_u y^{d,p,f}_u \xi^{d,p}_u \ge \frac{B^{d,p}}{M} \quad (26)$$

$$q^{d,p}_{uv}\left(Y^{d,p,f}_v - Y^{d,p,f}_u - y^{d,p,f}_v\right) = 0 \quad (27)$$

$$Y^{d,p,f}_{s_d} - y^{d,p,f}_{s_d} = 0 \quad (28)$$

$$\forall\, d \in D, p \in P_d,\ f \in F_d, u, v \in N$$

Eqs. 25-27 assert that a function f in demand d can be assigned only in some of the nodes of the selected path p and at least one of path p's nodes must host it. Eqs. 27 assert that, for all pairs of consecutive nodes u, v in the selected path p for demand d, $Y^{d,p,f}_v = Y^{d,p,f}_u + 1$ whenever a function is installed at node v (i.e., $y^{d,p,f}_v = 1$), which represents the defined transition in $Y^{d,p,f}_v$ regarding $Y^{d,p,f}_u$ at function f installation node; and $Y^{d,p,f}_v = Y^{d,p,f}_u$ in any other node. Eq.-28 deals with the possibility of function installation at the origin node.

In case function f of demand d shall not be processed at the origin node, it is enough to include it in the formulation:

$$y^{d,p,f}_{s_d} = 0 \quad \forall d \in D, p \in P_d, f \in F \quad (29)$$

B. *Stage 2: Reconfiguration for Future Demand $\hat{D}$, (Re-Optimization)*

In this stage of the network planning process, it is assumed modifications to network traffic demands to some specific configurations regarding the originally planned ones so that all constraints imposed on the new demanded traffic shall be satisfied. The MILP formulation can infer possible costs associated with a network redesign strategy due to traffic modification through the distinct objective functions. For instance, this could be used to estimate the cost of running network planning over a defined traffic demand. However, after deploying the network, the actual traffic demand may not be realized as expected or could change over time, requiring further adjustments to the designed network.

The proposed strategy in this subsection treats this problem from two cost perspectives, described below. Both perspectives are related to operational expenditures (OPEX) related to scaling up/down the allocated computational capacity (i.e., the number of cores). It is worth noting that other perspectives may be assumed. However, without the loss of generality, we understand that the two perspectives proposed in the

paper are reasonable enough and are the evidence in light of the reconfiguration assessment strategy proposed. Meanwhile, additional policies are out of the scope of this paper and may be evaluated in future studies.

Firstly, we define an indicator binary variable $\delta_{f,u}$ to distinguish the situations where no core is assigned to a VNF type f at node u from those where there are cores assigned to VNF type f at node u.

By introducing the following constraint, we can force $\delta_{f,u}$ to take the value 1 when $x_{f,u} > 0$:

$$x_{f,u} \geq 1 \rightarrow \delta_{f,u} = 1.$$

Therefore:

$$x_{f,u} - M\delta_{f,u} \leq 0, \tag{30}$$

We also wish to impose the condition:

$$x_{f,u} = 0 \rightarrow \delta_{f,u} = 0.$$

For satisfying that, it is enough to write:

$$\delta_{f,u} \leq M\, x_{f,u} \tag{31}$$

$$\delta_{f,u} \geq \frac{x_{f,u}}{M}, \quad \forall f \in F, u \in N \tag{32}$$

Together, Eqs 30-32 impose the conditions:

$$\delta_{f,u} = 1 \leftrightarrow x_{f,u} > 0$$
$$\delta_{f,u} = 0 \leftrightarrow x_{f,u} = 0.$$

1) Alternative 1: In this Case, the Reconfiguration Cost is Expressed by the Number of Network Functions That are Instantiated in or Depleted From a Node: Therefore, whenever a type of function is absent in a node and becomes necessary to meet traffic requirements, or when it is present but the optimal solution requires its complete removal, a unit cost is added to network planning. Below, we describe how the proposed minimized cost can be achieved.

Objective Function One for the re-optimization:

$$Minimize: \sum_{f \in F, u \in N} |\delta_{f,u} - \delta_{f,u}^{old}| \equiv cost_1 \tag{33}$$

Notice that any complete removal of a function or its introduction in a node is properly computed by the provided objective function. Such an objective function is a non-linear function, and it can be linearized by knowing that $\delta_{f,u}$'s are binary variables. Therefore, the term $|\delta_{f,u} - \delta_{f,u}^{old}|$ is equal to $\delta_{f,u}$, if $\delta_{f,u}^{old} = 0$. And $|\delta_{f,u} - \delta_{f,u}^{old}|$ is equal to (1 - $\delta_{f,u}$) if $\delta_{f,u}^{old} = 1$.

2) Alternative 2: In This Case, the Reconfiguration Cost is Quantified by the Number of Nodes Whose Amount of Cores Need to be Altered: The reason for such an objective function is the fact that, whenever the number of cores must be altered in a node, either for an additional or inferior value, an intervention of a network operator is required to relocate cores among nodes.

In order to achieve an objective function that can reproduce such purpose, let us define a binary variable, Γ_u, so that it is set to one whenever there is an alteration in the number of assigned cores to a node u and zero otherwise. This can be achieved by applying the two inequalities below:

$$\Gamma_u \geq \frac{|\sum_{f \in F} x_{f,u} - \sum_{f \in F} x_{f,u}old|}{M}$$
$$\Gamma_u \leq M\,|\sum_{f \in F} x_{f,u} - \sum_{f \in F} x_{f,u}old|$$

Notice again that these constraints are non-linear. However, they can be linearized by tricks from [28].

Finally, the intended objective can be achieved by:

Objective Function Two for the re-optimization:

$$Minimize: \sum_{u \in N} \Gamma_u \equiv cost_2 \tag{34}$$

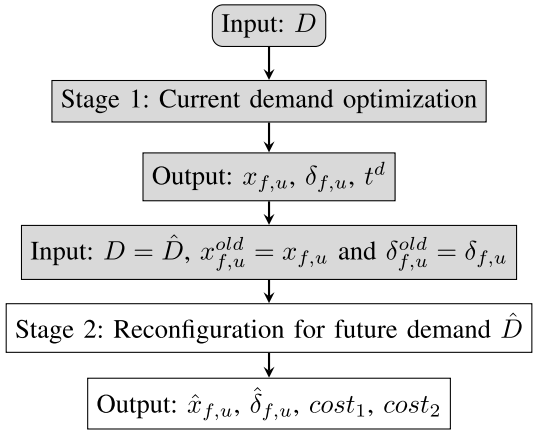


Fig. 2. Proposed two-stage reconfiguration assessment strategy.

C. Summary of the Two Stages

The action of VNF migration quantifies adjustments on predefined sites of VNF instances occurring in the network. We treat migration requirements and the strategy to efficiently deal with it as a subsequent MILP formulation that receives the output of the previously configured network, after focusing on optimizing a specific objective function (i.e., in our example the total end-to-end delay) and focuses on reducing migration costs, but constrained to all new conditions. This migration assessment approach is referred to as **Stage 2**. Figure 2 shows the general structure of the two-stage proposed network reconfiguration assessment strategy. Initially, it is provided an expected traffic matrix to the **Stage 1** (Eqs. 1-29) and the solution with the minimum end-to-end aggregate delay is obtained. The **Stage 2** receives the new network traffic condition, the output of the previous stage and it is emphasized the number of functions affected by migration. Mathematically, we have three new parameters: $\hat{D}$, which states the new matrix of traffic demand, $x_{f,u}^{old} = x_{f,u}$ and $\delta_{f,u}^{old} = \delta_{f,u}$, obtained from the result of **Stage 1**. The MILP formulation is run again (re-optimization), but with a new matrix demand $\hat{D}$ and the new objective function, related to the network migration assessment compared to the output configuration in **Stage 1**. The flowchart is presented in Fig. 2.

TABLE III
PARAMETERS OF SIMULATION SETUP FOR 6-NODE NETWORK

Symbol	Description	6-node	NSFNET
$\|D\|$	Number of demands	4	11
$\|\hat{D}\|$	Number of demands	4	11
h^d	Demands in Mbps	100-400	100-400
$\|P_d\|$	Number of predefined paths	2	4
$\|F\|$	Number of functions per demand	2	3
m_f	Processing capacity per function in Mbps	10	10
c_u	Max processing capacity per node in Mbps	5000	5000
M	Big number	10000	10000

TABLE IV
ORIGINAL TRAFFIC DEMANDS (D) FOR EACH SOURCE-DESTINATION (s-d) NODE PAIR ON THE 6-NODE NETWORK USED IN SIMULATIONS IN STAGE 1

Original Traffic D

	1	2	3	4	5	6
1	.	0	0	0	**100**	0
2	0	.	0	0	0	0
3	0	0	.	0	0	**100**
4	**100**	0	0	.	0	0
5	0	0	0	0	.	0
6	0	0	**100**	0	0	.

VII. EVALUATION

The proposed MILP formulations and recovery assessment strategies in the paper were solved using the IBM ILOG CPLEX solver [29] hosted on an Intel i7 3.6GHz machine with 32GB of RAM. We have analyzed two very distinct topologies: a) a small network, where the flow-based MILP formulation is capable of providing results in a feasible time, so that they can be compared to those provided by the path-based MILP formulation proposed in this paper; and b) a more realistic topology, frequently used in several works in the literature as a benchmark network. In the next subsections, we present these two topologies and discuss the obtained results.

A. Small Network

To test the effectiveness of the MILP formulation and the two-step reconfiguration assessment strategies, a first case study considers the 6-node physical network substrate illustrated in Fig. 1. The values of the parameters used in the simulations are listed in Table III. For this study, we consider four alternative future traffic demands, $\hat{D}$. In each provided traffic matrix, the value in line *i*, column *j* informs the demanded traffic intensity adopted between the source-destination node pair $i-j$.

Table IV shows the traffic intensity assumed for each source-destination node pair, which is used as the original traffic in Stage 1 of the MILP formulation. Table V shows the four future traffic scenarios used for analyzing the provided solutions of the two proposed reconfiguration assessment strategies. Scenarios I, II, III, and IV refer to each of the analyses performed hereafter in this paper with respect to the four adopted future traffic scenarios, $\hat{D}$.

Tables VI and VII show the number of established VNFs per node according to the solution to the MILP formulation in Stage 1 for the original traffic matrix (Table IV) and objective function (Eq. (1)) as well as in Stage 2 when the original traffic demand is replaced by one of the four provided new traffic demands and the original objective function is replaced by either Objective Function One or Two. Data in Table VI are related to the solutions obtained from solving Stage 2 with Objective Function One. Similarly, data in Table VII refer to the results obtained from running the reconfiguration method proposed in Stage 2, but with Objective Function Two. The left shadowed column refers to the results of the MILP formulation after Stage 1, whereas the subsequent four columns refer to the results after Stage 2 under the different traffic Scenarios I, II, III and IV, in this order.

From Table VI, it can be observed that, under Scenario I, which assumed the same total aggregated traffic as the original one, differentiated just by their source-destination node pairs, just one VNF site reconfiguration would be necessary to transition from the solution with the old traffic matrix to the solution with the new traffic matrix under Objective

TABLE V
FUTURE TRAFFIC DEMANDS ($\hat{D}$) FOR EACH SOURCE-DESTINATION (s-d) NODE PAIR ON THE 6-NODE NETWORK USED IN SIMULATIONS. EACH SCENARIO REPRESENTS A DIFFERENT $\hat{D}$ USED IN STAGE 2

(a) Scenario I: $\hat{D}$

	1	2	3	4	5	6
1	.	0	**100**	**100**	**100**	0
2	0	.	0	0	0	0
3	0	0	.	0	0	0
4	0	0	0	.	0	0
5	0	0	0	0	.	0
6	0	0	**100**	0	0	.

(b) Scenario II: $\hat{D}$

	1	2	3	4	5	6
1	.	0	**300**	**200**	0	0
2	0	.	0	0	**300**	**400**
3	0	0	.	0	0	0
4	0	0	0	.	0	0
5	0	0	0	0	.	0
6	0	0	0	0	0	.

(c) Scenario III: $\hat{D}$

	1	2	3	4	5	6
1	.	0	**200**	0	0	0
2	0	.	0	0	0	0
3	0	0	.	0	0	0
4	0	**100**	0	.	0	0
5	0	0	0	**300**	.	0
6	0	0	0	**200**	0	.

(d) Scenario IV: $\hat{D}$

	1	2	3	4	5	6
1	.	0	0	0	0	0
2	0	.	0	0	0	0
3	0	0	.	0	0	0
4	0	**200**	0	.	0	0
5	0	**400**	0	0	.	0
6	0	**400**	0	**200**	0	.

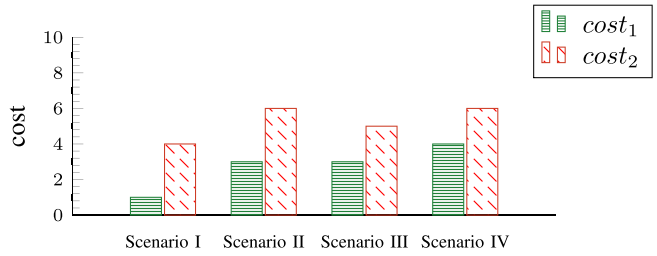
TABLE VI
NUMBER OF VNFS INSTANCES f PER NODE FOR THE ORIGINAL TRAFFIC DEMANDS (STAGE 1) AND NEW TRAFFIC DEMANDS (SCENARIOS OF STAGE 2) FOR OBJECTIVE FUNCTION ONE

Number of VNF instances f per node $x_{\hat{f},u}$

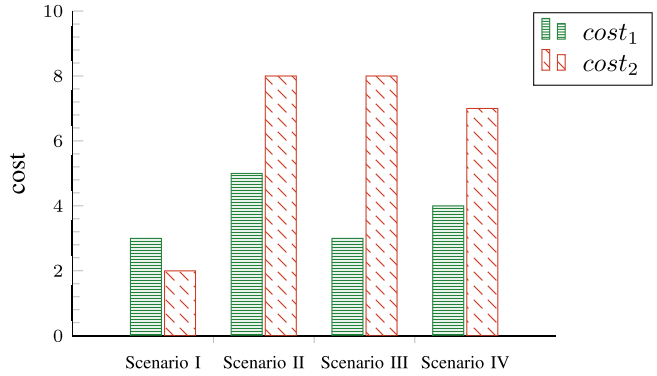
	$x^{old}_{f,u}$		Scen I		Scen II		Scen III		Scen IV	
node	f=1	f=2	f=1	f=2	f=1	f=2	f=1	f=2	f=1	f=2
1	0	10	0	0	30	0	0	0	0	40
2	10	0	20	0	20	0	50	30	20	60
3	0	10	0	30	0	50	0	30	0	20
4	10	0	10	0	30	0	0	0	0	0
5	20	20	10	10	40	30	30	20	60	0
6	0	0	0	0	0	40	0	0	40	0

TABLE VII
NUMBER OF VNF INSTANCES f PER NODE FOR THE ORIGINAL TRAFFIC DEMANDS (STEP 1) AND NEW TRAFFIC DEMANDS (SCENARIOS OF STAGE 2) FOR OBJECTIVE FUNCTION TWO

	$x^{old}_{f,u}$		Scen I		Scen II		Scen III		Scen IV	
node	f=1	f=2	f=1	f=2	f=1	f=2	f=1	f=2	f=1	f=2
1	0	10	0	0	30	0	0	0	80	0
2	10	0	20	0	30	0	0	10	0	120
3	0	10	0	10	0	0	40	0	0	0
4	10	0	0	10	0	50	0	70	0	0
5	20	20	20	20	40	0	40	0	40	0
6	0	0	0	0	20	70	0	0	0	0



(a) MILP (Objective Function One)



(b) MILP (Objective Function Two)

Fig. 3. Values of $cost_1$ and $cost_2$ for Stage 2: MILP simulation in the 6-node network under traffic Scenarios I, II, III and IV: (a) Simulation with Objective Function One (b) Simulation with Objective Function Two.

Function One. For Scenarios II, III and IV, on the other hand, the newly provided traffic increases three, two and three fold, respectively, which results in 3, 3 and 4 necessary (i.e., minimum) site reconfigurations. It is possible to observe there is a certain coherence between traffic increase and number of reconfiguration sites. However, the localization of the traffic change is also very important for the optimum site reconfiguration solution, as observed between Scenarios II and III, where the same solution was achieved with different traffic change intensities, and Scenarios II and IV, where the same total traffic intensity is assumed, but different site reconfiguration requirements were observed.

Table VII summarizes the results of the same four scenarios analysed before, but with Stage 2: MILP using Objective Function Two, which aims at reducing the variation in the number of cores in the network nodes. This is important to be applied when it is worth preventing an operator from moving to a node to add or remove cores whenever reconfiguration is required. We can see in Table VII that, in Scenarios I, II, III and IV, it was necessary to make adjustments in the number of cores at 2, 5, 3 and 4 nodes, respectively.

In order to highlight the difference in values of $cost_1$ and $cost_2$ when the objective function that properly treats them is used or not, we synthesize the values of $cost_1$ and $cost_2$ for each of the objective functions applied to Stage 2: MILP. Fig. 3(a) shows the values of $cost_1$ and $cost_2$ when Objective Function One is employed at Stage 2: MILP. Notice that, in this case, all $cost_1$ are optimum values, whereas $cost_2$ is the non-optimum resulting value extracted from the simulation. Fig. 3(b) shows the results of $cost_1$ and $cost_2$ but with the

TABLE VIII
COMPARISON BETWEEN MILP AND MILP$_{LUB}$ SOLUTIONS FOR THE 6-NODE NETWORK UNDER OBJECTIVE FUNCTION ONE

Objective 1 ($cost_1$)							
MILP				MILP$_{LUB}$			
Scen I	Scen II	Scen III	Scen IV	Scen I	Scen II	Scen III	Scen IV
1	3	3	4	2	4	4	4

TABLE IX
COMPARISON BETWEEN MILP AND MILP$_{LUB}$ SOLUTIONS FOR THE 6-NODE NETWORK UNDER OBJECTIVE FUNCTION TWO

Objective 2 ($cost_2$)							
$MILP$				$MILP_{LUB}$			
Scen I	Scen II	Scen III	Scen IV	Scen I	Scen II	Scen III	Scen IV
2	5	3	4	3	5	3	4

employment of Objective Function Two at Stage 2: MILP. Therefore, $cost_2$ in this case has optimum values and $cost_1$ values are non-optimum. Notice that $cost_1$ and $cost_2$ are limited to NF and N, respectively. Comparing $cost_1$ (green bars) in both graphs, it is observed that, in all analysed scenarios, additional function migrations are required in the solutions provided by Objective Function Two compared to those by Objective Function One, since the specified cost is that specifically treated by Objective Function One. Increases as high as 200%, 166% and 75% were faced. On the opposite direction, higher values to $cost_2$, which is associated with core migrations, adequately treated by Objective Function Two, are required when Objective Function One is used. For example, 100%, 20%, 66% and 50% more core migrations were required in the analysed scenarios. These results highlight the importance of choosing the proper Objective Function regarding the intended objective.

B. New Method: Latency as a Constraint

An important metric that can modify the reconfiguration solution is the maximum function latency. Limiting the value of this metric (by specifying an upper bound) can enable network operators to limit "VNF migration" $(cost_1)$ or "node core amendments" $(cost_2)$ in the network during the transition from traffic matrix D to $\hat{D}$. The following strategy includes Objective Function One and Objective Function Two as objective functions as well as latency of Stage 1 as a constraint, which represents an upper bound for Stage 2: MILP. Therefore, Stage 2 shall be modified by including the new constraint:

$$\sum_{\hat{d}} t^{\hat{d}} \leq \sum_{d} t^{d} = T_{max} \quad \forall\, \hat{d} \in \hat{D}, \quad d \in D, \qquad (35)$$

where T_{max} is the sum of the latency of all demands of D, i.e., the result of the original objective function in Eq. (1). Therefore, Stage 2 is a MILP with a latency upper bound, which will be referred to as MILP$_{LUB}$ in this paper.

Table VIII and Table IX present the solution of the two function reconfiguration strategies when the latency constraint is guaranteed by the MILP formulation. As it can be observed, the number of VNFs reconfiguration increased by one unit in all analyzed scenarios, except in Scenario IV, in which it was kept the same. On the other hand, MILP$_{LUB}$ could keep the

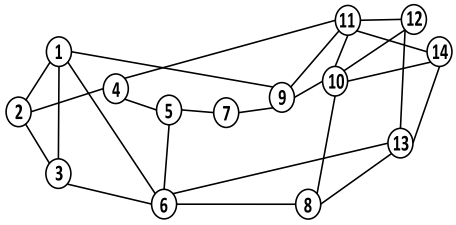


Fig. 4. NSFNET substrate network.

TABLE X
ORIGINAL TRAFFIC DEMANDS (D) FOR EACH SOURCE-DESTINATION (s-d) NODE PAIR ON THE 14-NODE NETWORK (NSFNET) USED IN SIMULATIONS IN STAGE 1

Original Traffic D

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	.	-	-	-	-	-	-	-	-	-	-	-	-	-
2	-	.	-	-	-	100	-	-	-	-	-	-	-	-
3	-	-	.	-	-	-	-	-	-	-	-	-	-	-
4	-	-	-	.	-	-	-	-	-	-	-	-	100	-
5	-	-	100	-	.	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	.	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	.	-	-	-	100	-	-	-
8	-	-	-	-	-	-	-	.	-	-	-	-	-	100
9	-	-	-	100	-	-	-	-	.	-	-	-	-	-
10	100	-	-	-	-	-	-	-	-	.	-	-	-	-
11	-	-	-	-	-	-	-	-	100	-	.	-	-	-
12	-	-	-	-	-	-	100	-	-	-	-	.	-	-
13	-	-	-	-	100	-	-	-	-	-	-	-	.	-
14	-	-	-	-	-	-	-	100	-	-	-	-	-	.

number of cores per node the same in all scenarios, except in Scenario I, where a unitary increment was necessary.

C. 14-Node Network

Even with the complexity of the problem for large networks, the complete strategy presented in the proposed MILP formulation does not demand excessive computational time. Indeed, the proposed MILP formulation with predefined paths makes the problem viable in terms of simulation time.

For investigating large networks, the NSFNET topology was used, where its physical substrate is presented in Fig. 4 and all assumed parameters and their values are listed in Table III. Table X shows the current demand, D, whereas the two assumed future traffic scenarios, $\hat{D}$, are presented in Table XI. Note that, in Scenario I, the future traffic demand has been maintained with the same spatial configuration, but with double-intensity traffic demands for each source-destination node pair. On the other hand, the number of source-destination node pairs in the future traffic demand of Scenario II is the same as in the original traffic demand, but these pairs and traffic intensity have been randomly selected.

Similarly to what was observed in the simulations for the 6-node network, the employment of the proper objective function for the aimed cost results in solutions with considerably lower cost. Table XII and Table XIII show the NFVI deployment solution of Stage 2: MILP formulation when Objective Function One and Two, respectively, are employed. Fig. 5(a) and Fig. 5(b) synthesizes the values of $cost_1$ and $cost_2$ for Stage 2: MILP under Objective Function One and Two, respectively. Notice therefore that just $cost_1$ is an optimized value in Fig. 5(a) and $cost_2$ in Fig. 5(b).

TABLE XI
FUTURE TRAFFIC DEMANDS ($\hat{D}$) FOR EACH SOURCE-DESTINATION (s-d) NODE PAIR ON THE 14-NODE NETWORK USED IN SIMULATIONS. EACH SCENARIO REPRESENTS A DIFFERENT $\hat{D}$ USED IN STAGE 2

(a) Scenario I: $\hat{D}$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	.	-	-	-	-	-	-	-	-	-	-	-	-	-
2	-	.	-	-	-	200	-	-	-	-	-	-	-	-
3	-	-	.	-	-	-	-	-	-	-	-	-	-	-
4	-	-	-	.	-	-	-	-	-	-	-	-	200	-
5	-	-	200	-	.	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	.	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	.	-	-	-	200	-	-	-
8	-	-	-	-	-	-	-	.	-	-	-	-	-	200
9	-	-	-	200	-	-	-	-	.	-	-	-	-	-
10	200	-	-	-	-	-	-	-	-	.	-	-	-	-
11	-	-	-	-	-	-	-	-	200	-	.	-	-	-
12	-	-	-	-	-	-	200	-	-	-	-	.	-	-
13	-	-	-	-	200	-	-	-	-	-	-	-	.	-
14	-	-	-	-	-	-	-	200	-	-	-	-	-	.

(b) Scenario II: $\hat{D}$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	.	-	-	-	-	-	-	100	-	-	-	-	-	-
2	-	.	-	-	-	-	-	-	-	200	-	-	-	-
3	-	-	.	-	-	-	-	-	-	-	-	-	-	-
4	-	-	-	.	-	-	300	-	-	-	-	-	-	-
5	-	-	-	-	.	-	-	-	-	-	-	-	-	-
6	400	-	-	-	-	.	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	.	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	.	-	-	-	100	-	-
9	-	-	100	-	-	-	-	-	.	-	-	-	-	-
10	-	-	-	300	-	-	-	-	-	.	-	-	-	-
11	-	-	-	-	-	200	-	-	-	-	.	-	-	-
12	-	-	-	-	-	-	100	-	-	-	-	.	-	-
13	100	-	-	-	-	-	-	-	-	-	-	-	.	-
14	-	-	-	-	-	-	400	-	-	-	-	-	-	.

TABLE XII
NUMBER OF VNFS INSTANCES f PER NODE FOR THE ORIGINAL TRAFFIC DEMANDS (STAGE 1) AND NEW TRAFFIC DEMANDS (SCENARIOS I AND II OF STAGE 2) FOR OBJECTIVE FUNCTION ONE

Number of VNFs instances f per node $x_{\hat{f},u}$

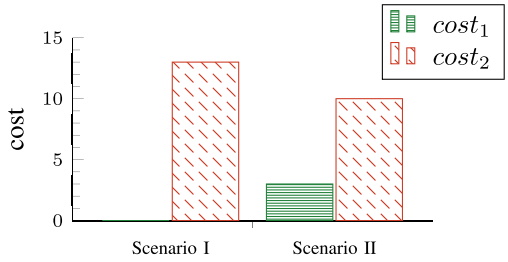
	x_{ij}^{old}			Scen I			Scen II		
node	f=1	f=2	f=3	f=1	f=2	f=3	f=1	f=2	f=3
1	0	20	10	0	60	20	0	50	40
2	0	0	0	0	0	0	0	0	0
3	10	0	10	20	0	20	10	0	10
4	0	0	10	0	0	20	20	0	30
5	0	0	10	0	0	20	0	0	10
6	10	30	10	20	20	20	30	30	20
7	0	0	10	0	0	20	0	0	70
8	20	0	10	20	0	20	40	0	10
9	10	10	10	20	20	20	20	50	20
10	20	20	0	20	40	0	20	50	0
11	20	20	10	60	60	20	40	50	10
12	0	10	0	0	20	0	0	0	0
13	10	0	10	20	0	20	10	0	10
14	10	0	10	40	0	20	40	0	0

It is interesting to observe the significant difference between the optimized and non-optimized values of $cost_1$ and $cost_2$. For instance, Objective Function One, especially designed for minimizing VNF migrations, was able to find a solution with no VNF migration under traffic delineation presented in Scenario I and three VNF migrations under Scenario II, whereas Objective Function Two demanded, for Scenario I and Scenario II, respectively 18 and 19 VNF migrations.

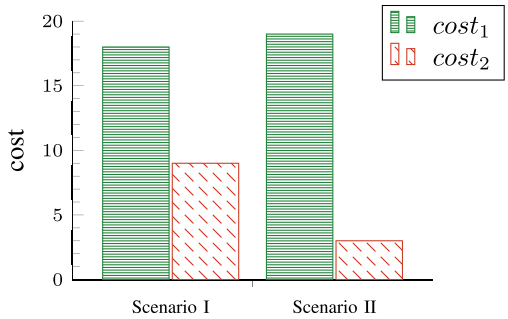
TABLE XIII
NUMBER OF VNFS INSTANCES f PER NODE FOR THE ORIGINAL TRAFFIC DEMANDS (STAGE 1) AND NEW TRAFFIC DEMANDS (SCENARIOS I AND II OF STAGE 2) FOR OBJECTIVE FUNCTION TWO

Number of VNFs instances f per node $x_{\hat{f},u}$

	x_{ij}^{old}			Scen I			Scen II		
node	f=1	f=2	f=3	f=1	f=2	f=3	f=1	f=2	f=3
1	0	20	10	0	40	0	30	0	0
2	0	0	0	0	0	0	0	0	0
3	10	0	10	0	0	20	10	0	10
4	0	0	10	80	0	60	0	10	0
5	0	0	10	0	80	0	70	0	40
6	10	30	10	0	0	40	0	20	30
7	0	0	10	20	0	40	0	70	70
8	20	0	10	20	0	40	0	30	0
9	10	10	10	40	0	0	30	90	40
10	20	20	0	0	40	0	20	0	20
11	20	20	10	20	40	0	50	0	0
12	0	10	0	40	0	0	0	0	10
13	10	0	10	0	20	0	10	0	10
14	10	0	10	0	0	20	10	10	0



(a) MILP (Objective Function One)



(b) MILP (Objective Function Two)

Fig. 5. Values of $cost_1$ and $cost_2$ for Stage 2: MILP simulation in the NSFNET network under traffic Scenarios I and II: (a) Simulation with Objective Function One; (b) Simulation with Objective Function Two.

A similar trend occurs when we analyse the opposite case, where Objective Functions One and Two are employed in Stage 2 of the MILP formulation, but the aim is in reducing the cost specifically treated by Objective Function Two $(cost_2)$. In this case, 9 and 3 node core amendments were required respectively in Scenarios I and II under Objective Function Two, and 13 and 10 under Objective Function One. In resume, Objective Function One required much more changes in the number of cores $(cost_2)$ at the network nodes than Objective Function Two for the analysed scenarios. It is interesting to perceive that, even under a twofold traffic increase present in Scenarios I and II, the MILP formulation was able to assign VNFs (and consequently cores) along the network nodes so that 60% (in Scenario I) and 20% (in Scenario II) of the nodes

TABLE XIV
RUNNING TIME, NSFNET

Objective Function One ($cost_1$)			
MILP		MILP-NL	
Scenario I	Scenario II	Scenario I	Scenario II
0 (0.25s)	3 (0.40s)	0 (≈2h)	3 (≈2h)

TABLE XV
RUNNING TIME, NSFNET

Objective Function Two ($cost_2$)			
MILP		MILP-NL	
Scenario I	Scenario II	Scenario I	Scenario II
9 (0.25s)	3 (0.40s)	9 (≈2h)	3 (≈2h)

were required to increase or decrease their number of cores, which is relevant mainly when one compares with 93% and 71% changes under Objective Function One. These results reemphasize the importance of choosing the proper objective function regarding the intended objective.

The results clearly show that the use of two objective functions provides significant resource savings compared to non-reconfiguration approaches, as also observed for the previously analyzed small network. There is a clear trade-off between the overall required resources for the two objectives.

D. Scalability

For a scalability analysis, the following two settings were compared for the NSFNET:

- MILP-NL: The MILP node link (NL) formulation proposed in [30], without predefined paths, which we have extended with the reconfiguration objectives from Section VI.
- MILP: The path-based formulation proposed in this paper in Section VI, with the same reconfiguration objectives from Section VI.

The advantage of the MILP-NL formulation over the MILP path-based formulation (with predefined paths) is that in the flow networks problems, the former formulation can make effective use of all available network paths to be used by demand flows. This cannot be accomplished so easily by the MILP path-based formulation. Therefore, the solution from a MILP-NL formulation is termed a baseline solution because it provides a mathematically rigorous, optimal solution. This makes it a foundational reference point for evaluating the performance of more practical or scalable methods in solving the network flow problem, as proposed in this paper.

As it can be seen in Table XIV and Table XV, the MILP-NL required about 2h to find the optimal solution for the NSFNET topology either under Objective Function One or Two and Scenario I or II. Our proposed new path MILP formulation was able to obtain the same optimal solution as the MILP-NL. However, it was able to reach the optimum value in fractions of seconds. This is well acceptable compared to the MILP-NL benchmark, which has a computational complexity of $O(N^4)$.

The evidence presented thus far supports the idea that MILP-NL requires defining flow conservation at every node, which can lead to a large number of constraints, especially in large networks. In contrast, the path-based approach works

TABLE XVI
ASSUMED VIRTUAL NETWORK FUNCTIONS AND THEIR ASSOCIATED PROCESSING CAPACITY

id-VNF	VNF name	m_f (Mbps)
NAT	Network Address Translator	10
FW	Firewall	20
TM	Traffic Monitor	30
WOC	WAN Optimization Controller	40
IDPS	Intrusion Detection Prevention System	50

TABLE XVII
ASSUMED SERVICE FUNCTION CHAINS, THEIR MAXIMUM LATENCY AND SEQUENCE OF VNFS

Latency value (t^d_{max})	Service chain	Chained VNFs
<=60ms	Online Gaming (O-G)	NAT-FW-TM-WOC-IDPS
<=100ms	Video Streaming (V-S)	NAT-FW-TM-WOC-IDPS
<=500ms	Web Services (W-S)	NAT-FW-TM-WOC-IDPS

directly with aggregate paths, which can simplify the problem and reduce the computational burden. As a result, the new path formulation makes it possible to solve networks of practical size fast enough to allow network designers and operators to perform extensive "what-if" analysis, so as to investigate large numbers of scenarios regarding forecast demands.

E. Latency Between VNFs

Although NFV brings several benefits, provisioning latency-sensitive network services in a virtualized-based infrastructure remains a challenge, as they require stringent service deadlines. For the NSFNET, in addition to the end-to-end latency, it has been analyzed in this section the robustness of the proposed migration strategy in terms of latency between VNFs.

We consider the following five VNFs, typically employed in service function chaining, to construct our VNF set, namely $F = \{NAT, FW, TM, WOC, IDPS\}$. A detailed description of these VNFs with their assumed processing capacity is given in Table XVI. Table XVII describes the three assumed Service Function Chains that were used in our simulations. They all use the five VNFs, but with very different end-to-end latency requirements.

In all the cases, we have assumed 6 demands of 100Mbps, whereas the source-destination node pairs of the demands are (1-3), (3-1), (5-12), (8-6), (12-4) and (13-6) under Stage 1 and (3-2), (3-6), (4-11), (6-13), (10-9) and (11-14) under Stage 2.

We assume the NSFNet network as the substrate network and that Stage 1 of the MILP formulation has already established six demanded SFCs with random characteristics from Table XVII.

In order to generate VNF latency requirements' diversity between the traffic demands in Stage 1 and Stage 2, we have assumed in Stage 1 that there aren't inter-VNF latency requirements, whereas in Stage 2 two inter-VNF latency requirement scenarios are assumed (Scenario III and Scenario IV), in addition to two latency-absent scenarios (Scenario I and Scenario II), as devoted to Stage 1 specification. All these scenarios are described in Table XVIII. We have also considered that Scenario I and Scenario III do not assume as a constraint the maximum latency acquired in the optimum solution of

TABLE XVIII
SCENARIO STUDIES WITH LIMITED LATENCY BETWEEN VNFS AND $l_{uv} = 10ms$ ON ANY PHYSICAL LINK IN NSFNET FOR FUTURE DEMAND $\hat{D}$. NOTE THAT $f = 1$=NAT, $f = 2$=FW, $f = 3$=TM, $f = 4$=WOC AND $f = 5$=IDPS

Scenarios	$MILP_{LUB}$	$(L^{d,2,1}_{max})$	$(L^{d,4,3}_{max})$
Scenario I	Not	unlimited	unlimited
Scenario II	Yes	unlimited	unlimited
Scenario III	Not	<=10ms	<=10ms
Scenario IV	Yes	<=10ms	<=10ms

TABLE XIX
STAGE 2: SCENARIO RESULTS WITH FUTURE DEMAND $\hat{D}$ FOR THE NSFNET

Cases	Objective 1		Objective 2	
	$cost_1$	$cost_2$	$cost_1$	$cost_2$
Scenario I	15	13	28	7
Scenario II	21	14	30	10
Scenario III	15	12	30	7
Scenario IV	20	13	31	10

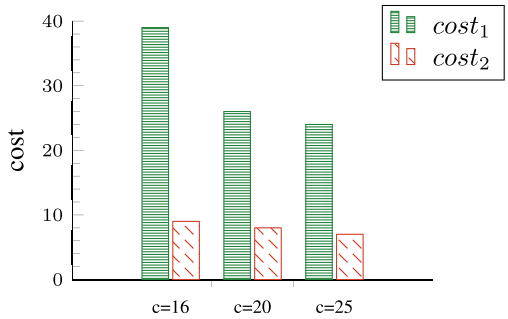
Stage 1, whereas such a constraint is assumed in Scenario II and Scenario IV. As a chain is formed by a large number of VNFs, we assume in this analysis that it is possible to place VNFs at the origin of the demand (either in Stage 1 or Stage 2), making a small modification in the MILP, constraint (29).

Table XIX summarizes the values of $cost_1$ and $cost_2$ when Objective Function One and Two are employed at Stage 2. We can see that, in all analysed scenarios, it was necessary adjustments in the number of VNF migrations $(cost_1)$ and cores $(cost_2)$. In addition, we perceive reductions of 50% and 54% in $cost_1$ and $cost_2$, respectively, when one compares the situation with and without the proper objective function for each cost. In Fig. 6 is shown analyses with Objective Function Two used at Stage 2 of the MILP formulation and the addition of a limit to the number of cores per node. Scenario I and Scenario IV, which represent, respectively, the cases without/with both maximum end-to-end latency and latency between VNFs are considered. It is important to emphasize that just $cost_2$ is the optimum solution in both cases. Notice that the limit of $c = 25$ for the number of cores provides the same $cost_2$ solution as the unbounded core cases under Objective Function Two and Scenarios I and IV (right column at Table XIX).

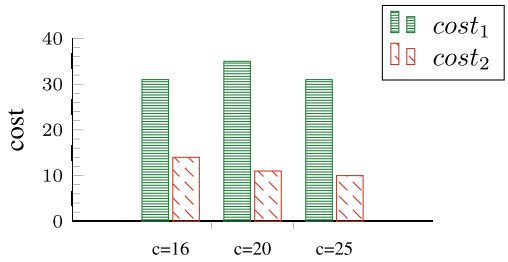
When the constraint to the number of cores in a node is tightened, it becomes harder for the MILP formulation to find a solution with low node core amendments, particularly under Scenario IV and $c = 16$, where an increase of 40% was observed in $cost_2$. This occurs because the few cores available per node, as well as the end-to-end and inter-VNF delay constraints, affect the VNF Placement and routing problem.

VIII. CONCLUSION

This paper addresses the VNF migration, given the network reconfiguration effect. We propose a model to formulate the effect of the network caused by reconfiguration in the VNF placement problem due to new traffic. Since the VNF placement problem is NP-hard, the path MILP is proposed to achieve our objectives, which are minimizing VNF spatial



(a) MILP (Objective Function Two, Scenario I)



(b) MILP (Objective Function Two, Scenario IV)

Fig. 6. Values of $cost_1$ and $cost_2$ for Stage 2: MILP simulation with Objective Function Two in the NSFNET network under different maximum number of cores at any node: (a) Scenario I (b) Scenario IV (more restrictive).

migration and change in the total number of cores per node, considering the resource limits, load balance, latency constraints, and migration cost. The simulation experiment has shown that our proposed MILP largely reduces the network effect of reconfiguration and CPU time compared with the MILP without predefined paths.

Future research could extend this work by integrating machine learning models to predict traffic patterns, further optimizing SFC reconfiguration processes. Additionally, exploring real-world implementations and incorporating multi-cloud environments would provide deeper insights into the practical adaptability of the proposed solutions. This study underscores the potential of advanced mathematical optimization methods in addressing emerging challenges in NFV and SFC management/planning.

REFERENCES

[1] J. M. Halpern and C. Pignataro, "Service function chaining (SFC) architecture," Internet Eng. Task Force, RFC 7665, Oct. 2015. [Online]. Available: https://www.rfc-editor.org/info/rfc7665

[2] D. Bhamare, R. Jain, M. Samaka, and A. Erbad, "A survey on service function chaining," *J. Netw. Comput. Appl.*, vol. 75, pp. 138–155, Nov. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804516301989

[3] S. Kim, Y. Han, and S. Park, "An energy-aware service function chaining and reconfiguration algorithm in NFV," in *Proc. IEEE 1st Int. Workshops Found. Appl. Self Syst.*, 2016, pp. 54–59.

[4] Z. Xiao, X. Fu, L. Zhang, and R. S. M. Goh, "Traffic pattern mining and forecasting technologies in maritime traffic service networks: A comprehensive survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1796–1825, May 2020.

[5] R. Viola, A. Martín, M. Zorrilla, J. Montalbán, P. Angueira, and G.-M. Muntean, "A survey on virtual network functions for media streaming: Solutions and future challenges," *ACM Comput. Surveys*, vol. 55, no. 11, pp. 1–37, Feb. 2023. [Online]. Available: https://doi.org/10.1145/3567826

[6] P. Tam, S. Kang, S. Ros, I. Song, and S. Kim, "Large-scale service function chaining management and orchestration in smart city," *Electronics*, vol. 12, no. 19, p. 4018, 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/19/4018

[7] P. Bellini, P. Nesi, and G. Pantaleo, "IoT-enabled smart cities: A review of concepts, frameworks and key technologies," *Appl. Sci.*, vol. 12, no. 3, p. 1607, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/3/1607

[8] B. N. Silva et al., "Urban planning and smart city decision management empowered by real-time data processing using big data analytics," *Sensors*, vol. 18, no. 9, p. 2994, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/9/2994

[9] G. Gardikis et al., "SHIELD: A novel NFV-based cybersecurity framework," in *Proc. IEEE Conf. Netw. Softw. (NetSoft)*, 2017, pp. 1–6.

[10] Z. A. Qazi, C.-C. Tu, L. Chiang, R. Miao, V. Sekar, and M. Yu, "SIMPLE-fying middlebox policy enforcement using SDN," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 3, pp. 27–38, 2013. [Online]. Available: https://doi.org/10.1145/2486001.2486022

[11] K. Kaur, V. Mangat, and K. Kumar, "A comprehensive survey of service function chain provisioning approaches in SDN and NFV architecture," *Comput. Sci. Rev.*, vol. 38, Nov. 2020, Art. no. 100298. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013720303981

[12] D. Petcu, "Multi-cloud: Expectations and current approaches," in *Proc. Int. Workshop Multi-Cloud Appl. Fed. Clouds*, 2013, pp. 1–6. [Online]. Available: https://doi.org/10.1145/2462326.2462328

[13] J. Li, Y. K. Li, X. Chen, P. P. Lee, and W. Lou, "A hybrid cloud approach for secure authorized deduplication," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 5, pp. 1206–1216, May 2015.

[14] P. Rodis and P. Papadimitriou, "Intelligent and resource-conserving service function chain (SFC) embedding," *J. Netw. Syst. Manage.*, vol. 31, no. 4, p. 81, 2023.

[15] P. Popovski et al., "A perspective on time toward wireless 6G," *Proc. IEEE*, vol. 110, no. 8, pp. 1116–1146, Aug. 2022.

[16] A. Larrañaga, M. C. Lucas-Estañ, I. Martinez, I. Val, and J. Gozalvez, "Analysis of 5G-TSN integration to support industry 4.0," in *Proc. 25th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, vol. 1, 2020, pp. 1111–1114.

[17] B. Yi, X. Wang, and M. Huang, "Design and evaluation of schemes for provisioning service function chain with function scalability," *J. Netw. Comput. Appl.*, vol. 93, pp. 197–214, Sep. 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804517302102

[18] M. Mechtri, C. Ghribi, and D. Zeghlache, "A scalable algorithm for the placement of service function chains," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 3, pp. 533–546, Sep. 2016.

[19] W. Liang, L. Cui, and F. P. Tso, "Low-latency service function chain migration in edge-core networks based on open Jackson networks," *J. Syst. Archit.*, vol. 124, Mar. 2022, Art. no. 102405. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S138376212200008X

[20] T.-M. Pham, "Optimizing service function chaining migration with explicit dynamic path," *IEEE Access*, vol. 10, pp. 16992–17002, 2022.

[21] Y. Qin, D. Guo, L. Luo, J. Zhang, and M. Xu, "Service function chain migration with the long-term budget in dynamic networks," *Comput. Netw.*, vol. 223, Mar. 2023, Art. no. 109563. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128623000087

[22] H. Hantouti, N. Benamar, and T. Taleb, "Service function chaining in 5G & beyond networks: Challenges and open research issues," *IEEE Netw.*, vol. 34, no. 4, pp. 320–327, Jul./Aug. 2020.

[23] Y. Xie, Z. Liu, S. Wang, and Y. Wang, "Service function chaining resource allocation: A survey," 2016, *arXiv:1608.00095*.

[24] M. F. Bari, S. R. Chowdhury, R. Ahmed, and R. Boutaba, "On orchestrating virtual network functions," in *Proc. 11th Int. Conf. Netw. Service Manage. (CNSM)*, 2015, pp. 50–56.

[25] L. Wang, Z. Lu, X. Wen, R. Knopp, and R. Gupta, "Joint optimization of service function chaining and resource allocation in network function virtualization," *IEEE Access*, vol. 4, pp. 8084–8094, 2016.

[26] Y. Yue, W. Yang, X. Zhang, R. Huang, and X. Tang, "A QoS guarantee mechanism for service function chains in NFV-enabled networks," in *Proc. Int. Conf. Comput. Commun. Netw. (ICCCN)*, 2022, pp. 1–2.

[27] A. Mouaci, É. Gourdin, I. LjubiĆ, and N. Perrot, "Virtual network functions placement and routing problem: Path formulation," in *Proc. IFIP Netw. Conf. (Netw.)*, 2020, pp. 55–63.

[28] H. P. Williams, *Model Building in Mathematical Programming*. Hoboken, NJ, USA: Wiley, 2013.

[29] "IBM CPLEX solver: Download, pricing & documentation—AMPL." Accessed: Feb. 13, 2025. [Online]. Available: https://ampl.com/products/solvers/solvers-we-sell/cplex/

[30] Z. Allybokus, N. Perrot, J. Leguay, L. Maggi, and E. Gourdin, "Virtual function placement for service chaining with partial orders and anti-affinity rules," *Networks*, vol. 71, no. 2, pp. 97–106, 2018.