




Modeling and Analysis of Different Reconfiguration Strategies for Virtual Network Function Placement and Chaining with Service Classes Identification

Samuel M. A. Araújo , Fernanda S. H. de Souza , and Geraldo R. Mateus 

Abstract—The Virtual Network Function Placement and Chaining problem (VNF-PC) is an important part of the Network Functions Virtualization (NFV) based-technologies implementation. VNF-PC problem focuses on the allocation of customer demands on the Substrate Network. Aiming to evaluate the impact of diverse modeling, various reconfiguration strategies based on implicit steps in solving the VNF-PC are proposed: resizing virtual network instances, re-routing chaining, and repositioning Network Functions (NF) instances on different servers. In addition, this work analyzes, compares, and discusses the advantages and disadvantages of each proposed reconfiguration strategy in an online scenario. Traditionally, VNF-PC solutions from literature process requests generated at random and do not take into account real-world demands. Complementing the analyses of reconfiguration strategies, works from the literature are surveyed to identify commonly used Network Services (NS). Following that, these NS are classified into service classes, and used to generate realistic requests to be mapped in the experimental stage of the reconfiguration approaches. The experiments are conducted using realistic requests and a real-world network topology. An Integer Linear Programming model is used to process the requests. Simulations show that repositioning NF instances can generate up to $\approx 25\%$ more profit than reconfiguring only the VNF instances, but the processing time increases by up to 99.99%. On the other hand, resizing virtual network instances and re-routing the chaining had no significant impact on runtime.

Index Terms—Network function virtualization, service function chaining, reconfiguration, classes of service.

I. INTRODUCTION

Among recent Internet innovation proposals, the use of network layer virtualization mechanisms has stood out for supporting new application perspectives, benefiting not only customers but also providers [1], [2]. NFV-based networks, as opposed to middlebox-based networks, have grown in popularity due to inherent benefits in resource sharing as well as a more flexible response to market demands [1]–[5].

NFV is a promising technology that is particularly important for emerging 5G/6G and IoT services such as ultra-reliable, low latency and delay communications [5]–[7]. Furthermore, NFV-based networks enable slicing as a novel service provisioning paradigm, boosting application development. With network slicing, VNFs can be instantiated at various SN's

points, with the placement and chaining determined by factors as maximum allowable delays and available resources [7].

NFV-based networks decouple NF from middleboxes and move them to virtualized servers that are emulated on Commercial-Off-The-Shelf hardware [2]. A Firewall is an example of NFs that can be virtualized [3]. The acronyms used in this paper are listed in Table 1. The well-known acronyms were omitted as they were considered unnecessary.

TABLE I
TABLE OF ACRONYMS

Symbol	Definition	Symbol	Definition
COTS	Commercial-Off-The-Shelf	FW	Firewall
IDS	Intrusion Detection System	IoT	Internet of Things
ILP	Integer Linear Programming	mIoT	massive IoT
NAT	Network Address Translation	NS	Network Services
NF	Network Function	NFV	NF Virtualization
QoS	Quality-of-Service	SFC	Service Function Chain
SN	Substrate Network	TM	Traffic Monitor
VOC	Video Optimization Controller	VNF	Virtual Network Function
VoIP	Voice over Internet Protocol	WAN	Wide Area Network
WOC	WAN Optimization Controller		

Following the definition proposed in [8], a set of VNFs sequenced (chained) for the purpose of serving a particular NS is called a SFC. In case, end-customers may demand different SFC configurations to meet their NS, *e.g.*, augmented reality: NAT-FW-TM-VOC-IDS and IoT: NAT-FW-IDS. As an example of an IoT-based NS, the authors of [9], propose the use of medical applications to predict disease in real time. Still addressing IoT-based NS, according to [4], in order to address vulnerability and security aspects, machine learning techniques can be employed in VNF sequence.

Among these SFCs, some of them demand a low end-to-end delay and must be executed in real time. Style transfer algorithms in photographs are an example of a network real-time application that is sensitive to end-to-end delays [10]. Another current NS is IoT-based video surveillance. In this case, the referred devices require a high bandwidth and data processing with minimal end-to-end latency [11]. An in-depth classification of these services is shown in Section V.

A. Background

One of the most difficult challenges in deploying NFV networks is allocating resources to meet user requirements. In this regard, providers provide virtualization layers as a service. There are several issues with allocating resources demanded by users over those offered by providers. In this paper, the problem investigated is known as VNF-PC. In case, SFCs

Samuel M. A. Araújo, Federal University of Minas Gerais Brazil e-mail:smaa@dcc.ufmg.br.

Fernanda S. H. de Souza, Federal University of Ouro Preto Brazil e-mail:fernanda.souza@ufop.edu.br

Geraldo R. Mateus, Federal University of Minas Gerais Brazil e-mail:mateus@dcc.ufmg.br.

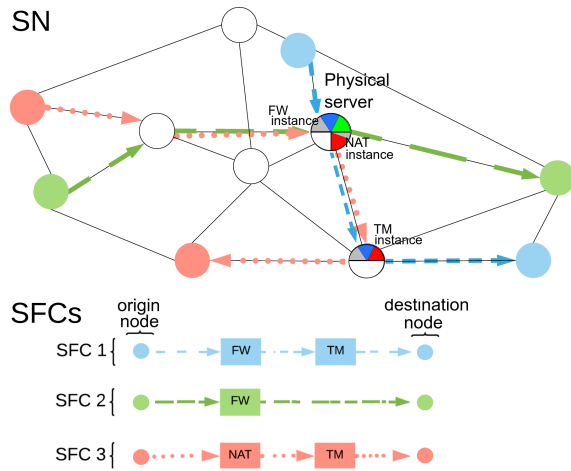


Fig. 1 Example of resource sharing between 3 SFCs.

must be mapped, which requires allocating the required VNF instances, clients positioning, and generating a route between the instantiated components (Fig. 1). The difficulties in solving the VNF-PC are related to its combinatorial nature, being a NP-hard problem [3].

As an example, three SFCs are mapped on the same SN and share the same server (Fig. 1). This action allows the SFCs involved to share the costs of keeping the respective server operational. In the example, two NF instances can serve three SFCs simultaneously; in this case, the instance of FW attends SFCs 1 and 2, and the instance of NAT attends SFC 3. In each VNF instantiated, there is an excess of unused resources within the instance itself (gray part), which can be used if any SFC increases its demand in the future. The notation used in Fig. 1 can be seen in Table I and Section III

Implementing virtual resources with availability and reliability is becoming increasingly important for the successful deployment of real-time NS. NFV-based networks use dynamic virtual resource systems capable of delivering the various network services required in an automated model.

An offline scenario is characterized by mapping an entire set of SFCs simultaneously. In case, the algorithm benefits from knowing the topologies, duration times, and other SFCs' characteristics. Given the dynamic requirements of NFV networks, there is a need for resource allocation proposals that can find online solutions [1]. Online scenarios are characterized by unknown arrival of SFCs demands, with no knowledge of topology, lifetime, or demanded requirements. In the online scenario, there may eventually be active SFCs in the SN, allocated and chained at an earlier time, *i.e.*, there are residual capacities of the original infrastructure available.

During the development of the paper [29], some research questions were identified as relevant, but not addressed by the literature. Thus, the research questions analyzed in this paper arise, being: *i)* When a new SFC to be mapped arrives, is it necessary to reconfigure the active SFCs on the SN? *ii)* Will the reconfiguration increase the profit of the providers? *iii)* How does using reconfiguration of all active SFCs affect the runtime? *iv)* When mapping SFCs from different classes of service, how do reconfiguration approaches behave?

B. Main contributions

As illustrated in literature works [12], [13], each SFC provides a unique NS. The first contribution of this work was the study and summarization of the main properties of different network services, in order to propose a service classification and process. The second contribution concerns the analysis of reconfiguration strategies aiming to identify trade-offs related to the different classes of service and the reconfiguration models. The reconfiguration models are as follows:

- i)* VNF instance resizing: re-optimizing VNF instances may be beneficial when it is necessary to increase or decrease an instance's capacity, implying a more rational use of resources;
- ii)* VNF instance resizing and SN path rerouting (chaining): when there is a high fragmentation of SN resources in the SN, re-optimization of virtual arcs can be beneficial, making it possible to access potentially isolated servers due to a lack of bandwidth on adjacent arcs;
- iii)* VNF instance resizing, SN path rerouting (chaining), and instance reallocation (placement): it is assumed that re-optimizing VNF instance allocation constraints can be useful when changing the allocation of one or more instances.

An exact ILP model is proposed to compare reconfiguration strategies. The importance of adopting the exact approach in this type of modeling lies in the analysis of the results. In this case, there are no approximations in decision making and biases that can potentially occur with heuristics. To represent a realistic scenario, a real-world network topology is used as SN, and the SFCs are based on different service classes, that cover a wide range of market applications.

The remainder of this paper is organized as follows. Section II presents relevant related work. In Section III, the VNF-PC problem is defined. Section IV presents the exact model used and the reconfiguration constraints applied. Section V shows the classes of service identified and adopted in this article. In Section VI we validate our reconfiguration proposals through computational experiments in a real-world scenario, and we discuss the results. Finally, we conclude in Section VII.

II. RELATED WORK

The virtualization technology, which refers to abstracting network resources that were previously delivered in hardware to software, has received a lot of attention in recent years [1], [2], [7]. NFV technology is fully explained in detail in [2]. Kaur *et al.* [2] presented some concepts in this area, showing a standardization of terminology and defining some use cases. According to Yi *et al.* [1], although different articles consider diverse objectives, VNF-PC constraints are generally the same and include resource allocation, and VNF placement and chaining. This paper work differs from the literature by focusing on the impact of reconfiguration strategies on the mapping of SFCs rather than evaluating the solution approach or making comparisons with existing approaches. Unlike the current literature, different SFCs with real market characteristics are used to add value to the proposed model.

According to Bagaa *et al.* [5], the SFC orchestration is an optimization problem that shall consider different constraints, such as end-to-end delay, bandwidth, and QoS. Bagaa *et*

al. presented a orchestration cost-efficient model, aimed to optimize the allocation and scheduling of NS. The work [7] focuses on the network slicing problem. According to the authors, one limitation of network slicing in NFV networks is the inability to re-optimize the SFC after the slice is done. Similarly, we examine reconfiguration trade-offs, but with a focus on wired networks, a precise analysis, and a variety of re-optimization strategies. Addressing the 5G communication networks, Le *et al.* [14] applies clustering techniques to identify applications with similar patterns. The generated clusters are used to build a classifier in order to better understand the traffic and accommodate it over the SN.

The approach proposed by Liu *et al.* [15] is based on ILP and column generation to solve the VNF-PC, and maps each SFC performing a reconfiguration of the previous active SFC. According to [16], recent advances in some IoT-based NS are trending towards high bandwidth demands and low end-to-end delay, and these constraints must be considered. The approaches proposed by Liu *et al.* [15] use pre-computed network paths and do not consider the end-to-end delay. Differently, these challenges are addressed in our formulation.

Padhy *et al.* [13] solve the VNF-PC with reconfiguration using an approach based on ILP formulation with implementation in CPLEX solver. Padhy *et al.* focus on aspects of dynamic data flow, and do not explore aspects of maximum end-to-end delay and processing consumption. The VNF-PC is solved in [17] by minimizing costs and applying end-to-end delay constraints. However, the authors ignore the costs associated with using instances of VNFs. In this work, delays for nodes and physical links are treated similarly to [17].

III. VNF-PC DEFINITION

The problem definition can be summarized as follows:

A. SN

The SN is represented as a weighted directed $G = (N, L)$, where N and L denote the set of servers and arcs. We assume that VNFs can be deployed on commodity servers located within any node on the network. Each node $i \in N$ has CPU (core) and memory (MB) capacities, represented by C_i , M_i . Each arc $(i, j) \in L$ has a bandwidth capacity BW_{ij} (Mbps) and an end-to-end delay (milliseconds). The function $delay(i, j)$ represents the end-to-end delay.

B. VNF

Different types of VNFs can be instantiated. Let F denote the set of NFs. Each NF $f \in F$ may be demanded by a SFC, assigned and virtualized over an already placed VNF instance. Set B_f represents the possible NFs instances of each type $f \in F$, which can be instantiated on a node $i \in N$. Each instance $b \in B_f$ demand a specific amount of CPU C_{bf} (core) and memory M_{bf} (MB). When a NF $b \in B_f$ is instantiated, it offers parts of C_{bf} and M_{bf} to attend some SFC, demands a fixed portion of resources C_{bf} and M_{bf} from SN servers. It is assumed that each mapped instance $b \in B_f$ may be shared by different SFCs. Each NF $f \in F$ has a processing delay

and generates a packet gain or loss on traffic data, caused by the processing associated with the NF demanded. The function $delay(f)$ returns the delay generated to process the NF $f \in F$.

C. SFCs

Different SFC configurations can be demand by end-users. Let V represent the set of SFCs. Each SFC $v \in V$ has an arrival time t_{in}^v , a duration time t_{dr}^v , and a maximum delay supported t_{dl}^v . Each SFC $v \in V$ is represented as a weighted directed graph $G^v = (N^v, L^v)$, where N^v and L^v represent the set of virtual nodes and arcs. Let $N^v = \{s^v \cup d^v \cup F_f^v\}$ denote the set of the join of the origin node s^v , destination node d^v ; and F_f^v the set of VNFs demanded ($F_f^v \subseteq F$). The nodes s^v and d^v do not demand resources, but must be assigned to nodes on the same geographical position. Each VNF $k \in F_f^v$ demands CPU (c_k^v) and memory (m_k^v), and generates a processing delay $delay(k)$. Each virtual arc has a bandwidth demand bw_{kl}^v , which can be altered by the gain/loss of data, resulting from the previously processed VNF.

D. VNF-PC solution

Consists in finding a mapping: $G^v \rightarrow G$. This action can be broken down into the assignment and placement of VNFs and origin/destination points. In this case, each node of the same SFC must be placed on a different SN node; and routing of virtual arcs, where each virtual arc may be mapped on a single arc, or on a path with more than one SN arc. Finally, the total delay caused by VNFs and SN arcs must be less than t_{dl}^v . If such an embedding is possible, the request is said to be accepted; otherwise, it is refused.

IV. MODEL AND RECONFIGURATION CONSTRAINTS

The VNF-PC problem mathematical formulation and reconfiguration constraints are presented in this section.

A. Variables and functions

Table II details the notation used in the model proposed in the following section. The first part of the table introduces the variables adopted. The second part explains the functions used to build the ILP model.

TABLE II
GLOSSARY OF SYMBOLS: VARIABLES AND FUNCTIONS

Variables	Definition
$y^v \in \{0, 1\}$	Indicates if the SFC $v \in V$ was successfully mapped
$s_i \in \{0, 1\}$	Indicates if the SN server $i \in N$ is being used
$w_{bf}^i \in \{0, 1\}$	Indicates if an instance $b \in B_f$ of the function $f \in F$ of the SN server $i \in N$ is being used
$z_{ki}^v \in \{0, 1\}$	Indicates if the virtual node $k \in N^v$ of the SFC $v \in V$ was assigned to the SN server $i \in N$
$x_{ij}^{vkl} \in \{0, 1\}$	Indicates if the virtual arc $(k, l) \in L^v$ of SFC $v \in V$ was mapped on the SN arc $(i, j) \in L$
Functions	Definition
$type(k) \rightarrow f \in F$	Returns NF type $f \in F$ of node $k \in N^v$
$delay(f) \rightarrow R_+$	Returns the delay of NF $f \in F$
$delay(i, j) \rightarrow R_+$	Returns the end-to-end delay of the arc $(i, j) \in L$

B. Offline Model Constraints

$$\sum_{f \in F} \sum_{b \in B_f} w_{b,f}^i C_{b,f} \leq C_i, \forall i \in N \quad (1)$$

$$\sum_{f \in F} \sum_{b \in B_f} w_{b,f}^i M_{b,f} \leq M_i, \forall i \in N \quad (2)$$

$$\sum_{v \in V} \sum_{\substack{k \in F_f^v : \\ \text{tipo}(f) = \text{tipo}(k)}} z_{k,i}^v c_k^v \leq \sum_{b \in B_f} w_{b,f}^i C_{b,f}, \forall i \in N, \forall f \in F \quad (3)$$

$$\sum_{v \in V} \sum_{\substack{k \in F_f^v : \\ \text{tipo}(f) = \text{tipo}(k)}} z_{k,i}^v m_k^v \leq \sum_{b \in B_f} w_{b,f}^i M_{b,f}, \forall i \in N, \forall f \in F \quad (4)$$

$$\sum_{i \in N} z_{k,i}^v = y^v, \forall k \in F_f^v, \forall v \in V \quad (5)$$

$$\sum_{i \in N: i=k} z_{k,i}^v = y^v, \forall k \in \{s^v \cup d^v\}, \forall v \in V \quad (6)$$

$$\sum_{k \in N^v} z_{k,i}^v \leq 1, \forall i \in N, \forall v \in V \quad (7)$$

$$\sum_{k \in N^v} z_{k,i}^v \leq S_i, \forall i \in N, \forall v \in V \quad (8)$$

$$\sum_{v \in V} \sum_{(k,l) \in L^v} x_{ij}^{vkl} b w_{kl}^v \leq B W_{ij}, \forall (i,j) \in L \quad (9)$$

$$\sum_{(i,j) \in L} \sum_{(k,l) \in L^v} \text{delay}(i,j) x_{ij}^{vkl} + \sum_{i \in N} \sum_{k \in F_f^v} \text{delay}(k) z_{k,i}^v \leq t_{dl}^v, \forall v \in V \quad (10)$$

$$\sum_{(i,j) \in L} x_{ij}^{vkl} - \sum_{(h,i) \in L} x_{hi}^{vkl} = z_{k,i}^v - z_{l,i}^v, \forall i \in N, \forall (k,l) \in L^v, \forall v \in V \quad (11)$$

Constraint sets (Eq. 1) and (Eq. 2) ensure that all instances $b \in B_f$, of NF $f \in F$ do not exceed the CPU C_i and memory M_i bounds existing on SN servers $i \in N$. Constraint sets (Eq. 3) and (Eq. 4) guarantee that placing all VNFs $k \in F_f^v$ over images $b \in B_f$ do not exceed the bounds of CPU $C_{b,f}$ and memory $M_{b,f}$. Such constraints, together with constraint sets (Eq. 1) and (Eq. 2) complete the VNFs placement. Constraint set (Eq. 5) enforces that all VNFs $k \in F_f^v$ are mapped for the SFC $v \in V$ to be accepted. Such constraints correspond to the assignment step of a VNF on SN nodes. Constraint set (Eq. 6) ensures that all endpoints (origin and destination) are mapped. Constraint set (Eq. 7) guarantees that all virtual nodes $k \in N^v$ are assigned to different SN nodes $i \in N$ for each SFC $v \in V$. These constraints are used to reduce the impact of a failure of the SN infrastructure [8]. Constraint set (Eq. 8) denotes which SN nodes $i \in N$ are being used. Constraints (Eq. 9) ensure that the summation of flows in each direction of the directed arc remains within its available bandwidth. Constraints (Eq. 10) guarantee that the summation of delays remains within the maximum allowed delay t_{dl}^v . The constraints (Eq. 11) are classic and apply principles of flow conservation to guarantee the mapping of each virtual arc $(k,l) \in L^v$ of each SFC $v \in V$ on paths of SN arcs $(i,j) \in L$.

C. Online Model Reconfiguration Constraints

When solving VNF-PC in online environments, there may be active SFCs that were previously placed and chained on the SN, represented by the set $V^{at} \subset V$. In this case, when mapping a new SFC, it must be ensured that the SFCs $v \in V^{at}$, active at time t , continue to have their services attended at a subsequent time t' . Therefore, new requests must be processed

following the constraints presented in the Subsection IV-B (Constraints (1) to (11)), and, at the same time, guarantee the solution's feasibility, with the re-optimization of the SFCs $v \in V^{at}$. For this, 3 re-optimization strategies are proposed:

i) NF instance resizing: the requests $v \in (V \cup V^{at})$ are processed together, but only for the SFCs $v \in V^{at}$ the resizing of the placed VNF instances is allowed (re-optimize constraints 1 to 4). By increasing an instance's capacity, one or more new VNFs can share the same VNF instance. By reducing an instance's capacity, a more concise and efficient use of resources is aimed, as well as a higher profit, because higher capacity NF instances are more expensive. This action, if performed individually, has polynomial complexity;

ii) NF instance resizing and SN path rerouting (chaining): the requests are $v \in (V \cup V^{at})$ processed together, but only for the SFCs $v \in V^{at}$, in addition to resizing the placed NFs instances, the rerouting of the used arcs is allowed (re-optimize constraints 1 to 4 and 10 to 12). When there is a SN resources fragmentation, which makes it impossible to access potentially isolated servers due to a lack of bandwidth on adjacent arcs, paths rerouting can be beneficial. Or even if it is required to meet a new SFC with constrained end-to-end delay. In this case, an old SFC with a less constrained end-to-end delay demand may have its routing changed to release resources previously allocated to the new SFC. This action is a variant of the Unsplittable Multicommodity Flow problem and has exponential complexity;

iii) NF instance resizing, SN path rerouting (chaining), and instance reallocation (placement): the requests $v \in (V \cup V^{at})$ are processed together, but only for the SFCs $v \in V^{at}$, in addition to resizing the placed VNF instances and rerouting the arcs, it is also possible to change the placement of the NF instances (reconfigures all constraints). This action is motivated by a server's low processing capacity or residual memory, which makes resource sharing impossible; or to reduce a SFC's end-to-end delay. This action can be viewed as a variant of the Virtual Network Embedding [18]. Only this strategy has guarantees of a globally optimal solution.

D. Objective Function

The objective function aims to maximize the total profit (Eq. 15). The revenue (R , Eq. 12) shows how much the providers will earn by accepting a new SFC. In this case, α represents the revenue charged per $Mbps$ of used bandwidth, β per core of used CPU, and γ per MB of used memory. The arc Cost (LC , Eq. 13), measures the provider's cost to serve each virtual arc of each SFC. This value is given to each $Mbps$ (δ) reserved for traffic on each SN arc used by a SFC. Server Cost (SC , Eq. 14), measures the cost to serve each VNF. Cost is assumed to be monetary cost. In this situation, there are different costs involved: cost of keeping the server $i \in N$ active (ϵ); cost of NF instantiation (ξ_b), memory cost (ζ) and the processing cost (ϵ). The ILP model induces the variable y^v of each SFC $v \in V$ to assume the value 1, ensuring the feasible mapping of all requests. Therefore, the objective function generates a monetary gain of each SFC successfully mapped, and minimizes the costs with sharing of physical resources.

$$R = \sum_{v \in V} y^v \left(\sum_{(k,l) \in L^v} bw_{kl}^v \alpha + \sum_{k \in N_f^v} (c_k^v \beta + m_k^v \gamma) \right) \quad (12)$$

$$LC = \sum_{v \in V} \sum_{(k,l) \in L^v} \sum_{(i,j) \in L} x_{ij}^{vkl} \theta_{kl}^v \delta \quad (13)$$

$$SC = \sum_{i \in N} \left(S_i \epsilon + \sum_{f \in F} \sum_{b \in B_f} w_{bf}^i \xi_b + \sum_{v \in V} \sum_{k \in F_f^v} z_{ki}^v (c_k^v \epsilon + m_k^v \zeta) \right) \quad (14)$$

$$\text{MAXIMIZE } R - LC - SC \quad (15)$$

V. SERVICE CLASSES DEFINITION

The SFCs proposed in this work are analyzed and categorized as belonging to classes of services, following a concept proposed in [19]. The examples of classified SFCs were taken from [6], [20]–[22] and shown in Table III. Although each NS is assigned to a different class, if necessary or if market demands change, such NS can be reclassified and/or the parameterization of the classes changed. To this end, consider t_{dr}^v as the duration time of a SFC and bw as the bandwidth reservation in *Mbps* demanded to meet the QoS constraints.

TABLE III
NETWORK SERVICES CLASSIFICATION

Real-time applications (Class 1)	VNFs chaining	bw	t_{dr}^v
Videoconferencing	NAT-FW-TM-VOC-IDS	5	3000
Online Games	NAT-FW-VOC-WOC-IDS	20	10800
Smart Factory	NAT-FW	20	900
VoIP	NAT-FW-TM-FW-NAT	0.064	180
Augmented Reality (5G)	NAT-FW-TM-VOC-IDS	5	1800
Interactive applications (Class 2)	VNFs chaining	bw	t_{dr}^v
low bandwidth web service	NAT-FW-TM-WOC-IDS	0.2	1
high bandwidth web service	NAT-FW-TM-WOC-IDS	10	1
Security requirements	PROXY-FW-IDS	0.1	1
File transfer applications (Class 3)	VNFs chaining	bw	t_{dr}^v
Video streaming Ultra HD	NAT-FW-TM-VOC-IDS	25	7200
Audio streaming	NAT-FW-TM-IDS	0.32	120
mIoT	NAT-FW-IDS	1	90
NB-IoT	NAT-FW-IDS	0.25	60

Content based on [12], [23]–[29]

- Class 1: there are strict t_{dl}^v requirements for this class. According to Alleg *et al.* [6], the NS in this class can tolerate end-to-end delay of up to 150 milliseconds (*ms*).
- Class 2: this class is more sensible to end-to-end delay than file transfer applications, but less sensitive than real-time applications. This class include NSs with real-time traffic characteristics. According to [6], the NSs in this class can tolerate end-to-end delay of up to 300*ms*.
- Class 3: this class include NS that are less sensitive to QoS metrics and/or have best-effort characteristics (non-guaranteed packet delivery). No problem occurs in this class if all packets are uniformly delayed by a few seconds.

VI. PERFORMANCE EVALUATION

In order to evaluate the different reconfiguration strategies, the ILP model was implemented and run in a simulator built in C language using CPLEX 12.6 API. All experiments were carried out on an Intel i3-8300 8th 3.20GHz 16GBDDR4 computer, and the Ubuntu 20.04 operating system.

A. VNF-PC metrics

Metrics are necessary to evaluate the quality of a solution. Three metrics, based on works [30], are used to quantify and compare different re-optimization strategies:

- Profit: the main metric, adopted as the objective function of the model (Eq. 15). Given by the difference in revenue generated by mapping a SFC and the expenses;
- Runtime: an important aspect to evaluate in the perspective of application of our reconfiguration strategies in real network environments.
- Acceptance ratio: metric for analyzing the rejections generated, as well as for supporting aspects related to profit analysis. Consists in the ratio of the mapped SFCs' number to the total number of SFCs that arrived until time t .

B. Simulation Setup

The first part of the Table III details the parameters adopted to configure the SN servers. The second part explains the characteristics of the NFs adopted in the experiments. Finally, the table shows the SN physical servers' characteristics.

1) *SN*: The physical topology used to generate SN, known as CompuServe Network, has 11 nodes and 14 arcs (available online in www.topology-zoo.org). Similarly to [30], we assume that the SN is composed of COTS servers (Table IV). Each SN node is assigned randomly a machine with the same resources as the Amazon EC2 network. The SN arcs capacities are randomly selected in $BW_{ij} \in \{20, 40, 60, 80, 100\}$. The estimation of each SN arc delay is defined by the Haversine formula (equation that calculates the shortest distance between two points on Earth's surface). Similar to [31], this value is perturbed by multiplying a random number in $[0.008; 0.012]$, to simulate a variability of delays that can exist in real networks. Various operation prices must be considered to define the costs of the SN components. In this work, we considered a concise model, but it can be altered if the model is adopted in real environments. These values are $\delta = 0.025, \epsilon = 0.125, \zeta = 0.25$ and $\epsilon = 30, \alpha = 0.05, \beta = 0.25, \gamma = 0.5$. These values aren't real, but were generated after a search for reference values in the literature, and they are in agreement with work [30].

2) *VNFs*: Table IV also shows the CPU and memory requirements for each VNF demanded. It is assumed that for each type of VNF $f \in F$, there are 5 distinct types of instances that can be used. Each instance has memory capacity (M_{bf}), CPU (C_{bf}) and fixed cost (ϵ_{bf}) (Table IV).

3) *SFC*: SFCs composition is based on the service classes described in Section V. Aforesaid SFCs are generated similarly to [12], [30]; in this case, two SN servers are randomly chosen from Table IV, and defined as the origin and destination endpoints of each SFC. It is assumed that, due to the attached NSs and the SN topology, the range of SFC arrivals follows a *Poisson* distribution with mean $\lambda = 600$ in Class 1, $\lambda = 400$ in Class 2, and $\lambda = 500$ in Class 3. For the experiments, 20000 time units (t) are considered, giving rise to the following scenarios:

- *Scenario 1 (S1)*: It consists of class 1 SFCs, with $\frac{1}{5}$ of requests of each type defined in the Table III, considering 20000*t*, 33 SFCs are processed;

TABLE IV
SN SERVER CONFIGURATIONS, VNFs AND VIRTUAL INSTANCES.

SN server configurations			
SN processor	CPU C_i (core)	Memory M_i (GB)	
Xeon Scalable	48	1800	
AWS Graviton2	64	3600	
AWS Graviton2	64	3800	
Xeon Scalable	96	1800	
Intel Xeon Platinum	96	3600	
AMD EPYC 7002	96	3800	

(Content available in www.aws.amazon.com/pt/ec2/instance-explorer)

NF	c_k	NFs adopted		Delay (ms)
		m_k	Flow increase or decrease	
Proxy	4	200	0.9	0.25
FW	4	400	0.9	0.8
TM	10	300	0.9	0.1
IDS	8	800	0.8	0.01
NAT	16	400	1.0	0.1
WOC	2	800	1.1	0.2
VOC	8	1000	1.2	0.25

Amazon EC2 Virtual Servers			
Virtual Server	CPU C_{bf}	memory M_{bf}	Pricing \$ ξ_b
M6 Double Extra Large	8	475	0.361
M6 Quadruple Extra Large	16	950	0.723
M5 Eight Extra Large	32	1200	1.648
M6 16 Extra Large	64	1900	2.892
M5 24 Extra Large	96	3600	5.424

(Price defined by the instance-hour, using Linux operating system and in US East)

- *Scenario 2 (S2)*: It consists of class 2 SFCs, with $\frac{1}{3}$ of requests of each type defined in the Table III, considering 20000t, 49 SFCs are processed;
- *Scenario 3 (S3)*: It consists of class 3 SFCs, with $\frac{1}{4}$ of requests of each type defined in the Table III, considering 20000t, 40 SFCs are processed;
- *Scenario 4 (S4)*: It consists of the union of the three previous scenarios, *i.e.*, the SFCs arrive with different intervals and characteristics, considering 20000t, 122 SFCs are processed.

C. Evaluated Approaches

The following reconfiguration strategies were used to analyze the solution of the VNF-PC problem:

- ILP^i : The VNF instance (i) sizing constraints are reoptimized. Although this strategy generates an exact solution, the placement and chaining of active VNFs aren't reoptimized. This strategy cannot guarantee the global optimal solution;
- ILP^{ic} : The VNF instance (i) sizing and chaining (c) constraints of the active VNFs are reoptimized. Although this approach generates an exact solution, since the VNFs placements aren't reoptimized, the proposed approach cannot guarantee the global optimal solution;
- ILP^{icp} : The constraints of VNF instance (i) sizing, chaining (c), and VNF placement (p) of all SFCs $v \in V^{at}$ are reoptimized. In this case, the result is guaranteed to be a global optimal solution. As a result, no other re-optimization strategy can generate a higher profit in terms of the objective function value, and an upper bound is obtained.

D. Experimental Results

1) *Profit analysis*: Because the proposed model's objective function is to maximize profit, we begin these analyses with this metric (Fig. 2). A globally optimal solution is a feasible solution with an objective function value that is better (or as

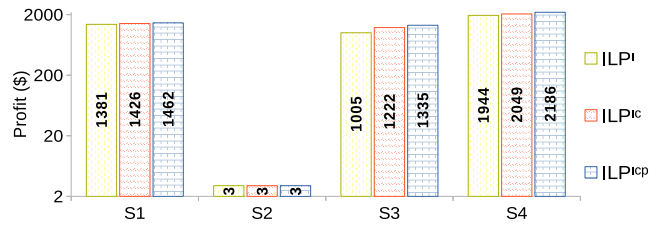


Fig. 2 Providers' Profit.

good) as all other feasible solutions. Since the ILP^{icp} approach guarantees the global optimal solution, such a value can be used to interpret the performance of the other approaches.

The acceptance rate has a direct impact on the revenue generated by reconfiguration approaches (Fig. 2). In this case, the more SFCs successfully mapped, the greater the revenue and, potentially, the greater the profit. Other factors, such as sharing SN servers and arcs-spread of each request, are also important for profit generation. It should be noted that the objective function gives priority to profit ahead of acceptance rate. Therefore, and in some cases, one can prioritize some SFC mapping with a placement on the SN that initially generates a higher profit, but that affects the mapping of other SFCs that may be processed later.

In Scenario 2, profit tends to be low for all approaches evaluated (Fig. 2). This occurs because such a scenario is characterized by SFCs with short durations. In case, a SFC with a short lifetime, generates little profit for the provider. Scenarios with longer duration requests, such as Scenario 4, with requests like online games and streaming audio or video, on the other hand, tend to generate higher profits.

The ILP^{icp} approach outperforms the other approaches in terms of profits. This is due to the fact that such an approach generates the highest revenues, a higher acceptance rate, and a more concise mapping with more shared SN servers. Because the path-spread and sharing SN servers are auxiliary metrics, they were not plotted. Consider comparing the profit percentage alteration between the ILP^i and ILP^{icp} approaches. In scenario 4, the ILP^i approach profits $\approx 11.07\%$ less, has a $\approx 5.23\%$ lower acceptance rate, in addition to a $\approx 99.97\%$ runtime reduction (Fig. 3). Despite the fact that the ILP^{icp} approach increased profits by 25.32% when compared to the other approaches (scenario S3), a significant runtime increase suggests that its use may be impractical.

In scenarios with fewer simultaneously active SFCs, the effect of reconfiguration between approaches is reduced. It is noted in the scenario 2 experiments. Due to the arrival interval in this scenario is greater than the SFC's lifetime, there are no concurrently active SFCs, and the effect of reconfiguration is null. With fewer active SFCs, there is less competition for resources, and less need for constraint re-optimization. Despite only reconfiguring the size of the assigned instances, the ILP^i approach generates a good profit ratio (Fig. 2), but with a lower final profit than the ILP^{ic} . As a result, while reconfiguring the routing does not significantly increase runtime, it can have a promising impact in some scenarios, such as scenario 3.

2) *Total runtime analysis*: Since there are fewer SN components to process, scenarios with fewer SFCs active simulta-

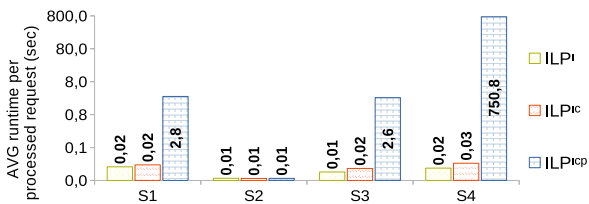


Fig. 3 Total runtime.

neously generate fewer variables and constraints to process in the mathematical model, implying a shorter runtime (Fig. 3). This idea is supported by the fact that in the Scenario 2, the runtime for all approaches is the same, due to there are no concurrent active requests. In another cases, when comparing Scenarios 1 and 4, the processing time increase is ≈ 747 seconds per SFC processed using the approach ILP^{icp} . This occurred due to the problem's complexity, in the present case, a linear increase in the number of components to be processed implies an exponential increase in the number of variables to be processed by the ILP model.

In scenario 2, where no SFCs are active simultaneously, all approaches produced the same acceptance rate, profit, and runtime. This result backs up the hypothesis that reconfiguration is only effective when multiple SFCs are active at the same time. The ILP^{icp} approach has a long runtime because it processes all constraints and variables from all active and incoming SFCs at the same time, taking approximately ≈ 25 hours to process the 122 requests and generate a global optimal solution (Fig. 3, Scenario 4). However, by re-optimizing only active VNF instances and/or chaining, the other approaches are faster than the ILP^{icp} approach, but at a lower profit (Fig. 2). In scenario 4, *e.g.*, the ILP^i approach has a $\approx 99.94\%$ shorter runtime than the ILP^{icp} approach but with $\approx 12\%$ lower profit.

3) *Acceptance Rate Analysis*: In all scenarios, all evaluated approaches have a consistent acceptance rate (Fig. 4). The experiments with scenario 2 have a high arrival rate and a short lifetime for each SFC. Therefore, there are no SFCs active simultaneously in such a scenario, no competition for SN resources, and the acceptance rate remains at 100%. When a new SFC arrives to be mapped in this case, several others have most likely terminated, restoring and making available the SN's resources previously reserved for serving new requests.

In scenarios 1 and 4 an acceptance rate of 87.88% and 96.97% respectively is observed with the ILP^i approach; 86.89% and 91.8% with the ILP^{icp} . According to this perception, both approaches can process the incoming load of such requests efficiently and without much loss, with a difference of $\approx 9.37\%$ and $\approx 5.36\%$ (Fig. 4). However, when the profit generated by these experiments is compared, the profit varies more than the acceptance ratio. In case, the ILP^{icp} approach outperforms the ILP^i by up to 25% (Scenario 3). As a result, reconfiguring the entire model is advantageous in terms of profit maximization. However, as previously demonstrated, this action significantly increases runtime.

Observing the percentage of mappings for each reconfiguration approach in scenario 4, it is observed that the analyzed

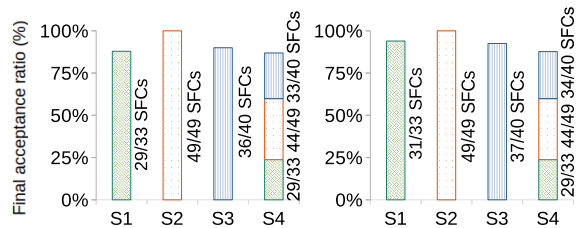
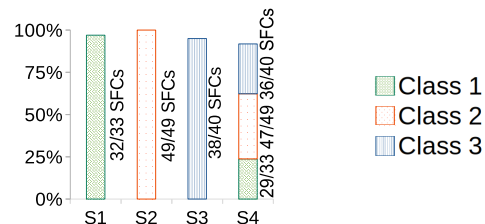
(a) Approach ILP^i (b) Approach ILP^{ic} (c) Approach ILP^{icp}

Fig. 4 Final acceptance ratio.

approaches generate unbiased behavior by not prioritizing the mapping of requests from one class over those from another. This reflects a policy of mapping neutrality. However, if necessary, the objective function can be changed to prioritize the mapping of requests from a specific class in order to increase profits.

VII. CONCLUSION

The VNF-PC problem solution has a direct impact on the internet service providers profit. Incrementally to current literature, different re-optimization strategies have been presented in this paper, implying a potential reallocation/migration of previously mapped components to meet a new SFC. In order to add value to the state of the art, this paper provided a detailed analysis of such strategies with NS found in the literature and widely used around the world. The main motivation for the exact approach presented in this paper is to generate good solutions and optimization bounds for the VNF-PC while respecting the QoS aspects of the clients, such as end-to-end delay, memory, processing, and bandwidth. Distinct types of NS were found in the literature and summarized in classes.

Computational experiments were carried out on these classes. Among the experiments with reconfiguration strategies, re-optimizing all the model's constraints is an extremely computationally expensive action. It is also observed that the approaches ILP^i and ILP^{ic} produce satisfactory results while taking less runtime. However, because such approaches do not re-placement of instantiated VNFs, they tend to generate lower profits. Finally, in real-world applications, the approach to be applied is determined by several factors related to the scenario at hand, in this case, no single approach can be chosen or ruled out. Once the classes of services have been defined, and different optimization bounds have been generated using the re-optimization approaches, as future work, we will use this theoretical reference in the validation of heuristic methods, which do not generate guarantees of satisfactory results.

ACKNOWLEDGEMENTS

This work is partially funded by CNPq, CAPES and FAPEMIG.

REFERENCES

- [1] B. Yi, X. Wang, K. Li, S. k. Das, and M. Huang, "A comprehensive survey of network function virtualization," *Computer Networks*, vol. 133, pp. 212–262, 2018.
- [2] K. Kaur, V. Mangat, and K. Kumar, "A review on virtualized infrastructure managers with management and orchestration features in nfv architecture," *Computer Networks*, vol. 217, p. 109281, 2022.
- [3] A. Laghrissi and T. Taleb, "A Survey on the Placement of Virtual Resources and Virtual Network Functions," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2018.
- [4] X. Zhu and H. Deng, "A security situation awareness approach for iot software chain based on markov game model," *Int. J. Interact. Multim. Artif. Intell.*, vol. 7, no. 5, p. 59, 2022.
- [5] M. Baga, T. Taleb, J. B. Bernabe, and A. Skarmeta, "Qos and resource-aware security orchestration and life cycle management," *IEEE Transactions on Mobile Computing*, vol. 21, no. 8, pp. 2978–2993, 2022.
- [6] A. Alleg, T. Ahmed, M. Mosbah, R. Riggio, and R. Boutaba, "Delay-aware vnf placement and chaining based on a flexible resource allocation approach," in *2017 13th International Conference on Network and Service Management (CNSM)*, pp. 1–7, Nov 2017.
- [7] G. Garcia-Aviles, C. Donato, M. Gramaglia, P. Serrano, and A. Banchs, "Acho: A framework for flexible re-orchestration of virtual network functions," *Computer Networks*, vol. 180, p. 107382, 2020.
- [8] ETSI, "Network Functions Virtualisation; Infrastructure; Network Domain," GS NFV-INF 005 V1.1.1. Industry Specification Group, 2014.
- [9] J. Ahamed, M. Kohli, K. Ahmad, M. Jamal, and B. B. Gupta, "Cdps-iot: Cardiovascular disease prediction system based on iot using machine learning," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. In Press, pp. 1–9, 09 2021.
- [10] R. Tahir, K. Cheng, B. Memon, and Q. Liu, "A diverse domain generative adversarial network for style transfer on face photographs," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, p. 5, 08 2022.
- [11] R. F. Mansour, C. Soto, R. Soto-Díaz, J. Escorcía-Gutiérrez, D. Gupta, and A. Khanna, "Design of integrated artificial intelligence techniques for video surveillance on iot enabled wireless multimedia sensor networks," *Int. J. Interact. Multim. Artif. Intell.*, vol. 7, no. 5, p. 14, 2022.
- [12] L. Askari, F. Musumeci, and M. Tornatore, "Latency-aware traffic grooming for dynamic service chaining in metro networks," in *IEEE International Conference on Communications (ICC)*, pp. 1–6, May 2019.
- [13] S. Padhy and J. Chou, "Finding the optimal reconfiguration for network function virtualization orchestration with time-varied workload," in *Proceedings of the 3rd International Workshop on Systems and Network Telemetry and Analytics, SNTA '20*, (New York, NY, USA), p. 49–52, Association for Computing Machinery, 2020.
- [14] L. Le, B. P. Lin, L. Tung, and D. Sinh, "Sdn/nfv, machine learning, and big data driven network slicing for 5g," in *2018 IEEE 5G World Forum (5GWF)*, pp. 20–25, July 2018.
- [15] J. Liu, W. Lu, F. Zhou, P. Lu, and Z. Zhu, "On dynamic service function chain deployment and readjustment," *IEEE Transactions on Network and Service Management*, vol. 14, pp. 543–553, Sep. 2017.
- [16] R. Mansour, C. Soto Montaña, R. Soto Diaz, J. Escorcía-Gutiérrez, D. Gupta, and A. Khanna, "Design of integrated artificial intelligence techniques for video surveillance on iot enabled wireless multimedia sensor networks," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, p. 14, 09 2022.
- [17] A. Fischer, D. Bhamare, and A. Kassler, "On the construction of optimal embedding problems for delay-sensitive service function chains," in *2019 28th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–10, July 2019.
- [18] S. Mehraghdam, M. Keller, and H. Karl, "Specifying and placing chains of virtual network functions," in *2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)*, pp. 7–13, Oct 2014.
- [19] A. S. Tanenbaum and D. Wetherall, *Computer networks, 5th Edition*. Pearson, 2011.
- [20] A. Korn, K. Nemeth, G. Feher, and I. Cselenyi, "Benchmarking Terminology for Resource Reservation Capable Routers." RFC 4883, 2007.
- [21] A. Abdelhamid, *Service Function Placement and Chaining in Network Function Virtualization Environments*. PhD thesis, Université de Bordeaux, 07 2019.
- [22] A. Hmaity, M. Savi, F. Musumeci, M. Tornatore, and A. Pattavina, "Protection strategies for virtual network functions placement and service chains provisioning," *Networks*, vol. 70, no. 4, pp. 373–387, 2017.
- [23] Deezer, "Deezer audio quality," 2021. [Online]; accessed 07/01/21, <https://support.deezer.com/hc/en-gb/articles/Deezer-Audio-Quality>.
- [24] A. Ouni, M. Kessentini, K. Inoue, and M. O. Cinné, "Search-based web service antipatterns detection," *IEEE Transactions on Services Computing*, vol. 10, no. 4, pp. 603–617, 2017.
- [25] Google, "Bandwidth, data usage, and stream quality: Google Support," 2021. [online] <https://support.google.com/stadia/answer/9607891?hl=en>.
- [26] J. Holub, M. Wallbaum, N. Smith, and H. Avetisyan, "Analysis of the dependency of call duration on the quality of voip calls," *IEEE Wireless Communications Letters*, vol. 7, no. 4, pp. 638–641, 2018.
- [27] Google, "System requirements," 2021. [Online]; 07/01/21, <https://support.google.com/youtube/answer/78358?hl=pt-BR>.
- [28] D. Bhamare, M. Samaka, A. Erbad, R. Jain, L. Gupta, and H. A. Chan, "Optimal virtual network function placement in multi-cloud service function chaining architecture," *Computer Communications*, vol. 102, pp. 1–16, 2017.
- [29] Y. Li and W. Tu, "Traffic modelling for iot networks: A survey," in *Proceedings of the 2020 10th International Conference on Information Communication and Management, ICICM 2020*, (New York, NY, USA), p. 4–9, Association for Computing Machinery, 2020.
- [30] S. M. Araújo, F. S. de Souza, and G. R. Mateus, "A hybrid optimization-machine learning approach for the vnf placement and chaining problem," *Computer Networks*, vol. 199, p. 108474, 2021.
- [31] Y. Jia, C. Wu, Z. Li, F. Le, and A. Liu, "Online scaling of nfv service chains across geo-distributed datacenters," *IEEE/ACM Transactions on Networking*, vol. 26, pp. 699–710, April 2018.



Samuel M. A. Araújo received the BS degree in Computer Science from Federal University of São João del-Rei, Brazil, in 2016 and MSc degree from Federal University of Minas Gerais, Brazil, in 2018. Currently, he is a PhD candidate at the Operational Research Laboratory (LaPO) at Federal University of Minas Gerais. His research topic includes virtualization, optimization, and simulation.



Fernanda S. H. de Souza is an Assistant Professor at the Department of Computer Science, Federal University of Ouro Preto, Brazil. She received her MSc and PhD in computer science from Federal University of Minas Gerais, Brazil, in 2007 and 2012, respectively. She spent 2010 at the Université de Montréal, Canada, during her PhD. Her research interests include mathematical programming, heuristics and combinatorial optimization.



Geraldo R. Mateus is Full Professor in Computer Science at Federal University of Minas Gerais, Belo Horizonte, Brazil. He received his PhD and MSc in computer science from Federal University of Rio de Janeiro, Brazil, in 1980 and 1986, respectively. He spent 1991 and 1992 at the University of Ottawa, Canada, as a visiting researcher. His research interests span network optimization, combinatorial optimization, algorithms, logistic, transport and telecommunication. He is a member of INFORMS, IFORS, SBC, SIAM and SOBRAPO. He has published over 250 scientific papers, 50 journal papers and book chapters and two books, and is a leader of several national and international projects. He has worked as a consultant for some companies such as Usiminas, CVRD, MBR, Telemig, Telemar, France Telecom, Embratel and for the Brazilian government.