Critical Function Placement based on Service Chains in Multi-administrative Federated Networks

Dariusz Nogalski*, Dallal Belabed*, Alexandre Triollet, Konstanty Junosza-Szaniawski, Slim Abdellatif, Pascal Berthou, Stanislas Pedebearn, and Adam Dudko

***Abstract*—Although the Service Function Chains (SFCs) embedding problem is broadly investigated in the literature, few works address it in a sliced multi-administrative network federation. In this work, we provide several insights into the problem. First, we describe a new federated-level topology abstraction. Second, we introduce a novel optimization model and heuristic (for large scale), which solve SFC embedding. Third, we conduct experiments on various multi-domain topologies and compare the algorithms regarding resource allocation efficiency and runtime. We analyze the trade-off between slice deployment costs and link utilization. Finally, we emulate two security scenarios on Containernet, Docker, and Open vSwitch architecture.**

***Index Terms*—SDN, NFV, multi-administrative multi-domain networks, service function chain embedding, slicing, QoS.**

I. Introduction

In recent years, due to the development of network systems based on Software Defined Networking (SDN) and Network Function Virtualization (NFV), computer networks do not rely only on transmission resources but also on computational resources. This is especially important in coalition networks; By taking advantage of new technologies SDN/NFV, various network operators (nations, in the context of military networks) can provide efficient and attack-resistant network services by sharing and complementing each other's network cyber-defense capabilities (Li and Wang [1]) provided in the form of network functions. This idea goes in the direction of building cooperative defenses for network protection.

To define attack-resistant end-to-end (E2E) services, we need additional security functions (also called critical functions in this paper) that perform threat detection and mitigation to prepare the network against cyber-attacks (Pióro et al. [2], Junosza-Szaniawski and Nogalski [3]). Such critical protection and mitigation security functions could be, e.g., Firewall (FW), Intrusion Detection System (IDS), Distributed Denial of Service (DDoS) attack sensor (Junosza-Szaniawski and Nogalski [4] [5], Blazek et al. [6]), traffic scrubber (Fayaz et al. [7]), DDoS mitigation (Belabed et al. [8]), encryption/decryption, traffic filtering, etc. In addition, there are other crucial functions of network protection (of Quality of Service (QoS) nature), e.g., compression, decompression, traffic shaping, traffic optimizer, and load balancer (Beck and Botero [9]).

A resilient-to-attack network service can then be represented as a Service Function Chain (SFC) composed of a set of Virtual Network Functions (VNFs) and the directional virtual links that connect them (the flow of packets passes through the chain of VNFs that constitute the SFC). In a multi-domain network, NFV can be deployed and hosted by different domains.

This paper addresses the embedding of service chains in a multi-domain context. When embedding a multi-domain SFC in a multi-administrative domain federation, the initiating domain relies on the compact views of each domain's network topology (the abstract topologies) to form an E2E SFC. Typically, domains limit the disclosure of topology information (Toumi et al. [10]). Classically, the abstract topology shared with other domains consists of border nodes and abstract links connecting them. Some domains may also share compute nodes with a set of VNFs they can host. Our work also proposes sharing other elements, e.g., domain-level slices and non-border transit nodes.

An example of SFC embedding at the federated level within a two-domain network is presented in the Fig. 1. Domain 1 exposes two data centers (each exposes two VNFs: FW, IDS). Domain 2 exposes one data center (VNFs: FW, IDS). Both domains expose edge nodes (green) and non-edge transit nodes (black). Some links represent domain-level slices (dotted). We consider the embedding of a single user demand in the form of E2E SFC. The user demand (blue) requires a chain of two VNFs (FW, IDS). For simplicity, bandwidth and latency characteristics are omitted here. More details on the federated view of the network graph are described in Section IV.

This work comes as an extension of previously presented work (Nogalski et al. [11]), in which we introduced the ILP model and heuristic to solve the problem. The contribution of

Manuscript received December 19, 2024; revised January 6, 2025. Date of publication February 3, 2025. Date of current version February 3, 2025. The associate editor prof. Claudia Canali has been coordinating the review of this manuscript and approved it for publication.

D. Nogalski and A. Dudko are with the Military Communications Institute - National Research Institute, Zegrze, Poland (dariusz.nogalski@wil.waw.pl, adam.dudko@wil.waw.pl). D. Belabed and A. Triollet are with the Airbus Defence and Space, Research and Development department, Issy-les-Moulineaux, France (dallal.belabed@airbus.com, alexandre.triollet@fr.airbus.com). K. Junosza-Szaniawski is with the Warsaw University of Technology, Faculty of Mathematics and Information Science, Warsaw, Poland (konstanty.szaniawski@pw.edu.pl). S. Abdellatif is with the LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France (slim@laas.fr). P. Berthou and S. Pedebearn are with the LAAS-CNRS, Université de Toulouse, CNRS, UPS, Toulouse, France (berthou@laas.fr, spedebearn@laas.fr).

The paper was presented in part at the International Conference on Software, Telecommunications and Computer Networks (SoftCOM) 2024.

Digital Object Identifier (DOI): 10.24138/jcomss-2024-0117

∗ These two authors contributed equally to this work

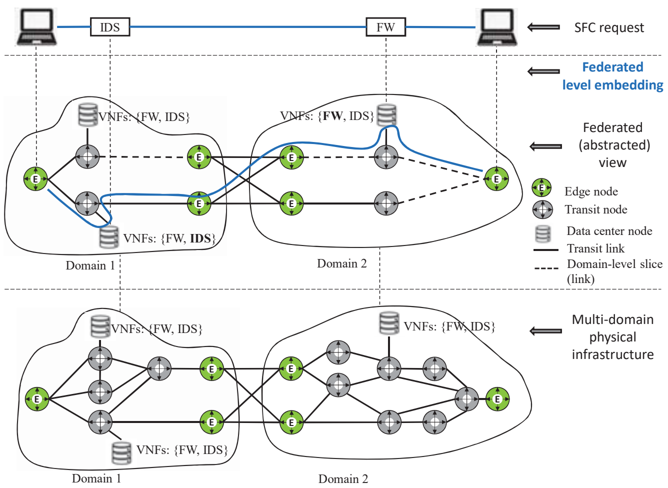


Fig. 1. SFC embedding in sliced multi-administrative multi-domain network (federated-level view)

this work with respect to the previously published work is as follows:

- Extensive state-of-the-art analysis concerning the SFC embedding problem.
- Introduction of a new federation abstraction structure that extends the classical domain abstraction (focusing solely on network resources) by proposing an abstraction model for the computational resources that domains expose and make available to other domains to run their VNFs (or SFCs).
- Extension of the experimentation with new figures to analyze slice deployment cost and link utilization in order to validate the ILP model and heuristics.
- Emulation results of the algorithm (ILP model) implemented as part of the proof of concept prototype with the use of the Containernet platform. Two new security scenarios (IDS placement) under different objective functions were verified. First, minimizing slice deployment, and second, minimizing link utility.

We assume that our approach will be beneficial in coalition networks. In federated military networks, allied nations connect and share part of their network infrastructure to build a Federation Mission Network (FMN) [12]. The nations can complement each other's capabilities to build comprehensive network services. The multi-domain coordinator (there may be more than one) acquires advertised resources and coordinates the E2E service configuration/modification (scaling up/down). However, the management of physical resources at the national level is the responsibility of each nation.

The rest of the paper is organized as follows. Section II discusses the related work. Section III overviews SFC embedding in a multi-administrative federated network. Section IV discusses the proposed domain abstraction. Section V introduces the mathematical formulation. Section VI presents the experiments and results. Finally, Section VII presents the conclusions and future work.

II. Related Work

A. *SFC in Single-Domain Context*

In the literature, multiple existing works are applied in the single-domain context. Addis et al. [13] formulate via mathematical programming the VNF Placement and Routing (VNF-PR) optimization problem, including compression constraints. Each demand requests a subset of VNFs; the order of network functions is not required (as opposed to this work). The formulation minimizes the maximum network link utilization (Traffic Engineering (TE) goal) and minimizes the number of cores (Central Processing Unit (CPU)) used by the instantiated VNFs. The two competing goals are prioritized.

In Wion et al. [14], the SFC routing problem is formulated as the Integer Linear Program (ILP) model. The goal is to minimize the service function cost (proportional to the requests' bandwidth) and network link cost (static, proportional to the used bandwidth). The order of network functions is not required (as in this work). The bitrate of each demand flow doesn't change along a VNF chain (as in this work).

Peng and Di [15] maximize the compute resource utilization efficiency by jointly optimizing the VNF deployment, power, and spectrum resource allocation.

Yang et al. [16] address the risk of network attacks and allocate SFC based on honeypots and backup technology to reduce the resource cost of protecting air traffic information networks while enhancing network security. They deploy SFC VNFs close to the shortest path between the source and destination endpoints, aiming to reduce SFC latency and save bandwidth.

Murray et al. [17] presents a VNF placement and routing algorithm based on a column generation method that iterates between generating improved paths and optimizing VNF placement based on the generated paths. They optimize throughput, latency, and availability in a multi-layer radio access network (RAN).

Ko et al. [18] assume a network of service nodes (SN), where each service node is exposing service functions (SFs) by means of network function virtualization (NFV). They propose an integer non-linear model that considers link latency and SN resources (e.g., CPU capacity, storage capacity, and memory capacity). The goal is to optimize latency. For example, the chain for latency-sensitive services such as multimedia streaming and voice-over IP (VoIP) has a tight latency requirement, while the chain for the file download service has a loose latency requirement.

Popokh et al. [19] address efficient resource allocation for VNF placement while minimizing the communications latencies between VMs that are part of the VNF deployment.

A related problem to the SFC is virtual network embedding. Embedding virtual networks in a substrate network (SN) is the main resource allocation challenge in network virtualization and is usually referred to as the *Virtual Network Embedding problem (VNE)* (Herrera et al. [20], Belbekkouche et al. [21]). VNE concentrates on the allocation of virtual resources both in nodes (mapping to substrate nodes and their compute resources) and links (mapping to substrate network resources - links/paths) (Botero et al. [22]). Computing optimal VNE is an NP-hard optimization problem (Herrera et al. [20], Fischer et al. [23]). Even in the case when we map only virtual links (without information about node resources, e.g., VNFs, CPUs, etc.) to the substrate network links, in the single-path setting (i.e., the flow of a user in the SN follows a single path), the problem is NP-complete. Such a problem reduces to the decision version of the m-commodity flow problem with fixed rates and a single-path setting which is NP-complete (Drwal [24], Junosza-Szaniawski and Nogalski [25]). This is proved by the reduction from the decision version of the bin-packing problem. In Even et al. [26], it is shown that even the decision version of the two-commodity integral flow problem is NP-complete by reduction from the boolean satisfiability problem (SAT) problem.

B. SFC in Large Scale Context

In the literature, multiple existing works have been applied in the large-scale network context. Following Beck and Botero [9], resource allocation algorithms for setting up virtual network services should scale with the size of the substrate network. A solution should be found in the range of minutes or seconds – even in larger scenarios [9].

Tastevin et al. [27] propose the ILP formulation for SFC. Each network node is a Point of Presence (PoP) and can possibly host VNFs. It also has a limited CPU capacity, thus, multiple PoPs should be needed to serve all traffic demands. The work minimizes Operating Expenses (OPEX) composed of two components: i) the VNF deployment cost, which is directly linked to the number of PoPs hosting VNF instances, and ii) the cost of forwarding traffic, which is linked to the number of hops in a traffic request path and its bandwidth usage. The longer the SFC path is, the more it will cost. SFC request is an ordered logical sequence of VNFs. They also propose a graph-based heuristic that combines graph centrality and multi-stage graphs.

Beck and Botero [9] propose an NFV resource allocation problem (NFV-RA) that is divided into two problem stages: 1) service chain composition to find VNF-FGs (VNF Forwarding Graph, a directed acyclic graph representing the request) to be embedded in the substrate network and 2) service chain embedding (VNF-FG embedding in the substrate network). They propose a heuristic method to solve the composition of VNF chains and their embedding into the substrate network in one coordinated step.

Obadia et al. [28] present ILP formulation of SFC placement problem. Since the problem is NP-complete, they provide a heuristic based on game theory and implement the best-response algorithm. They assume different NFV operating costs: a) VNF license cost (operating cost) (some VNF can have a license cost that depends on whether the VNF is running or not); b) energy cost (operating cost) of having a server with the VNF software running in idle mode; c) processing cost, a piece-wise linear function of the load; d) link cost.

Our work addresses the SFC problem on a large scale by proposing efficient heuristic for large network instances. However, our formulation is different from that of the above-cited works.

C. SFC in Multi-Domain Context

In multi-domain, the literature presents various studies to address slice embedding, particularly within the 5G/6G (Addad et al. [29]). The centralized orchestration requires a global view of the infrastructure, which raises scalability concerns. The desire to reduce communication costs has led to the development of distributed slice embedding solutions. Other related works on multi-domain SFC address, e.g., reduction of deployment time (El Amine et al. [30]), dynamic orchestration (Wu and Zhou [31]), and SFC placement with limited visibility (Toumi et al. [10]).

Our work addresses the multi-domain slice embedding by proposing an ILP model that solves VNF and Link embedding simultaneously (in existing works, they are usually solved separately).

III. SFC Embedding in a Multi-administrative Federated Network

When embedding an end-to-end SFC in a multi-domain network context, the initiating domain may solicit multiple domains to support different portions of the SFC. To do this, it needs to collect information from other domains, build, and then maintain a compact view of the network topology of each domain with the available resources. In a multi-administrative multi-domain context, domains are reluctant to disclose their detailed topology information. In fact, a synthetic abstraction of their topology and their computing resources are disclosed to other domains. Each domain is sovereign by the abstraction that it exposes to the others. The abstraction policy, which specifies how to derive the abstracted infrastructure from the real infrastructure, is specific to each domain. Moreover, at the time of service embedding, a domain can decline the grant of disclosed resources to a requesting domain. This is shown in Fig. 2, which describes the main stages that Resource Allocators (RA) must go through to set up multi-domain end-to-end SFCs. Notably, it highlights and positions the federated-level resource allocation (which is the focus of this paper) with respect to the domain-level resource allocation.

Below, we describe the domain abstraction that we propose for embedding a multi-domain SFC atop a multi-administrative federated network.

IV. Proposed Domain Abstraction

We distinguish two aspects in the abstraction exposed by a domain. First, the computing abstraction abstracts the computing resources (used for running VNFs) that a domain holds and offers to other domains. Second, the network topology abstraction abstracts how exposed nodes are connected to each other. These are successively detailed hereafter. The computing abstraction proposed in this paper extends our previous work (Pedebearn et al. [32]), which focused on network topology abstractions.

A. Computing Abstraction

The computing abstraction is represented by the concept of an abstract compute node, which collects a pool of computing

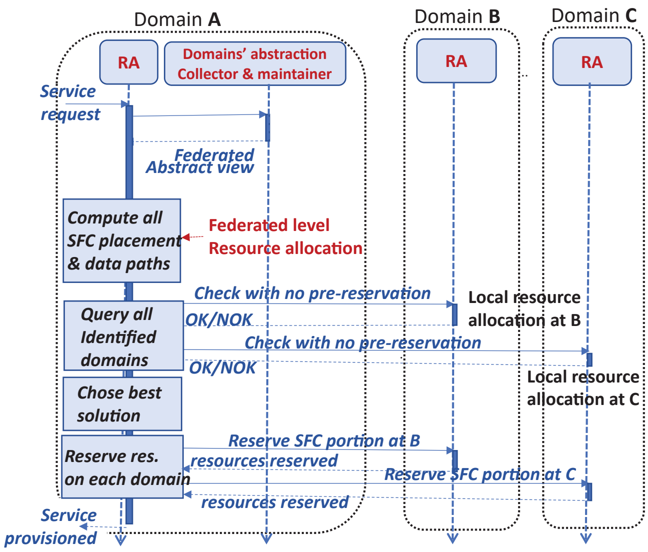


Fig. 2. Multi-domain SFC embedding stages

resources and a list of VNFs that can be supported by them. It may represent a single physical server, a group of servers, an aggregated collection of servers, or data centers spread across the domain.

An exposed compute node is characterized by the following main attributes:

- Maximum and available capacity in terms of compute units. A compute unit represents a bundle of processing power, memory, storage, etc., corresponding to a unit of measure used to specify the computing resources needed by a VNF instance.
- List of supported VNFs. In turn, the following information is also provided for each supported VNF.
  - The resource cost (i.e. amount of resources) in compute units per $Mbps$.
  - The maximum allowed/supported flow rate in $Mbps$.
  - Maximum processing delay in ms.
  - Average flow compression rate, when applicable.
  - Provisioning time specifies the time needed to instantiate the VNF in the exposed compute node.
  - Additional information could also be considered to reflect some cost for deploying the VNF or to be used to enable some policy-based decisions.
- Location, when appropriate.

B. Network Topology Abstraction

It aims to specify how exposed network nodes and abstracted compute nodes are connected. Exposed network nodes necessarily include edge nodes (the entry and exit points of the domain). In addition, our topology abstraction allows the disclosure of non-edge network nodes (which are typically abstract and can be used to capture some topological constructs of the domain) as well as network slices supported and made available by the domain to the federation. Both bring valuable benefits in terms of efficient resource usage (Pedebearn et al. [32]). For space reasons, the slice abstraction is not described. An exposed network node may be characterized by various attributes: the maximum packet forwarding rate, the flow table size, etc. If so, it inherits the compute nodes' related attributes with the list of VNFs that it supports.

Exposed links are typically unidirectional abstract links and correspond to one or multiple data paths (with multiple physical hops). They are characterized by the following main attributes:

- Source-node and end-node.
- Max & available capacity: corresponds to the maximum and available bandwidth of the link.
- Service Level Agreement (SLA): which specifies the expected performance in terms of maximum/average transfer delay, maximum/average packet loss rate, maximum jitter, and more.
- Links may also be labeled with cost information or any other information that can be used to enforce some policy-based decisions, e.g. confidence/security level, etc.

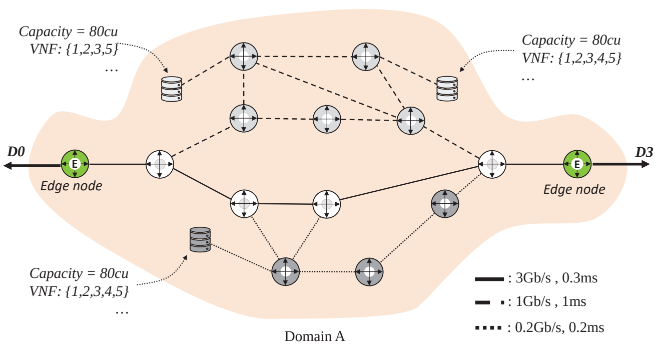


Fig. 3. Domain A physical network infrastructure.

To illustrate the proposed abstraction, let's consider the infrastructure of domain A as depicted in Fig. 3. Three data centers (pool of resources) are spread in three different locations connected to each other thanks to the network infrastructure composed of a dozen switches connected via bidirectional physical links. We assume the following abstraction policy for domain A:

- Bottom pool of resources is hidden to other domains.
- The two top pools of resources are aggregated and only half of the capacity of the left-top pool is exposed.
- Domains are only allowed to deploy VNFs $1, 3$ and 5, with the condition that VNF 5 can only be provisioned on the left-top pool.
- The top part of the network topology is used to support traffic destined to the resource pools, while the bottom part is used to support the traffic that flows between edge nodes.

One possible resulting domain abstraction is described in Fig. 4. One aggregate compute node is exposed to the federation with a capacity of $120cu$ (compute unit). VNFs 1, 3, and 5 can be deployed on behalf of any other domain. The maximum flow rate of VNF 5 is affected by the fact that it can only be deployed on the top-left pool of resources.

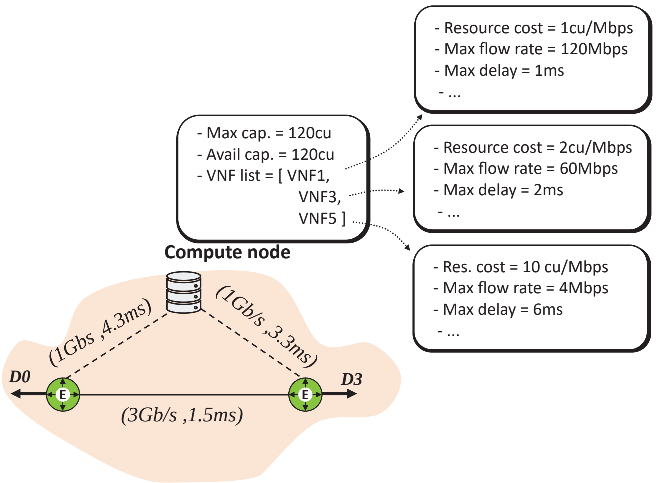


Fig. 4. Example of Domain A abstraction

Domain abstractions are collected by each domain and then used as input to build their own view of the multi-domain network, which is then used as input for their resource allocation algorithm. Next, we propose such an algorithm.

V. Mathematical Formulation

A. Notation

The notation is described in Table I. The notation describes parameters and variables that define the network and demand model. This is a general notation, common for both the ILP model (Section V-B) and greedy heuristic (Section V-C).

We model the federated-level abstract network (abstracted view) for the purpose of federated-level SFC embedding as a directed graph $G = (N, A)$, where N is a set of nodes, and A is a set of directed links. A node can be an edge node or a non-edge transit node. A data center (hosting VNFs) can be co-located with either of them. A link $a \in A$ can be a slice link $a \in A_s$ (it represents a domain-level slice) or a classical transit link. An example federated level network graph is shown in the middle part of Fig. 1.

The central part of the demand model is the definition of the service chains (see Table I). The tuple $(f^{k,1}, .., f^{k,T_k})$ (list of ordered services) is defined for each k-th demand $k \in D$. Each E2E demand is represented by a single SFC. The user may define more than one demand, e.g., a separate one for data and control plane purposes.

Additionally to Table I, since for each $f \in F$ we might have an individual compression factor μ_f we define b_k^p as follows: for $p = 0$, $b_k^p = b_k$; for $p \in \{1, \ldots, T_k\}$, $b_k^p = b_k \cdot \prod_{q=1}^{p} \mu_{f^{k,q}}$. The b_k^p denotes bandwidth used by demand $k \in D$ after the first p services $f^{k,1}, .., f^{k,p}$ are performed.

Concerning the proposed domain abstraction (Section IV), we assume without loss of generality, the nodes that host the VNF have enough resources to support all VNF requests.

TABLE I
Notation

Parameters	
N	Set of nodes
A	Set of links (arcs)
$A_s \subseteq A$	Subset of links (arcs) which are slice-able
$\gamma_{i,j}^{max}$	Maximum capacity of the arc (i, j)
$\gamma_{i,j}$	Available capacity of the arc $(i, j) \in A$. 'Available' since we consider iterative allocations (for heuristic purpose)
$l_{i,j}$	Latency of the link $(i, j) \in A$, which is an upper bound that includes the transmission delay and the propagation delay
F	Set of service types (VNF types)
D	Set of demands
o_k, t_k	$o_k, t_k \in N$ origin and target of k-th demand $k \in D$
b_k, L_k	Bandwidth and max latency of k-th demand $k \in D$
$f^{k,p}$	$f^{k,p} \in F$ is p-th service step of k-th demand $k \in D$
T_k	For k-th demand it is the length of its service chain $(f^{k,1}, .., f^{k,T_k})$
μ_f	Compression/decompression factor for service $f \in F$, where $\mu_f = 1$ no compression/decompression; $\mu_f \leq 1$ compression VNF; $\mu_f \geq 1$ decompression VNF
N_f	$N_f \subset N$ set of nodes which provide a service (VNF) of type $f \in F$
N_F	$N_F = \bigcup_{f \in F} N_f$ set of nodes which provide a service (VNF) of any type
b_k^p	Bandwidth used by demand $k \in D$ after the first p services $f^{k,1}, .., f^{k,p}$ are performed
Decision Variables	
$\phi_{i,j}^{k,p}$	Continuous variable, represents the flow on the arc $(i, j) \in A$ of demand $k \in D$ for the p-th service step. The p-th service step is the state of demand processing after p-th service $(f^{k,p})$ is performed and before $p+1$-th service $(f^{k,p+1})$ is performed. For the initial service step $p = 0$, the flow $\phi_{i,j}^{k,0}$ denotes state before the first service $(f^{k,1})$ is performed
$x_{i,j}^{k,p}$	Binary $(x_{i,j}^{k,p} \in \{0, 1\})$, $x_{i,j}^{k,p} = 1$ iff $\phi_{i,j}^{k,p} > 0$ (the arc $(i, j) \in A$ is used by the flow $\phi_{i,j}^{k,p}$)
$e_{i,j}$	Binary $(e_{i,j} \in \{0, 1\})$, $e_{i,j} = 1$ iff the arc $(i, j) \in A$ is used by at least one flow $\phi_{i,j}^{k,p}$ for $k \in D, p \in \{0, \ldots, T_k\}$ $(e_{i,j} = \max\{x_{i,j}^{k,p} : k \in D, p \in \{0, \ldots, T_k\}\})$
$z_i^{k,p}$	Binary $(z_i^{k,p} \in \{0, 1\})$, $z_i^{k,p} = 1$ iff for demand $k \in D$ the p-th service $(f^{k,p} \in F)$ is executed at node $i \in N_{f^{k,p}}$
U	Continuous variable, represents maximum link utilization rate

B. The ILP Model

The model takes two types of federated-level parameters as input. First, the network model: topology, bandwidth, latency, VNF types, and locations (node). Second, the demand model: bitrate, delay, and SFC requirement (Table I). A detailed description of federated-level abstraction can be found in our previous work [32]. The optimization problem is to find:

- the optimal domain cost of the slice deployment (by minimizing the number of deployed national slices)
- the optimal link utilization (by minimizing maximum link utilization - to increase future demand admissibility)

subject to the following constraints:

1) Flow conservation laws are defined via the relationship between ϕ and z variables. We distinguish the following main cases: $p = 0$, $p = T_k$, and otherwise.
   For $p = 0$ (unprocessed flow of demand $k \in D$), we have two sub-cases. First, if $i = o_k$ (1a), then the flow balance (outgoing flow value - incoming flow value) equals b_k^0, which is b_k. Second, if $i = N - \{o_k\}$ (1b), we have two states: either $z_i^{k,1} = 0$ ($f^{k,1}$ service is not placed at i) then the flow balance is 0 or $z_i^{k,1} = 1$ then flow $\phi^{k,0}$ terminates in the node i and thus the flow balance is $-b_k^0$.

For $p = T_k$ (flow of demand $k \in D$ processed by service f^{k,T_k}) we have two sub-cases. First, if $i = t_k$ (1e), then the flow balance equals $-b_k^{T_k}$. Second, if $i = N - \{t_k\}$ (1d), we have two states: either $z_i^{k,T_k} = 0$ (f^{k,T_k} service is not placed at i) then the flow balance is 0 or $z_i^{k,T_k} = 1$ then the flow ϕ^{k,T_k} starts at the node i and thus the flow balance is $b_k^{T_k}$.
Otherwise, for $p > 0$ and $p < T_k$ (1c) we have three states. If $z_i^{k,p} = 0$ and $z_i^{k,p+1} = 0$, then the flow balance is 0. If $z_i^{k,p} = 1$ and $z_i^{k,p+1} = 0$, then the flow $\phi^{k,p}$ starts at the node i and thus the flow balance is b_k^p. If $z_i^{k,p} = 0$ and $z_i^{k,p+1} = 1$, then the flow $\phi^{k,p}$ terminates at the node i and thus the flow balance is $-b_k^p$.

$$\forall k \in D \; \forall i \in N \; \forall p \in \{0,..,T_k\}$$
$$\sum_{(i,j)\in A} \phi_{i,j}^{k,p} - \sum_{(j,i)\in A} \phi_{j,i}^{k,p}$$
$$= \begin{cases} b_k^0 & p = 0, i = o_k \quad (1a) \\ -b_k^0 \cdot z_i^{k,1} & p = 0, i \in N - \{o_k\} \quad (1b) \\ b_k^p \cdot z_i^{k,p} - b_k^p \cdot z_i^{k,p+1} & p \in \{1,..,T_k - 1\}, i \in N \quad (1c) \\ b_k^{T_k} \cdot z_i^{k,T_k} & p = T_k, i \in N - \{t_k\} \quad (1d) \\ -b_k^{T_k} & p = T_k, i = t_k \quad (1e) \end{cases}$$

2) For non-VNF nodes $i \in N \setminus N_F$ variable z equals zero

$$\forall k \in D \; \forall p \in \{1,..,T_k\} \quad z_i^{k,p} = 0 \tag{2}$$

3) For demand $k \in D$, the p-th service ($f^{k,p}$) is performed by at most one service available at some node $i \in N_{f^{k,p}}$

$$\forall k \in D \; \forall p \in \{1,..,T_k\} \sum_{i \in N_{f^{k,p}}} z_i^{k,p} = 1 \tag{3}$$

4) There is flow only on used edges - the connection between ϕ and x

$$\forall k \in D \; \forall (i,j) \in A \; \forall p \in \{0,..,T_k\} \; \phi_{i,j}^{k,p} = b_k^p \cdot x_{i,j}^{k,p} \tag{4}$$

5) Maximum latency has two components: delay on transport link A, and delay caused by VNF processing (the latter can be easily included)

$$\forall k \in D \sum_{(i,j)\in A} \sum_{p\in\{0,..,T_k\}} l_{i,j} \cdot x_{i,j}^{k,p} \le L_k \tag{5}$$

6) If the slice arc $(i,j) \in A_s$ is used by at least one demand $k \in D$, then the slice is enabled

$$\forall k \in D \; \forall p \in \{0,..,T_k\} \; \forall (i,j) \in A_s \quad x_{i,j}^{k,p} \le e_{i,j} \tag{6}$$

7) If the slice arc $(i,j) \in A_s$ is not used by any demand $k \in D$, then the slice is disabled

$$\forall (i,j) \in A_s \sum_{k\in D} \sum_{p\in\{0,..,T_k\}} x_{i,j}^{k,p} \ge e_{i,j} \tag{7}$$

8) The sum of flows does not exceed the edge capacity. For online version of the algorithm (also for Section V-C)

$$\forall (i,j) \in A,$$
$$1 - \frac{1}{\gamma_{i,j}^{max}} \cdot \left(\gamma_{i,j} - \sum_{k\in D} \sum_{p\in\{0,..,T_k\}} \phi_{i,j}^{k,p} \right) \le U$$

We consider two objective functions:

- Traffic Engineering (TE) goal: minimize the maximum network link utilization (e.g., to increase future demand admissibility):
$$\min U \tag{8}$$
- Slice Deployment (SD) goal: minimize the number of used slices (e.g., to reduce slice setup time):
$$\min \sum_{(i,j)\in A_s} e_{i,j} \cdot \frac{1}{|A_S|} \tag{9}$$
We define S as equal to $\sum_{(i,j)\in A_s} e_{i,j} \cdot \frac{1}{|A_S|}$.
Thus, depending on the federated operator need, the objective is to balance between TE and SD goals by adjusting α, where $\alpha \in [0,1]$ is a parameter describing the importance of TE goal over SD goal

$$\min \alpha \; U + (1-\alpha) \; S \tag{10}$$

We define G as equal to $\alpha \; U + (1-\alpha) \; S$.

Fig.11 in Section VI-C gives two verification scenarios of slice embedding, taking into consideration the two goals separately, TE (Fig. 11a) and SD (Fig. 11b), respectively.

C. The Greedy Heuristic

The heuristic is given as Algorithm 1. Firstly, the demand set is sorted by latency (could also be sorted by bandwidth if needed) (line: 3). Secondly, in each iteration (line: 5-10) the subset of d demands $D_d \in D$ is selected (line: 5), processed by model (line: 6), embedded (line: 7-8), and demands marked as processed (line: 9).

VI. Performance Evaluation

The model (Section V-B) was implemented in Optimization Programming Language (IBM CPLEX v22.1), and the heuristic (Section V-C) was implemented in Python v3.10.

A. Evaluation1 - Simulation on Cost266 network

For this experiment, we defined a multi-domain topology based on the *Cost266* topology [33], by assigning nodes (cities) to domains (countries) (Fig. 5).

1) Result of the ILP Model: We conducted the tests of the model with the following configurations (4x2):

- 4 topology configurations. Each topology configuration was generated with a randomly assigned maximum link capacity and latency. Links in the red dotted area (Fig. 5) received a capacity in the range of 5-10 Gbit/s. Links in the orange dotted area 2-5 Gbit/s, and peripheral links 1-2 Gbit/s. All links received latency in the range of 3-10 ms.

Algorithm 1 SFC Greedy Heuristic

INPUT: $G(N, A)$; A_s; for each link $(i, j) \in A$ capacity $\gamma_{i,j}^{max}$, $\gamma_{i,j}$, latency $l_{i,j}$; VNF advertisements F, N_f; demands $k \in D$; d - number of demands processed in a single algo iteration
OUTPUT: result ϕ (list of embedded slices (path definitions)), in other words $|D|$ E2E slices (paths) one for each requested demand $k \in D$, defined via $\phi_{i,j}^{k,p}$ for each $k \in D$, $(i, j) \in A$ and $p \in \{0, .., T_k\}$
Sort demands D by latency (from low latency to high latency (latency insensitive))
while $D \neq \emptyset$ **do**
    Select d demands $D_d \subseteq D$ (if $d > |D|$, select remaining)
    Run SFC ILP Model (sec V-B) with the input $(N, A, A_s, \gamma_{i,j}^{max}, \gamma_{i,j}, l_{i,j}, F, N_f, D_d)$ to get $\phi_{i,j}^{k,p}$ for each $k \in D_d$, $(i, j) \in A$ and $p \in \{0, .., T_k\}$
    Add $\phi_{i,j}^{k,p}$ assignments to the result ϕ (list of embedded paths)
    Update available capacity $\forall (i, j) \in A$, $\gamma_{i,j} = \gamma_{i,j} - \sum_{k \in D_d} \sum_{p \in \{0,..,T_k\}} \phi_{i,j}^{k,p}$
    $D = D \setminus D_d$
    return ϕ
end while

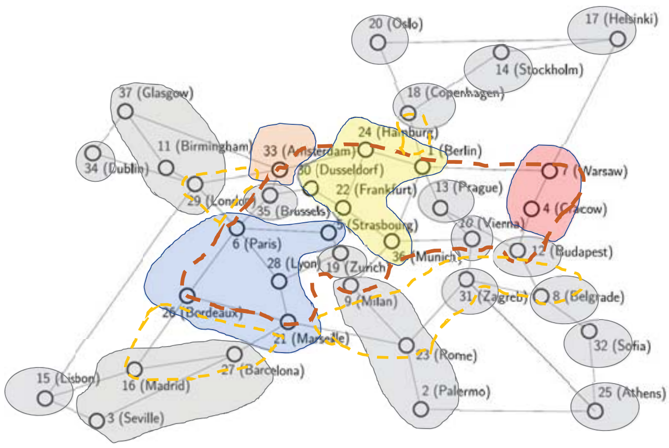


Fig. 5. Evaluation1 Multi-domain topology based on *Cost266* [33]

- 2 data center (DC) configurations. Each DC configuration contained seven randomly selected DCs - two located in France, three in Germany, one in Poland, and one in the Netherlands. For each DC, we randomly assigned two VNFs out of four from the VNF catalog = {FW, IDS, Deep Packet Inspection (DPI), Traffic Storage}).

We conducted three series of measurements, each corresponding to a fixed number of user demands {4, 6, 8}. Demand sets were generated incrementally. To the demand set of four demands, we added two demands (6) and again two demands (8). A single user demand was defined in the following way:

- a random start and end (in a different domain)
- a random bitrate in the range of 100-400 Mbit/s
- a random SFC (two ordered VNFs selected from the VNF catalog)
- a random maximum latency. In total, 25% of all demands in a demand set (D) were low latency (100 ms), and others latency insensitive.

For each series, we assumed the following values $\alpha = \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ (eq. 10). For fixed α, we conducted eight measurements (4x2) by changing topology and DC configuration and computed the average value of the goal (G), link utilization (U), slice deployment (S), and execution time (T). These average values were taken to analyze the experiment results (Fig. 6).

As one can observe, when the number of demands goes up, the combined goal value (G) (eq. 10) goes up (or stays at the same level) for any value of α (Fig. 6a). This shows the expected behavior of the model since, as we embed more demands, either more links are used (e.g. $\alpha = \{0, .., 0.8\}$ (Fig. 6c)), or the maximum link utility might be increased (but will never decrease) (e.g. $\alpha = \{0.2\}$ (Fig. 6b)).

Additionally, the average U and S (eq. 8 and 9) values behave as expected for any demand series (Table II, Fig. 6b-c). For any number of demands: first, when α increases, the U decreases (as we increase the importance of the link utility minimization factor) (Fig. 6b); second, when α decreases, the S decreases (as we increase the importance of the slice deployment cost minimization factor) (Fig. 6c).

Since the problem is NP-Hard, we show the model significant computation times (Fig. 6d, up to 1730 s for only 8 demands). This is the reason why we designed the heuristic.

TABLE II
EVALUATION1 COST266: THE RESULTS OF THE MODEL FOR DEMAND SET $\{4, 6, 8\}$ - U AND S

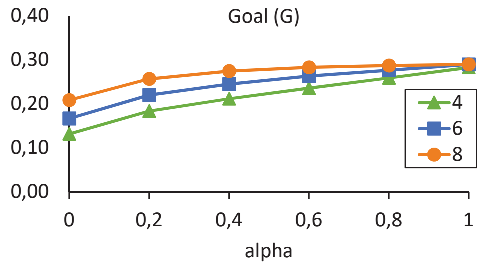
$|D| \setminus \alpha$	0	0,2	0,4	0,6	0,8	1	U or S
4	-	0,303	0,293	0,282	0,282	0,282	U
6	-	0,389	0,309	0,294	0,290	0,290	U
8	-	0,406	0,310	0,297	0,290	0,290	U
4	0,132	0,154	0,157	0,166	0,166	-	S
6	0,167	0,178	0,202	0,216	0,225	-	S
8	0,208	0,219	0,250	0,262	0,274	-	S

2) Result of the Heuristic: We conducted tests of the heuristic with the same 4 topology configurations and 2 DC configurations as in the previous subsection (same 4x2 configurations as for the testing of the ILP model). This time we conducted four series of measurements, each corresponding to a fixed number of user demands {10, 25, 45, 60}. Demand sets were generated incrementally. The heuristic was run with step $d = 3$.

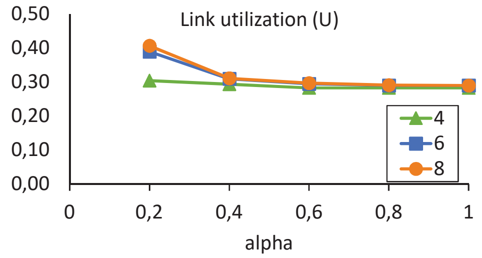
As one can observe (Fig. 7a), the average G values behave as expected; for any α, they increase as we increase the number of demands. The execution time is much lower compared to the one measured for the ILP model. A single observed average max computation time is up to 35 s (for 60 demands, Fig. 7d).

B. Evaluation2 - Simulation on NSFNET

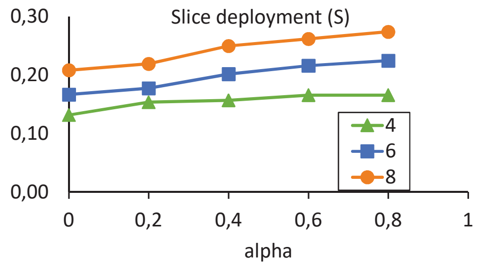
We proved experimentally that the ILP model (Section V-B) is more efficient than the heuristic (Section V-C). First, we transformed the *NSFNET* topology [34] to "multi-domain" by



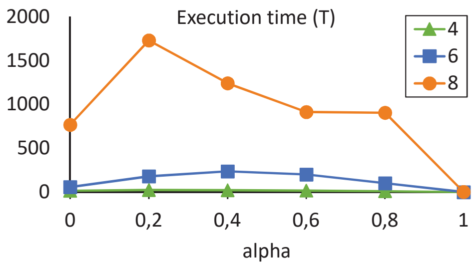
(a) The average goal value



(b) The average link utilization



(c) The average slice deployment



(d) The average execution time (s)

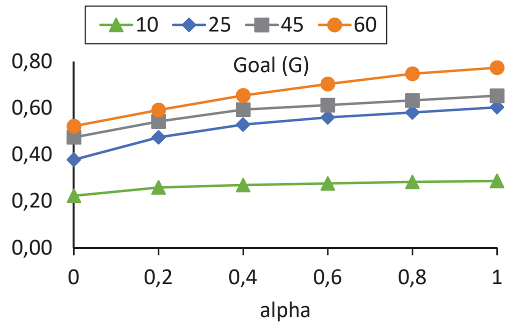
Fig. 6. Evaluation1 Cost266: The results of the model for demand set $\{4, 6, 8\}$ - G, U, S and T

assigning nodes to domains (Fig. 8). Both the ILP model and the heuristic were run with the following configurations (5x2):

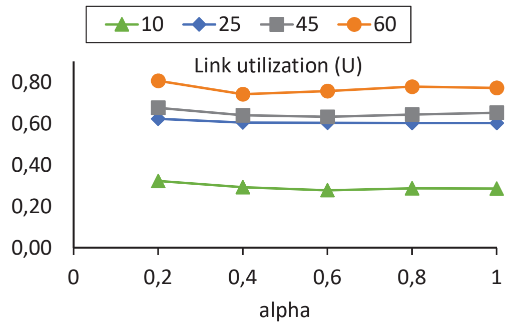
- 5 topology configurations. Each topology configuration was generated with a randomly assigned maximum link capacity and latency. Each link (Fig. 8) received a capacity from the range of 5-10 Gbit/s. All links received latency in the range of 3-10 ms.
- 2 data center (DC) configurations. Each DC configuration contained seven randomly selected DCs - two located in the East, one in the Center, one in the South, and three in the West. For each DC, we randomly assigned two VNFs out of four from the VNF catalog.

We generated eight random demands. A single user demand was defined in the following way:

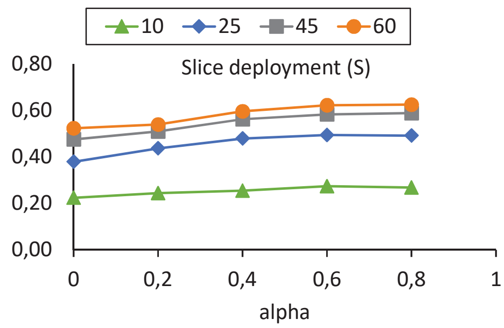
- a random start and end (in a different domain)
- a random bitrate in the range of 100-400 Mbit/s
- a random SFC (two ordered VNFs selected from the VNF catalog)
- a random maximum latency. In total, 25% of all demands in a demand set (D) were low latency (100 ms), and others were latency insensitive.



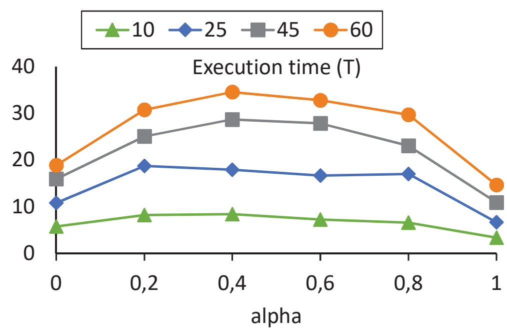
(a) The average goal value



(b) The average link utilization



(c) The average slice deployment



(d) The average execution time (s)

Fig. 7. Evaluation1 Cost266: The results of the heuristic for demand set $\{10, 25, 45, 60\}$ - G, U, S and T

The heuristic was run with step $d = 2$. As observed in this scenario, the ILP model (M) is more efficient (for any α, it uses less network resources to embed the demand set) than the heuristic (H) by up to 27% (Fig. 9a). On the other hand, the heuristic is much faster (Fig. 9d). A single average max computation is approximately 1 s. The model works fast for α=0 and 1. For these cases, the model only minimizes S or U. In such cases, it is easier to find an optimal solution. When $\alpha * U$ and $(1 - \alpha) * S$ have similar values, there are more feasible solutions with similar values to be analyzed by the model.

Additionally, we note that the G function is decreasing (Fig. 9a), which is different compared to Fig. 6a and Fig. 7a, which are both increasing. The behavior of G depends on: the proportion of the sum of all demanded bitrates per average link capacity; the proportion of the number of demands per cardinality of the set A_s; and the structure of the network. The average link capacities in the two experiments were different. The networks also have different size and structure. Furthermore, the monotonicity of the goal function G, in general, is not guaranteed.

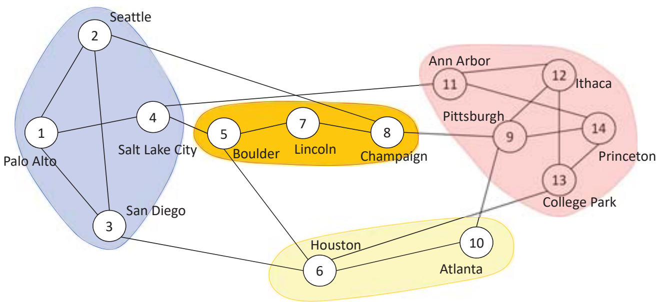
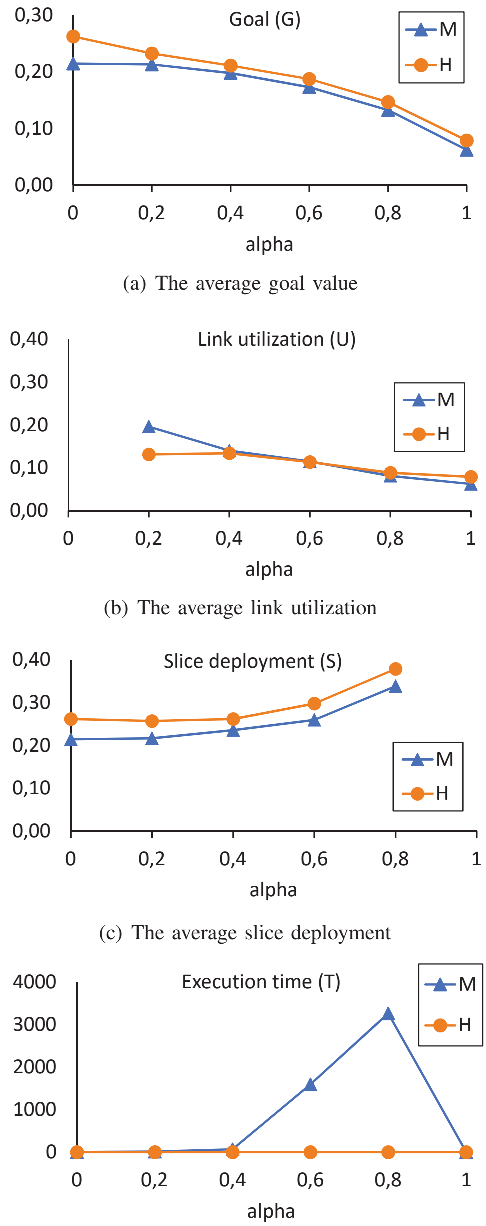


Fig. 8. Evaluation2 Multi-domain topology based on *NSFNET* [34]



(a) The average goal value

(b) The average link utilization

(c) The average slice deployment

(d) The average execution time (s)

Fig. 9. Evaluation2 NSFNET: comparison of the model (M) and heuristic (H) for demand set {8}

C. Multi-domain SFC Emulation - Proof of Concept

The SFC proof of concept is based on Containernet, Docker (VNFs), Open vSwitch (OVS), and Linux Router. We set up a network of three domains (Domain 1-3, Fig. 10), with several OpenFlow-enabled OVS switches (performing packet forwarding within a domain) and Linux IP Routers (that connect different domains). Two user networks (UN) are connected to each domain. To emulate the security scenario (IDS resource share), we use Snort IDS (VNF). Snort VNFs are located in Data Centers 1, 2, and 3 (DC for short). DC4 contains a Storage VNF. For simplicity, each link capacity (inter-domain and intra-domain) has the same capacity, equal to 2 units.

We assume that slice-able links are switch-to-switch links (s_i, s_j). In our topology, there are 15 slice-able links (6 in Domain 1, 5 in Domain 2, 1 in Domain 3, and 3 inter-domain slice links).

If, according to the slice definition, the network traffic is to be handled by a VNF service (e.g. IDS - Snort), e.g. transfer of packets from switch S6 to the data center DC1 (Fig. 11a, orange slice), this is done by copying the traffic to the appropriate network interface of a DC. Additionally, the network interface in the DC (e.g. Snort machine) must be configured in promiscuous mode to enable the capture and reading of every network packet that reaches its interface.

For emulation of our two scenarios, we define a set of two E2E demands (E2E slices), each requires a single VNF in the chain:

- D1 (orange)={origin: UN1, target: UN3, bitrate=1 unit, VNF=(IDS)}
- D2 (green)={origin: UN2, target: UN4, bitrate=1 unit, VNF=(IDS)}

For both scenarios, we compute E2E slices using the ILP model and provision them in our Containernet environment: first by installing OpenFlow rules on OVS switches (intra-domain), and then by installing IP routing entries (inter-domain).

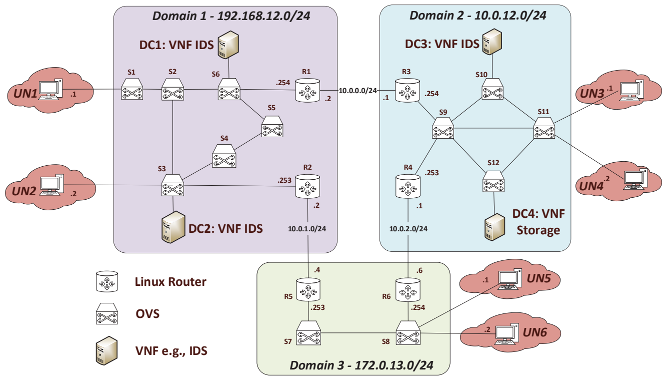


Fig. 10. Containernet emulation - environment

1) Emulation Scenario1 - Maximum Link Utility minimized: In this scenario (Fig. 11a), we minimize maximum link utility. In our goal function $\min \alpha\ U + (1-\alpha)\ S$, by setting $\alpha = 1$ we receive $\min\ U$. The computation resulted in $U = 0.5$ (any link is used in a maximum of 50%). As one can observe (Fig. 11a), slices tend to be disjoint (spread). As explained above, the computed data paths (slices) are provisioned by generating and installing the corresponding OpenFlow rules in all traversed domains.

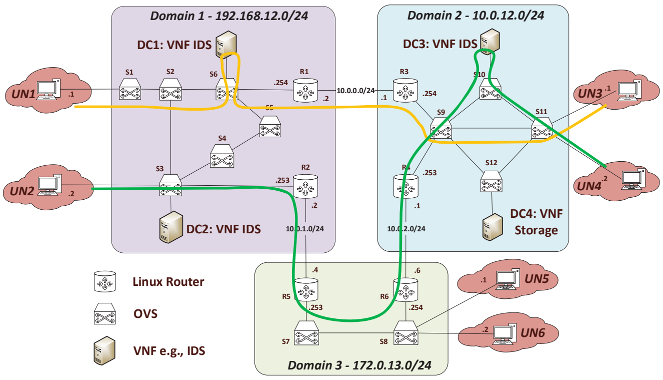
2) Emulation Scenario2 - National Slice Usage minimized: We minimize the number of used national slices in this scenario (Fig. 11b). In our goal function $\min \alpha\ U+(1-\alpha)\ S$, by setting $\alpha = 0$ we receive $\min\ S$. The computation resulted in $S = 0.33(3)$ (used 5 slices out of 15; however, link utility is sacrificed $U = 1$). As one can observe (Fig. 11b), slices tend to overlap (orange, green). As explained above, the computed data paths (slices) are provisioned by generating and installing the corresponding OpenFlow rules in all traversed domains.

VII. Conclusions

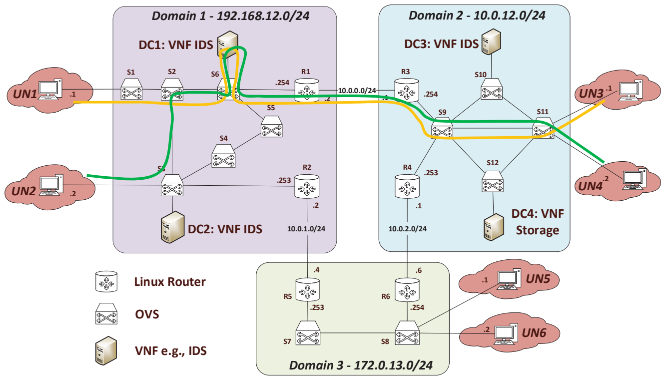
Since modern coalition networks are expected to simultaneously provide connectivity (i.e., QoS-enabled paths) and computational capabilities (i.e., network functions), it is necessary to use efficient methods for embedding E2E slices. Such efficient methods are crucial in security scenarios to properly handle network and compute resources and protect the network against cyber-attacks.

The computational results reveal that:

1) Extensive simulations of the ILP model and heuristic, on realistic topologies show the correct trend of the goal values (TE & SD) as the α parameter (trade-off) and the number of demands change.
2) In terms of goal value, the ILP model is more efficient than the heuristic (it uses less network resources to embed the demand set). The observed average gap is up to 27% (8 demands) on a network of practical size (based on *NSFNET*).
3) On the other hand, the heuristic is much faster. On a large-scale network (based on *Cost266*) the observed average execution time is below 35 s (60 demands), while the ILP model's average run time is up to 1730 s (for only 8 demands).
4) While the ILP model efficiently solves SFC problems of a practical size, the heuristic can solve large-scale problems very fast with decent efficiency, leading to a trade-off between the two.



(a) Scenario1: maximum link utility minimized



(b) Scenario2: national slice usage minimized

Fig. 11. Containernet emulation - evaluation scenarios

As for future work, the following extensions to our methodology are worth considering:

1) In the evaluation, we assume the simplification that the intra-domain topology fully contributes to the federated topology. In future work, we plan to address the intra-domain SFC mapping problem algorithmically.
2) Investigate how different ways of defining a federated topology from a local topology (taking into account local slices and compute nodes) affect the performance and efficiency of the entire two-level system in terms of resource utilization and admission ratios.

Acknowledgment

This work was supported by the European Defence Agency (EDA) under project No B 1520 IAP4 GP ”Software Defined Tactical and Theatre Network”.

References

[1] H. Li and L. Wang, “Online orchestration of cooperative defense against ddos attacks for 5g mec,” in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, 2018, pp. 1–6, doi: 10.1109/WCNC.2018.8377309.

[2] M. Pióro, M. Mycek, A. Tomaszewski, K. Junosza-Szaniawski, and D. Nogalski, "Finding optimal mixed strategies in a matrix game between the attacker and the network operator," in *13th International Workshop on Resilient Networks Design and Modeling, RNDM 2023, Hamburg, Germany, September 20-22, 2023*. IEEE, 2023, pp. 1–8, doi: 10.1109/RNDM59149.2023.10293051.

[3] K. Junosza-Szaniawski and D. Nogalski, "Game-theoretic approach to attack planning and controller placement in software defined networks," in *International Conference on Military Communications and Information Systems, ICMCIS 2023, Skopje, North Macedonia, May 16-17, 2023*. IEEE, 2023, pp. 1–8, doi: 10.1109/ICMCIS59922.2023.10253594.

[4] K. Junosza-Szaniawski, D. Nogalski, and P. Rzazewski, "Exact and approximation algorithms for sensor placement against ddos attacks," *Int. J. Appl. Math. Comput. Sci.*, vol. 32, no. 1, pp. 35–49, 2022, doi: 10.34768/amcs-2022-0004.

[5] K. Junosza-Szaniawski, D. Nogalski, and A. Wójcik, "Exact and approximation algorithms for sensor placement against ddos attacks," in *2020 15th Conference on Computer Science and Information Systems (FedCSIS), 13th International Workshop on Computational Optimization, Sofia, Bulgaria*, 2020, pp. 295–301, doi: 10.15439/2020F106.

[6] P. Blazek, T. Gerlich, and Z. Martinasek, "Scalable ddos mitigation system," in *2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*, July 2019, pp. 617–620, doi: 10.1109/TSP.2019.8768869.

[7] S. K. Fayaz, Y. Tobioka, V. Sekar, and M. D. Bailey, "Bohatei: Flexible and elastic ddos defense," in *24th USENIX Security Symposium, USENIX Security 15, Washington, D.C., USA, August 12-14, 2015*, J. Jung and T. Holz, Eds. USENIX Association, 2015, pp. 817–832. [Online]. Available: https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/fayaz

[8] D. Belabed, M. Bouet, and V. Conan, "Centralized defense using smart routing against link-flooding attacks," in *2nd Cyber Security in Networking Conference, CSNet 2018, Paris, France, October 24-26, 2018*, P. Hebrard and S. Ghernaouti, Eds. IEEE, 2018, pp. 1–8, doi: 10.1109/CSNET.2018.8602966.

[9] M. T. Beck and J. F. Botero, "Scalable and coordinated allocation of service function chains," *Comput. Commun.*, vol. 102, pp. 78–88, 2017, doi: 10.1016/j.comcom.2016.09.010.

[10] N. Toumi, O. Bernier, D. Meddour, and A. Ksentini, "On using physical programming for multi-domain SFC placement with limited visibility," *IEEE Trans. Cloud Comput.*, vol. 10, no. 4, pp. 2787–2803, 2022, doi: 10.1109/TCC.2020.3046997.

[11] D. Nogalski, D. Belabed, A. Triollet, K. Junosza-Szaniawski, S. Abdellatif, P. Berthou, S. Pedebearn, and A. Dudko, "Federated SFC placement in sliced collaborative multi-administrative multi-domain networks," in *2024 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, 2024, pp. 1–6, doi: 10.23919/SoftCOM62040.2024.10721844.

[12] G. Hallingstad and S. Oudkerk, "Protected core networking: an architectural approach to secure and flexible communications," *IEEE Commun. Mag.*, vol. 46, no. 11, pp. 35–41, 2008, doi: 10.1109/MCOM.2008.4689242.

[13] B. Addis, D. Belabed, M. Bouet, and S. Secci, "Virtual network functions placement and routing optimization," in *4th IEEE International Conference on Cloud Networking, CloudNet 2015, Niagara Falls, ON, Canada, October 5-7, 2015*. IEEE, 2015, pp. 171–177, doi: 10.1109/CloudNet.2015.7335301.

[14] A. Wion, M. Bouet, L. Iannone, and V. Conan, "Change in continuity: Chaining services with an augmented IGP," *IEEE Trans. Netw. Serv. Manag.*, vol. 16, no. 4, pp. 1332–1344, 2019, doi: 10.1109/TNSM.2019.2944011.

[15] Y. Peng and B. Di, "Joint VNF deployment and resource allocation in integrated terrestrial-aerial access networks enabled by network slicing," in *20th IEEE International Conference on Embedded and Ubiquitous Computing, EUC 2022, Wuhan, China, December 9-11, 2022*. IEEE, 2022, pp. 74–80, doi: 10.1109/EUC57774.2022.00021.

[16] Y. Yang, B. Wang, J. Tian, and P. Luo, "Efficient sfc protection method against network attack risks in air traffic information networks," *Electronics*, vol. 13, no. 13, 2024, doi: 10.3390/electronics13132664.

[17] A. Murray, A. Arulselvan, M. Roper, M. Cashmore, S. K. Mohalik, I. Burdick, and S. David, "The cost of quality of service: Sla aware vnf placement and routing using column generation," in *2023 13th International Workshop on Resilient Networks Design and Modeling (RNDM)*, 2023, pp. 1–8, doi: 10.1109/RNDM59149.2023.10293056.

[18] H. Ko, D. Suh, H. Baek, S. Pack, and J. Kwak, "Optimal placement of service function in service function chaining," in *Eighth International Conference on Ubiquitous and Future Networks, ICUFN 2016, Vienna, Austria, July 5-8, 2016*. IEEE, 2016, pp. 102–105, doi: 10.1109/ICUFN.2016.7536993.

[19] L. Popokh, J. Su, S. Nair, and E. V. Olinick, "Illumicore: Optimization modeling and implementation for efficient VNF placement," in *International Conference on Software, Telecommunications and Computer Networks, SoftCOM 2021, Split, Hvar, Croatia, September 23-25, 2021*. IEEE, 2021, pp. 1–7. [Online]. Available: https://ieeexplore.ieee.org/document/9559076

[20] J. Gil Herrera and J. F. Botero, "Resource allocation in nfv: A comprehensive survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 518–532, 2016, doi: 10.1109/TNSM.2016.2598420.

[21] A. Belbekkouche, M. M. Hasan, and A. Karmouch, "Resource discovery and allocation in network virtualization," *IEEE Commun. Surv. Tutorials*, vol. 14, no. 4, pp. 1114–1128, 2012, doi: 10.1109/SURV.2011.122811.00060.

[22] J. F. Botero, M. Molina, X. Hesselbach-Serra, and J. R. Amazonas, "A novel paths algebra-based strategy to flexibly solve the link mapping stage of VNE problems," *J. Netw. Comput. Appl.*, vol. 36, no. 6, pp. 1735–1752, 2013, doi: 10.1016/j.jnca.2013.02.029.

[23] A. Fischer, J. F. Botero, M. T. Beck, H. de Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Commun. Surv. Tutorials*, vol. 15, no. 4, pp. 1888–1906, 2013, doi: 10.1109/SURV.2013.013013.00155.

[24] M. Drwal, "Approximation algorithms for utility-maximizing network design problem," in *Progress in Systems Engineering - Proceedings of the Twenty-Third International Conference on Systems Engineering, ICSEng 2014, Las Vegas, NV, USA, August 19-21, 2014*, ser. Advances in Intelligent Systems and Computing, H. Selvaraj, D. Zydek, and G. Chmaj, Eds., vol. 366. Springer, 2014, pp. 411–416, doi: 10.1007/978-3-319-08422-0_60.

[25] K. Junosza-Szaniawski and D. Nogalski, "Exact and approximation algorithms for joint routing and flow rate optimization," in *Preprints of Communication Papers of the 2019 Federated Conference on Computer Science and Information Systems, FedCSIS 2019, Leipzig, Germany, September 1-4, 2019*, ser. Annals of Computer Science and Information Systems, M. Ganzha, L. A. Maciaszek, and M. Paprzycki, Eds., vol. 20, 2019, pp. 29–36, doi: 10.15439/2019F85.

[26] S. Even, A. Itai, and A. Shamir, "On the complexity of time table and multi-commodity flow problems," in *16th Annual Symposium on Foundations of Computer Science (sfcs 1975)*, 1975, pp. 184–193, doi: 10.1109/SFCS.1975.21.

[27] N. Tastevin, M. Obadia, and M. Bouet, "A graph approach to placement of service functions chains," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Lisbon, Portugal, May 8-12, 2017*. IEEE, 2017, pp. 134–141, doi: 10.23919/INM.2017.7987273.

[28] M. Obadia, J. Rougier, L. Iannone, V. Conan, and M. Bouet, "Revisiting NFV orchestration with routing games," in *2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Palo Alto, CA, USA, November 7-10, 2016*. IEEE, 2016, pp. 107–113, doi: 10.1109/NFV-SDN.2016.7919484.

[29] R. A. Addad, M. Bagaa, T. Taleb, D. L. C. Dutra, and H. Flinck, "Optimization model for cross-domain network slices in 5g networks," *IEEE Trans. Mob. Comput.*, vol. 19, no. 5, pp. 1156–1169, 2020, doi: 10.1109/TMC.2019.2905599.

[30] A. El-Amine, O. Brun, S. Abdellatif, and P. Berthou, "Shortening the deployment time of sfcs by adaptively querying resource providers," in *IEEE Global Communications Conference, GLOBECOM 2021, Madrid, Spain, December 7-11, 2021*. IEEE, 2021, pp. 1–6, doi: 10.1109/GLOBECOM46510.2021.9685356.

[31] Y. Wu and J. Zhou, "Dynamic service function chaining orchestration in a multi-domain: A heuristic approach based on srv6," *Sensors*, vol. 21, no. 19, p. 6563, 2021, doi: 10.3390/s21196563.

[32] S. Pedebearn, S. Abdellatif, P. Berthou, D. Nogalski, and D. Belabed, "Virtual link embedding in collaborative sliced multi-administrative multi-domain networks," in *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing, SAC 2024, Avila, Spain, April 8-12, 2024*, J. Hong and J. W. Park, Eds. ACM, 2024, pp. 1088–1095, doi: 10.1145/3605098.3636067.

[33] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski, "Sndlib 1.0 - survivable network design library," *Networks*, vol. 55, no. 3, pp. 276–286, 2010, doi: 10.1002/net.20371.

[34] Q. Zhu, X. Yu, Y. Zhao, A. Nag, and J. Zhang, "Auxiliary-graph-based energy-efficient traffic grooming in ip-over-fixed/flex-grid optical networks," *Journal of Lightwave Technology*, vol. 39, no. 10, pp. 3011–3024, 2021, doi: 10.1109/JLT.2021.3057389.

Dariusz Nogalski received his M.Sc. degree in applied computer science from the Warsaw University of Technology, Faculty of Mathematics and Information Science, Poland. He works as a scientist at the Military Communications Institute – National Research Institute, Poland. He is currently pursuing his Ph.D. degree. His current research interests include network optimization, and cybersecurity in computers and networks. His focus is resource allocation, network virtualization, and software-defined networking, also in multi-domain context.

Dallal Belabed , since 2011, she has been involved in network virtualization research and development in academia and defense/space industries. She earned her Ph.D. from LIP6-CNRS (UPMC) in 2015 and worked as a Temporary Teacher and Researcher there from 2014 to 2015. In 2016, she joined Thales R&T as an SDN and NFV Network Architect, where she led the H2020 Anastacia project. She transitioned to Airbus Defense & Space in 2020, becoming an Expert in Software Defined Networks. Currently, she serves as the Airbus technical leader of virtualized network orchestration projects for the SOFTANET European Defense Agency (EDA) project. She is an inventor of patents and author of book chapters and papers published internationally. Her career has spanned academia, research, and industry, focusing on cutting-edge technologies in networking and virtualization.

Alexandre Triollet is a graduate engineer from ESIEA in system and network security. Since graduation, he has worked as a consultant on multiple subjects around systems and networks in various environments. His domain of expertise is in tasks automation of network and systems infrastructure, which made him fit to define and configure the environment as well as develop the tools to experiment on this publication topic.

Konstanty Junosza-Szaniawski received his Ph.D. in mathematics from the Warsaw University of Technology (Faculty of Mathematics and Information Science) in 2004. He obtained his habilitation in mathematics from Warsaw University of Technology in 2020. He works as a professor at the Faculty of Mathematics and Information Science of the Warsaw University of Technology. His research interests are related to cybersecurity, graph theory, especially graph coloring, geometrically defined graphs, networks, and graph algorithms.

Slim Abdellatif received the M.Sc. and Ph.D. degrees in computer science both from the University of Toulouse, France. He is currently an Assistant Professor at the Institut National des Sciences Appliquées of Toulouse and a research scientist at the Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS) from the French National Center for Scientific Research (CNRS). His current research interests include network virtualization, Software Defined Networking, QoS and resource management in softwarized wired and wireless networks.

Pascal Berthou received his Ph.D. in computer science and telecommunications from the National Polytechnic Institute of Toulouse in 2001. He is currently a Professor at Toulouse University of Science and has been a research staff member at LAAS/CNRS since 1998. His research focuses on two key areas: satellite communication systems and sensor networks. In satellite communications, his work aims to enhance Quality of Service (QoS) management, with recent efforts concentrated on virtualization techniques. In sensor networks, he specializes in wireless sensor network design for instrumentation, with a focus on reducing energy consumption by optimizing hardware/software network interfaces. Over the years, he has contributed to 5 books on satellite communication systems and 2 on wireless systems, reflecting his extensive work in these domains.

Stanislas Pedebearn is a Ph.D. student at LAAS-CNRS, France, since 2021. His research focuses on network optimization across federated and multi-administrative networks, and he was actively involved in the Softanet project. Stanislas holds a master's degree in Computer Science, Internet technologies, from Université de Pau et des Pays de l'Adour.

Adam Dudko received his B.Sc. degree in environmental engineering from the Białystok University of Technology, Poland. He works as a computer network engineer at the Military Institute of Communications—National Research Institute. He is an expert in Cisco switching and routing and an active Python programmer.