



HAL
open science

Federated SFC Placement in Sliced Collaborative Multi-Administrative Multi-Domain Networks

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

► To cite this version:

Dariusz Nogalski, Dallah Belabed, Alexandre Triollet, Konstanty Junosza-Szaniawski, Slim Abdellatif, et al.. Federated SFC Placement in Sliced Collaborative Multi-Administrative Multi-Domain Networks. 2024 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), Sep 2024, Split, Croatia. pp.1-6, 10.23919/SoftCOM62040.2024.10721844 . hal-04866703

HAL Id: hal-04866703

<https://laas.hal.science/hal-04866703v1>

Submitted on 6 Jan 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Federated SFC Placement in Sliced Collaborative Multi-Administrative Multi-Domain Networks

1st Dariusz Nogalski*

*Military Communications Institute
National Research Institute
Zegrze, Poland
dariusz.nogalski@wil.waw.pl*

1st Dallal Belabed*

*Research and Development department
Airbus Defence and Space
Issy-les-Moulineaux, France
dallal.belabed@airbus.com*

2nd Alexandre Triollet

*Research and Development department
Airbus Defence and Space
Issy-les-Moulineaux, France
alexandre.triollet@fr.airbus.com*

3rd Konstanty Junosza-Szaniawski

*Faculty of Mathematics and Information Science
Warsaw University of Technology
Warsaw, Poland
konstanty.szaniawski@pw.edu.pl*

4th Slim Abdellatif

*LAAS-CNRS
Université de Toulouse, CNRS, INSA
Toulouse, France
slim@laas.fr*

5th Pascal Berthou

*LAAS-CNRS
Université de Toulouse, CNRS, UPS
Toulouse, France
berthou@laas.fr*

6th Stanislas Pedebearn

*LAAS-CNRS
Université de Toulouse, CNRS, UPS
Toulouse, France
spedebearn@laas.fr*

7th Adam Dudko

*Military Communications Institute
National Research Institute
Zegrze, Poland
adam.dudko@wil.waw.pl*

Abstract—This paper presents a method to embed end-to-end federated service chains. First, the ILP model is provided, which solves the resource allocation problem (network and compute), where each demand is defined via bandwidth, delay, and service function chain requirements. Second, an efficient heuristic is proposed and compared experimentally with the ILP model on various known multi-domain topologies. Finally, a network emulation security scenario is implemented in Containernet as a proof of concept.

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

I. INTRODUCTION

With the advent of network function virtualization (NFV), software networks are no longer limited to providing connectivity (i.e., a path with QoS) but also offer computational capabilities (i.e., network functions) along the path. A network service can then be represented as an SFC (Service Function Chain) composed of a set of VNFs (Virtual Network Functions) and directional virtual links that connect them (a flow of packets goes through the chain of VNFs composing the SFC). In a multi-domain network, NFV may be deployed and hosted by different domains. These latter may belong to the same (or other) authority and correspond to different network technologies or segments (access, backhaul, core). In this work, we deal with SFC embedding in a multi-domain (multi-administrative) environment and compose E2E multi-domain SFC services based on available national network and

compute resources (slices, VNFs). In particular, we consider the case of federated military coalition networks.

When embedding a multi-domain SFC into a multi-administrative federation of domains, the initiating domain relies on the abstracted topologies exposed by other domains (i.e., a compact view of each domain's network topology) to combine the multiple parts and create an E2E SFC. Classically, to limit the disclosure of topology information, the abstract topology shared with other domains typically consists of border nodes and abstract links connecting them. Some domains may also expose compute nodes with a set of VNFs they can host. Finally, as the adoption of network slicing increases, domain-level slices providing predefined service types (e.g., low-latency slice) may also be exposed and seen by other domains as abstract domain-level slice links with some QoS guarantees.

The embedding algorithm executed on these abstracted topologies is referred to as federated-level embedding. It assigns network and computing resources exposed by domains to the requested multi-domain SFC. All of these resources should be checked and confirmed, as some aspects of the abstractions may be out of date, even though they were valid at the time of announcement. Domain-level sub-SFC embeddings are then triggered on the selected domains.

Without loss of generality, in this paper we focus on federated-level embedding and implicitly assume that all advertised abstractions remain valid at the time of provisioning of the requested multi-domain SFC.

* These two authors contributed equally to this work

A. Related work

Existing works on single-domain address e.g., SFC in centralized [2] [14] and decentralized setting [14], latency [12]. Addis et al. [2] solve the VNF chain routing problem; however, the chain is unordered. In this work, we propose an ordered SFC. Wion et al. [14] compare centralized and decentralized settings. Peng and Di [10] optimize jointly the VNF deployment, power, and spectrum resource allocation.

In multi-domain, the literature presents various studies to address slice embedding, particularly within the 5G/6G [1]. The centralized orchestration requires a global view of the infrastructure, which raises scalability concerns. The desire to reduce communication costs has led to developing distributed slice embedding solutions. Our work addresses the multi-domain slice embedding by proposing an Integer Linear Programming (ILP) model that solves VNF and Link embedding simultaneously (in existing works, they are usually solved separately). Other related works on multi-domain SFC address, e.g., reduction of deployment time [3], dynamic orchestration [15], and SFC placement with limited visibility [13].

B. Motivation

We assume our approach will be beneficial in coalition networks. In the case of federated military networks, allied nations connect and share part of their network infrastructure to build a Federation Mission Network (FMN) [5]. In addition to such a coalition-based multi-domain network, it is crucial to establish special multi-domain SFC-based E2E virtual networks to achieve mission goals. All nations advertise their resources for E2E purposes. Nations can complement each other's capabilities to build comprehensive network services. The responsible party acquires the advertised resources and coordinates the configuration/modification (up-scaling/down-scaling) and decommissioning of the service. This coordinating entity asks participating nations to provide resources (e.g., a link with QoS or VNF). Still, the management of the physical resources of the segment at the national level is the responsibility of each nation (responsible for the segment).

Our work provides important insights for adopting slicing in a federated network for the following reasons. First, a single nation may not have all the necessary network capabilities to define an E2E SFC (e.g., degradation due to a DDoS attack [6] or a control plane attack [11]). Second, it may support several interesting security scenarios e.g.: (1) Following [4], traffic scrubbing by remote VNFs. As soon as we open military networks toward the Internet of Things the risk of volumetric attacks will increase [6]. (2) Following [7], MEC (Multi-Access Edge Computing) (5G) resilience to attacks from end-user networks can be increased by using shared IDS (Intrusion Detection System) resources.

C. Our contribution

First, we deal with SFC embedding in a multi-domain context. The structure of a federated topology is described in more detail in our previous work [9]. Users are allowed to compose E2E SFC services spanning different domains.

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 $(f^{k,1}, \dots, f^{k,T_k})$
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 $f^{k,1}, \dots, f^{k,p}$ 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 $k \in D, p \in \{0, \dots, T_k\}$ ($e_{i,j} = \max\{x_{i,j}^{k,p} : k \in D, p \in \{0, \dots, T_k\}\}$)
$z_i^{k,p}$	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

Second, we provide an exact ILP model (sec. II-A), which is different from previous works [2] [14]. We use a different goal function (a combination of slice deployment cost and link utility minimization). We use a (de)compression VNF and SFC ordering. Each topology link can be sliceable (national slice). In addition, since the problem is NP-hard [1], we provide an efficient heuristic (sec. II-B).

Third, we performed extensive experiments on known topologies: in sec. III-A, we demonstrate relatively high model computation time (and low of the heuristic); in sec. III-B, we show the efficiency gap between the model and the heuristic, in favor of the former. Finally, our SFC contribution begins with a proposal of the embedding algorithm (sec. II-A, II-B) up to prototyping on ContainerNet emulation environment (with Docker-based VNFs) with effective provisioning of SFC demands (sec. III-C).

II. MATHEMATICAL FORMULATION

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 (sec. II-A) and greedy heuristic (sec. II-B).

The central part of the demand model is the definition of the service chains (see Table I). The tuple $(f^{k,1}, \dots, f^{k,T_k})$ (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 $p \in \{1, \dots, T_k\}$, $b_k^p = b_k \cdot \prod_{q=1}^p \mu_{f^{k,q}}$. The b_k^p denotes bandwidth used by demand $k \in D$ after the first p services $f^{k,1}, \dots, f^{k,p}$ are performed.

A. 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 [9]. 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 $-b_k^p$.

$$\forall k \in D \forall i \in N \forall p \in \{0, \dots, T_k\}$$

$$\sum_{(i,j) \in A} \phi_{i,j}^{k,p} - \sum_{(j,i) \in A} \phi_{j,i}^{k,p}$$

$$= \begin{cases} b_k^0 & p = 0, i = o_k & (1a) \\ -b_k^0 \cdot z_i^{k,1} & p = 0, i \in N - \{o_k\} & (1b) \\ b_k^p \cdot z_i^{k,p} - b_k^p \cdot z_i^{k,p+1} & p \in \{1, \dots, T_k - 1\}, i \in N & (1c) \\ b_k^{T_k} \cdot z_i^{k,T_k} & p = T_k, i \in N - \{t_k\} & (1d) \\ -b_k^{T_k} & p = T_k, i = t_k & (1e) \end{cases}$$

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

$$\forall k \in D \forall p \in \{1, \dots, T_k\} z_i^{k,p} = 0 \quad (2)$$

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

$$\forall k \in D \forall p \in \{1, \dots, T_k\} \sum_{i \in N_{f^{k,p}}} z_i^{k,p} = 1 \quad (3)$$

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

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

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

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

- 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

$$\forall k \in D \forall p \in \{0, \dots, T_k\} \forall (i, j) \in A_s x_{i,j}^{k,p} \leq e_{i,j} \quad (6)$$

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

$$\forall (i, j) \in A_s \sum_{k \in D} \sum_{p \in \{0, \dots, T_k\}} x_{i,j}^{k,p} \geq e_{i,j} \quad (7)$$

- 8) The sum of flows does not exceed the edge capacity. For online version of the algorithm (also for sec. II-B)

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

We consider two objective functions:

- Traffic Engineering (TE) goal: minimize the maximum network link utilization (e.g., to increase future demand admissibility):

$$\min U \quad (8)$$

- Slice Deployment (SD) goal: minimize the number of used slices (e.g., to reduce slice setup time):

$$\min \sum_{(i,j) \in A_s} e_{i,j} \cdot \frac{1}{|A_s|} \quad (9)$$

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

$$\min \alpha U + (1 - \alpha) S \quad (10)$$

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

B. 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).

Algorithm 1 SFC Greedy Heuristic

- 1: 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
- 2: 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 $p \in \{0, \dots, T_k\}$
- 3: Sort demands D by latency (from low latency to high latency (latency insensitive))
- 4: **while** $D \neq \emptyset$ **do**
- 5: Select d demands $D_d \in D$
- 6: Run SFC ILP Model (sec II-A) 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 $k \in D$, $(i, j) \in A$ and $p \in \{0, \dots, T_k\}$
- 7: Add $\phi_{i,j}^{k,p}$ assignments to the result ϕ (list of embedded paths)
- 8: Update available capacity $\forall (i, j) \in A$, $\gamma_{i,j} = \gamma_{i,j} - \sum_{k \in D} \sum_{p \in \{0, \dots, T_k\}} \phi_{i,j}^{k,p}$
- 9: $D = D \setminus D_d$
- 10: return ϕ
- 11: **end while**

III. PERFORMANCE EVALUATION

The model (sec. II-A) was implemented in Optimization Programming Language (IBM CPLEX v22.1), and the heuristics (sec. II-B) 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 [8], by assigning nodes (cities) to domains (countries) (Fig. 1).

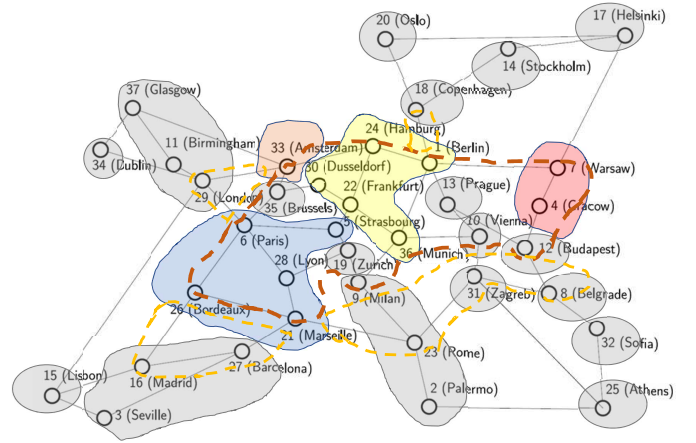


Fig. 1. Evaluation1 Multi-domain topology based on Cost266 [8]

1) *Result of ILP model:* We conducted the tests of the model with the following configurations (4x2): 4 topology configurations, random capacity from range 5-10 Gbit/s (links in red dotted area (Fig. 1)), 2-5 Gbit/s (orange dotted area), and 1-2 Gbit/s (peripheral); 2 random data center (DC) configurations, each with 7 randomly selected DCs - two in France, three in Germany, one in Poland and one in Netherlands. We assigned two VNFs out of four (VNF catalog = {FW (Firewall), IDS, DPI (Deep Packet Inspection), Traffic Storage}) to each selected DC. We vary the number of demands {4, 6, 8}. Each demand (req. random bitrate 100-400 Mbit/s, 2 VNFs) starts and ends in a different domain. In total, 25% of all demands are low latency (100 μ s), and others are latency insensitive. Each measurement with different value of α (eq. 10).

As one can observe, when the number of demands goes up, the combined utility (G) (eq. 10) goes up, for any value of α (Fig. 2a), showing the expected behavior of the model.

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

Since the model is NP-Hard, we show its significant computation times (Fig. 2b, up to 1730 s for only 8 demands). This is the reason why we designed the heuristic.

TABLE II
EVALUATION1 THE RESULTS OF THE MODEL, DEM. {4, 6, 8} - U AND S

$ D \setminus \alpha$	0	0.2	0.4	0.6	0.8	1	U or S
4	1.00	0.30	0.29	0.28	0.28	0.28	U
6	1.00	0.39	0.31	0.29	0.29	0.29	U
8	1.00	0.41	0.31	0.30	0.29	0.29	U
4	0.13	0.15	0.16	0.17	0.17	1.00	S
6	0.17	0.18	0.20	0.22	0.22	1.00	S
8	0.21	0.22	0.25	0.26	0.27	1.00	S

2) *Result of heuristic:* We conducted tests of the heuristic with the following configurations (4x2): 4 topology configu-

rations, 2 DC configurations. Assumptions as in sec. III-A1. The 60 demands were split into 10, 25, 45, and 60 demand series. The heuristic was run with step $d = 3$.

As one can observe (Fig. 3a), the average G values behave as expected; for any α , they increase as we increase the number of demands. Execution time is much lower compared to the model. A single average max computation is 35 s (60 dem. Fig. 3b).

B. Evaluation2 - simulation on NSFNET

We prove that the model (sec. II-A) is more efficient than the heuristic (sec. II-B). First, we transformed the NSFNET topology [16] to “multi-domain” by assigning nodes to domains (Fig. 4). Both the model and the heuristic were run with configurations (5x2): 5 topology config. (link capacity random from range 5-10 Gbit/s); 2 DC config. (in each random 2 DCs in the East, Centre 1, South 1, and West 3; two VNFs out of four in each DC). We assumed 8 random demands (req. random bitrate 100-400 Mbit/s; 2 VNFs). The heuristic was run with step $d = 2$. As observed in this scenario, the model (M) is more efficient (for any α) than the heuristic (H) by up to 27% (Fig. 5a). On the other hand, the heuristic is much faster (Fig. 5b). A single average max computation is approx. 1 s.

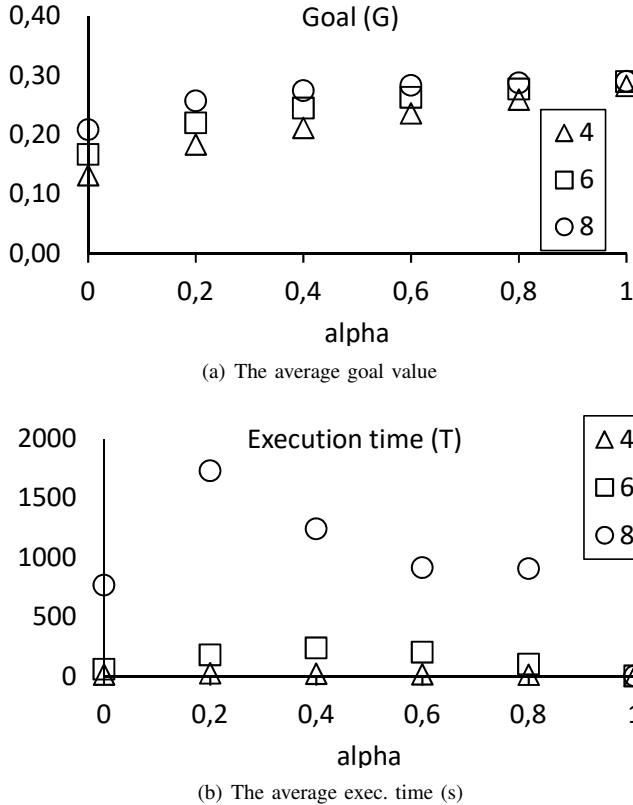


Fig. 2. Evaluation1 The results of the model, dem. {4, 6, 8} - G and T

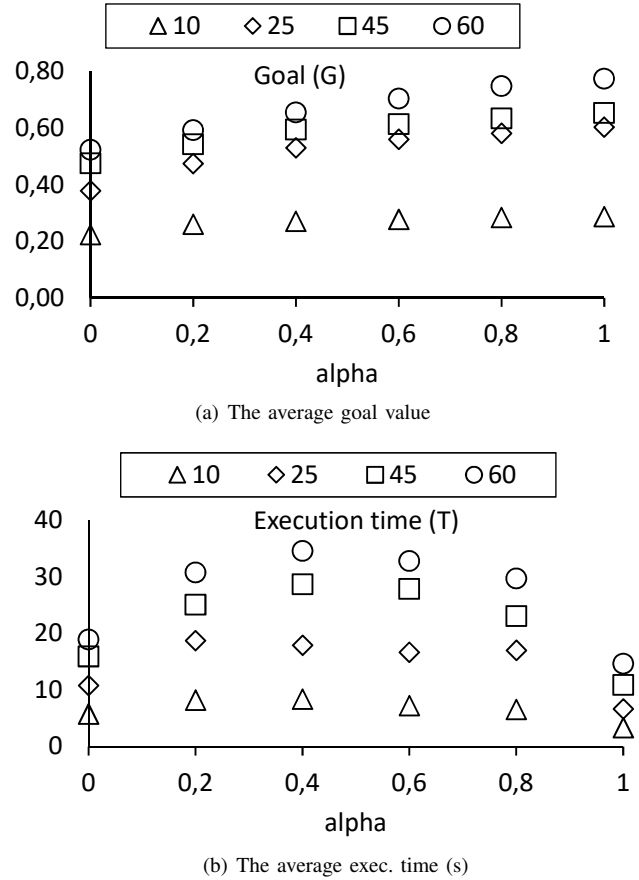


Fig. 3. Evaluation1 The results of the heuristic, dem. {10, 25, 45, 60} - G and T

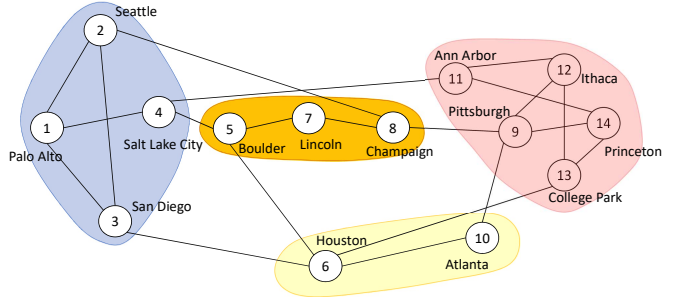
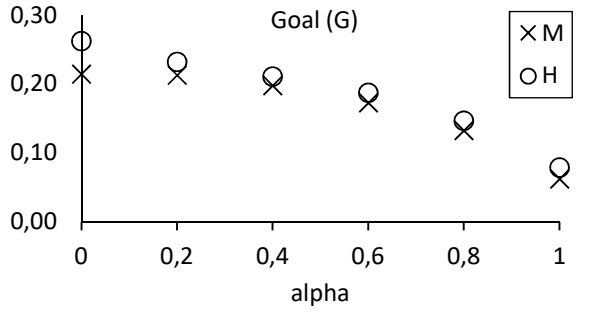


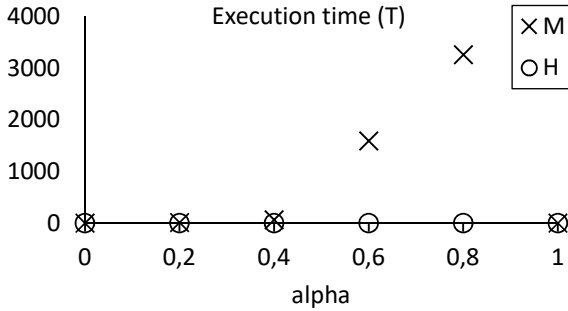
Fig. 4. Evaluation2 Multi-domain topology based on NSFNET [16]

C. Multi-domain SFC emulation - proof of concept

The SFC proof of concept is based on Containernet, Docker (VNFs), OVS (Open vSwitch), and LinuxRouter. We set up the network of three domains, with several OVS switches (intra-domain control) and LinuxRouter (inter-domain control). We compute E2E slices and embed them as OVS switch rules (routing rules for inter-domain). To emulate the security scenario (IDS resource share), we use Snort IDS (VNF) (Fig. 6).



(a) The average goal value



(b) The average exec. time (s)

Fig. 5. Evaluation2 comparison of the model (M) and heuristic (H), dem. {8}

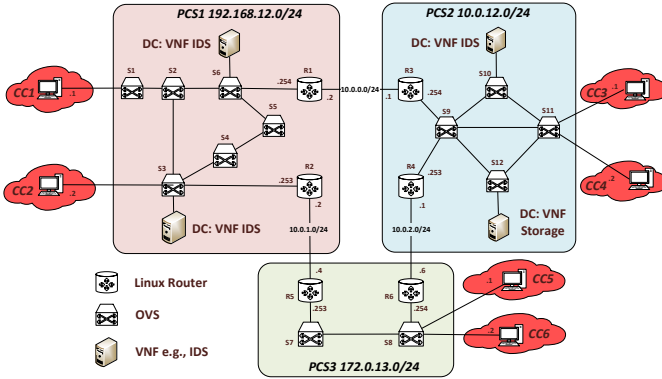


Fig. 6. Containernet emulation environment - proof of concept

IV. CONCLUSIONS

Providing an efficient method for slice embedding is a key enabler for coalition networks. We have proposed abstracted topologies combined with the model and the heuristic. Extensive simulations 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. Second, the model is more efficient (observed gap 27%), while the heuristic is much faster (time below 35 s) on a large network.

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] R. A. Addad, M. Bagaa, T. Taleb, D. L. Cadette Dutra, and H. Flinck. Optimization model for cross-domain network slices in 5g networks. *IEEE Trans. Mob. Comput.*, 19(5):1156–1169, 2020.
- [2] B. Addis, D. Belabed, M. Bouet, and S. Secci. Virtual network functions placement and routing optimization. In *4th CloudNet*, pages 171–177. IEEE, 2015.
- [3] A. El-Amine, O. Brun, S. Abdellatif, and P. Berthou. Shortening the deployment time of sfc by adaptively querying resource providers. In *GLOBECOM*, pages 1–6. IEEE, 2021.
- [4] S. K. Fayaz, Y. Tobioka, V. Sekar, and M. D. Bailey. Bohatei: Flexible and elastic ddos defense. In *USENIX Security*, pages 817–832, 2015.
- [5] G. Hallingstad and S. Oudkerk. Protected core networking: an architectural approach to secure and flexible communications. *IEEE Commun. Mag.*, 46(11):35–41, 2008.
- [6] K. Junosza-Szaniawski, D. Nogalski, and P. Rzazewski. Exact and approximation algorithms for sensor placement against ddos attacks. *Int. J. Appl. Math. Comput. Sci.*, 32(1):35–49, 2022.
- [7] H. Li and L. Wang. Online orchestration of cooperative defense against ddos attacks for 5g mec. In *WCNC*, pages 1–6, 2018.
- [8] S. Orłowski, R. Wessály, M. Pióro, and A. Tomaszewski. Sndlib 1.0 - survivable network design library. *Networks*, 55(3):276–286, 2010.
- [9] S. Pedebearn, S. Abdellatif, P. Berthou, D. Nogalski, and D. Belabed. Virtual link embedding in collaborative sliced multi-administrative multi-domain networks. In *39th ACM/SIGAPP SAC*. ACM, 2024.
- [10] Y. Peng and B. Di. Joint VNF deployment and resource allocation in integrated terrestrial-aerial access networks enabled by network slicing. In *20th EUC*, pages 74–80, 2022.
- [11] M. Pióro, Mariusz 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 RNDM*, pages 1–8. IEEE, 2023.
- [12] L. Popokh, J. Su, S. Nair, and E. V. Olinick. Illumicore: Optimization modeling and implementation for efficient VNF placement. In *29th SoftCOM*, pages 1–7. IEEE, 2021.
- [13] N. Toumi, O. Bernier, D.-E. Meddour, and A. Ksentini. On using physical programming for multi-domain SFC placement with limited visibility. *IEEE Trans. Cloud Comput.*, 10(4):2787–2803, 2022.
- [14] A. Wion, M. Bouet, L. Iannone, and V. Conan. Change in continuity: Chaining services with an augmented IGP. *IEEE Trans. Netw. Serv. Manag.*, 16(4):1332–1344, 2019.
- [15] Y. Wu and J. Zhou. Dynamic service function chaining orchestration in a multi-domain: A heuristic approach based on srv6. *Sensors*, 21(19):6563, 2021.
- [16] 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*, 39(10):3011–3024, 2021.