# Virtual Link Resource Allocation in Federated Multi-Administrative Networks

Stanislas Pedebearn, Slim Abdellatif, Pascal Berthou, Dariusz Nogalski, Dallal Belabed

# Virtual Link Resource Allocation in Federated Multi-Administrative Networks

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

Slim Abdellatif
LAAS-CNRS, Université de Toulouse, CNRS, UPS
Toulouse, France
slim@laas.fr

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

Dariusz Nogalski
Military Communications Institute - National Research Institute
Zegrze, Poland
Dariusz.Nogalski@wil.waw.pl

Dallal Belabed
Airbus Defence and Space, France
Issy-les-Moulineaux, France
dallal.belabed@airbus.com

## ABSTRACT

A federated network can be seen as a collection of network domains, typically under different authorities, who collaborate and share network resources to enable the provision of end-to-end multi-domain services with possibly performance guarantees. To provide these services, domains need to disclose a compact portrayal (or an abstraction or aggregation) of their topology. The interconnection of all exposed abstracted topologies serves as input to compute the appropriate paths (and the resources along these paths) that are needed to support an end-to-end service. Domains basically resort to a topology abstraction that reduces to a mesh of abstract links that connect to their border nodes. Such an abstraction voluntarily limits the information disclosed to other domains, but, on the other hand, leads to inefficient resource usage. In order to enable effective collaborations between domains, in this paper, we propose to enrich the topology aggregation exposed by domains by including additional abstract topology constructs, e.g. abstract non-border nodes, domain-level slices, etc. We then revisit the virtual link embedding problem to include the proposed aggregation. Our evaluations on real and random topologies show significant gains in terms of admission ratios and resource usage.

## CCS CONCEPTS

• **Networks** → **Network resources allocation**; **Network control algorithms**; **Network design and planning algorithms**; **Programmable networks**; **Network domains**; *Logical / virtual topologies*; *Topology analysis and generation*;

## KEYWORDS

Link embedding, multi-administrative multi-domain networks, topology aggregation, network slicing, QoS



## 1 INTRODUCTION

When embedding an end-to-end service in a multi-administrative multi-domain network, the domain in charge of setting up the service solicits the cooperation of multiple domains to support portions of the service, usually with a financial reward. Classically, the goal of the service initiator is to succeed in setting up the service with the required performance at minimum costs (regarding its resource usage and monetary costs that it pays to the other domains). On the other hand, the goal of the solicited domains is to provide the most competitive offer to one or multiple portions of the end-to-end service to maximize its profits. To identify the appropriate portions compliant with the performance guarantee needs and to compose the requested service by combining them, the service initiator relies on the aggregated topologies exposed by other domains (i.e., a more or less simplified representation of the network topology with the available resources of each domain). In practice, domains restrict the topology information they expose to other domains. This aggregated topology is commonly reduced to the border nodes (Edge nodes connected to other domains), which are wholly or partially interconnected through abstract links. Some QoS (Quality of service) information may be attached to these latter, such as the maximum and available capacity, the maximum transfer delay, the maximum packet loss rate, etc. Some cost information may also be used to enforce some policy-based decisions. Below, we name this way of aggregating the network topology of a domain as the legacy aggregated /abstracted topology.

In a federation of networks, even if each domain has some individual goals, domains work together and collaborate to achieve a shared ultimate goal. This latter can be: optimally spreading the overall load on the different domains to improve the admissibility of forthcoming demands; minimizing resource usage in all domains; minimizing the end-to-end service provisioning time, etc. This is, for instance, the case in federated military networks where ally nations connect and share some of their network infrastructure to build a multi-domain Federated Mission Network (FMN) to support the communication services needed during military missions [1]. In this case, all nations share one ultimate goal, which is to efficiently use the federated resources and maximize the support of the end-to-end communication services needed during military

missions. A similar situation exists in the civilian context since service providers usually have privileged partners; their respective domains may be federated and set to work in close collaboration.
When belonging to a network federation, we argue that domains need to expose a more detailed topology abstraction than the legacy aggregation to avoid overestimation or underestimation of the available resource and, consequently, generate less efficient resource utilization, more service demand denials, etc. To that end, we propose two main additions. First, domains can expose non-border nodes, which allow them to abstract or compact some of their infrastructure topology constructs. As shown in the evaluations, their inclusion allows more precise abstractions, leading to significant improvements in service demand admissibility and a decent decrease in service demand admission delays as well as network and computing overhead. Still as part of the disclosed topology aggregation, domains can also expose domain-level slices. Indeed, as network slicing is gaining momentum, domains may support multiple domain-level slices, each providing predefined types of domain-level services. Some may be exposed and made available to other domains (with the characterization of the type of services provided by the slice and the available capacity) in order to compose their end-to-end multi-domain services. In this work, we focus on transport slices, i.e., slices providing connectivity with some predefined QoS between exposed nodes. One example of such a slice is a Low-latency slice, which interconnects a set of nodes exposed by the domain with a transfer delay of a couple of tenths of milliseconds. Another example is a slice providing efficient point-to-multipoint transmission services between a set of nodes exposed by the domain.

In this paper, we address the problem of embedding point-to-point and point-to-multipoint end-to-end virtual links with QoS requirements in a federation of networks, where network domains are allowed to expose an aggregated topology that combines border nodes and their connecting abstract links, and also, non-border abstract nodes and domain-level slices. We formulate the corresponding resource algorithm and then evaluate its performance regarding service admission and resource utilization. We also assess the benefit of the proposed topology aggregation when compared to the legacy aggregated topology.
This paper is organized as follows. Section 2 reviews the state of the art. Then, by considering an example of a real network topology, we describe the motivations behind the proposed topology aggregation and then define it. Based on this, section 4 revisits the virtual link embedding problem in a collaborative multi-domain network context. Section 5 evaluates the performance impact of the proposed topology aggregation compared to the legacy one.

## 2 STATE OF THE ART

Virtual link and virtual network embedding have been extensively researched during the last decades [11] by implicitly assuming a single administrative domain with a detailed knowledge of the substrate network. This has particularly concerned data-center networks [13]. With the advent of network slicing and Service Function Chaining (SFC), some recent research is considering the context of multi-administrative multi-domain networks [25]. In this respect, one key factor influencing the embedding technique is the data shared with others, such as domain topology and network resources linked to domains' topology abstraction. To the best of our knowledge, four main approaches were considered in the literature. First, exposing the full substrate topology with limited node and link attributes. This is the case of the work in [14] where domains expose their full topology with a limited set of static attributes, e.g. Forwarding table, geographic location and cost. Dynamic attributes such as the current available capacity or live QoS performance are not shared between domains. This approach is also adopted in other pioneering work on network virtualization in a multi-administrative context, e.g. [6].
The second approach restricts the disclosed information to a mesh of abstract links between border nodes. This approach is the most common and legacy one, which finds its roots in previous research addressing the problem of providing end-to-end communication services in ATM networks [17] and multi-domain optical networks [9]. Shen et al. [23] assume the legacy abstraction and disclose node-related localization and cost information, as well as link-related costs and inter-domain links capacities. Dietrich et al. [8] additionally expose the virtual node types that each domain supports along with the associated cost. A legacy-based domain abstraction is also considered in the context of multi-domain SFC embedding in [5, 16, 18, 26].

A third approach, known as the "Big Switch "[24], reduces the domain to a single compact abstract switch that connects to the inter-domain links. It is considered in [21]. The last and the most conservative approach is to avoid disclosing topology information. In return, the service initiator queries the domains to assess their ability to support portions of the end-to-end service, e.g., [10, 20]. Alternatively, some works propose disclosing a set of mathematical inequalities or relationships that characterize the available resources instead of any domain-related topology information [4, 19, 28].

It is worth to state that some works have also addressed the specific problem of topology abstraction or topology simplification with the motivation of improving the scalability of maintaining, processing and exchanging large network topologies. [27] classifies and compares topology summarization techniques that can be used in the context of QoS routing on the Internet. Different abstractions are considered: legacy abstraction, star abstraction, and tree abstraction, with some variants including bypass links. A comparable abstraction survey is also discussed in [4] in the context of TE (Traffic Engineered) networks. Again, three types of abstractions are distinguished: the white abstraction (full topology is exposed), the black (a.k.a Big switch abstraction) and the grey one (legacy mesh abstraction). Last, [12, 22] respectively propose graph transformation rules and methods that contract network topologies into smaller and resource (bandwidth) equivalent topologies. Some of the proposed rules ensure that any bandwidth allocation on the contracted topology is feasible with the initial topology.

In this paper, we take the opposite direction of most of the abovecited works and promote the adoption of an enriched domain abstraction. We then investigate its benefits in the context of federated

networks, where increased collaboration between domains is plausible. The domain abstraction that we propose and the associated virtual network embedding method differ in the following aspects: they allow the disclosure of non-border abstract nodes, as well as domain-level network slices. They also consider network node switching resources.

## 3 TOPOLOGY AGGREGATION IN FEDERATED MULTI-ADMINISTRATIVE NETWORKS

By considering an example of a real network, this section introduces some topology contraction rules and delves into some intuitive methods of contracting a network topology with the associated trade-offs. It also illustrates how non-border abstract nodes can be derived from the initial topology. Then, our proposed domain topology aggregation is described. Last, we present the abstracted multi-domain resource graph, which is derived from the aggregations of the topologies of the domains that compose the federation. This graph is used for the embedding of end-to-end virtual links.

### 3.1 Motivating example

Consider the topology of Figure 1, which describes the real topology of a Polish network domain taken from [2] with its physical nodes and links. The labels attached to the links represent their capacity in units of bandwidth; Link delays are omitted for clarity; The borders nodes of the domain (edge nodes) are represented in green, i.e., nodes E1, E2, and E3; Last, the red clouds correspond to clients that can be the initiator or one of the destinations of an end-to-end virtual link demand. Various aggregated topologies may be derived based on the aggregation policy employed by the domain. Below, we elaborate on an intuitive method, which leads to different aggregated topologies that can be exposed by the domain. We discuss the advantages and flaws of each obtained topology. The initial goal is to sequentially identify basic local patterns that can be reduced to a single node. This reduction ensures that any set of demands admissible in the original non-contracted subgraph is also admissible in the contracted subgraph and vice versa.

Figure 3 illustrates a first rule, which shows that any set of bandwidth demands assigned on the link (n1,n2) can also be supported on the link (E3,n1) in both directions. From a bandwidth allocation perspective, nodes E3 and n1 can be merged into an abstracted node A2. The capacity of the directional links between this abstracted node and n2 (right-hand figure) captures the bandwidth constraints on all the demands that flow between nodes E3 and n2.

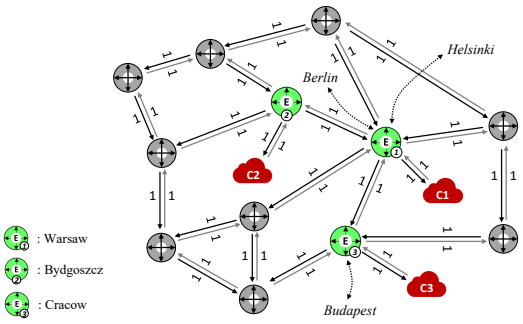


**Figure 1: Polish network domain.**

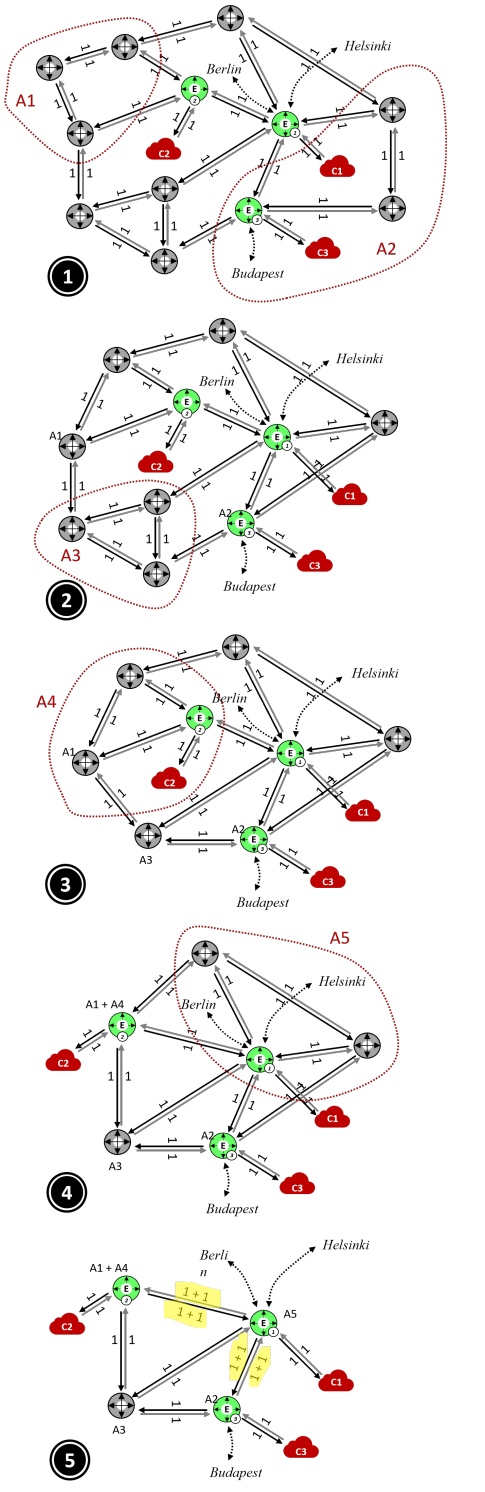


**Figure 2: Contraction rule application example.**

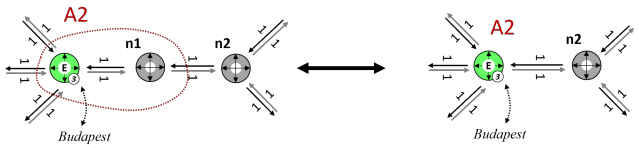


**Figure 3: Illustration of contraction rule 1.**

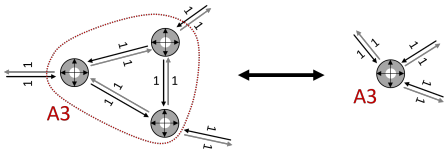


**Figure 4: Illustration of contraction rule 2.**

At the first step of the contraction process, the defined rule can be applied twice. The first time on the top left corner of the topology (highlighted by the dotted red circle A1). And a second time in the bottom right corner of the topology (highlighted by the dotted red circle A2). The result of applying this first rule to these two particular areas of the topology can be observed in the second step depicted in figure 2.

Figure 4 illustrates the second contraction rule that applies to the group of three nodes. All bandwidth demands that meet the bandwidth constraints of the external links that connect the group of three nodes to other nodes can be supported by bandwidth allocations on internal links (connecting the three nodes). From a resource allocation perspective, the three nodes can be merged into an abstract node A3. This rule is applied for the first time to the Polish topology in the red-dotted zone A3, visible in the second part of Figure 2. The outcome of this contraction can be seen in the third step of the same figure. Step 3 also emphasizes the second use of the rule (shown by zone A4). Lastly, the rule on topology is implemented for the third time. Zone A5 is depicted in part 4 in Figure 2, indicating the location of this application. It results in the contracted topology shown in Figure 2 part 5.

It is worth to state that any demand that the contracted topology can support is also admissible by the non-contracted initial topology of Figure 1 and vice versa. Also, at this point, the obtained graph includes a non-border node e.g., which is an abstracted node obtained after topology contraction. With our proposed topology aggregation, the contracted graph of Figure 5, which includes a non-border node (depicted in blue), can be disclosed by the considered domain. If the domain's policy requires the exposure of the legacy aggregated topology with only border nodes, additional contraction of the preceding graph becomes necessary. Two options can be considered:

- Exposing an over-provisioned topology : A topology that advertises more resources than what is available. In such cases, some demands may be accepted at the multi-domain level, but when they go through the domain-level admission control for confirmation, they are rejected. An over-provisioned aggregated legacy topology can be obtained by artificially increasing the link (A1, E3) capacity in both directions to 2 units. The resulting aggregated topology that the considered domain can expose is depicted in left-hand Figure 6.
- Exposing an under-provisioned topology : A topology that advertises less resources than what is available, ensuring that any demand that is admitted at the multi-domain level is admitted at the domain level. In such cases, some demands may be rejected uselessly at the multi-domain level while sufficient resources are available. An under-provisioned aggregated topology can be obtained by removing link (E1,A1) from the topology of Figure 4. The resulting aggregated topology that the considered domain can expose is depicted in right-hand Figure 6.

As summarized in table 1, the aggregation allows a significant graph size reduction. Considering the number of links and the total number of nodes, the aggregation composed of non-border nodes is 64% smaller than the original. However, despite this important reduction, the total sum of the capacities provided by the ingress and egress links of border nodes remains similar. By keeping the non-border node, the domain can share relevant network capacity information (in terms of bandwidth) without revealing its real network topology. If the non-border node is removed to stick to the legacy aggregation, the topology is smaller (9% less disclosed information than the aggregation with the non-border nodes), but this has an impact on the network capacity veracity (± 7% increase or decrease of the overall exposed bandwidth). Similar results are presented in table 1 for the french network topology depicted in Figure 15.

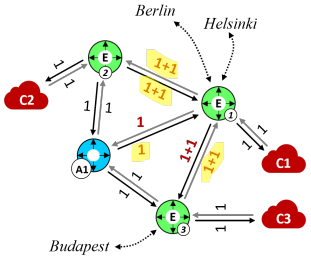


**Figure 5: Aggregated topology with non-border node.**

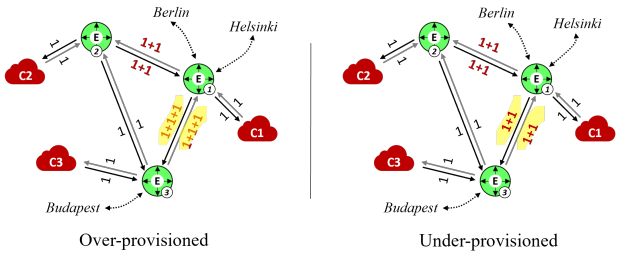


**Figure 6: Aggregated topology with over/under-provisioning.**

**Table 1: Topology contraction results for two real topologies**

| | | Total Nb of Nodes | Nb of E-nodes | Nb of non-E-nodes | Nb of End-Users | Nb of Links | Total E-nodes' Egress link cap | Total E-nodes' Ingress link cap |
|---|---|---|---|---|---|---|---|---|
| Poland | ***Initial topology*** | 12 | 3 | 9 | 3 | 42 | 14 | 14 |
| | ***Non-border nodes*** | 4 | 3 | 1 | 3 | 16 | 14 | 14 |
| | ***Over-provisioned*** | 3 | 3 | 0 | 3 | 12 | 15 | 15 |
| | ***Under-provisioned*** | 3 | 3 | 0 | 3 | 12 | 13 | 13 |
| France | ***Initial topology*** | 25 | 5 | 20 | 9 | 102 | 34 | 34 |
| | ***Non-border nodes*** | 7 | 5 | 2 | 9 | 44 | 31 | 31 |
| | ***Over-provisioned*** | 3 | 5 | 0 | 9 | 34 | 33 | 33 |
| | ***Under-prov-1*** | 5 | 5 | 0 | 9 | 32 | 26 | 26 |
| | ***Under-prov-2*** | 5 | 5 | 0 | 9 | 34 | 29 | 29 |

## 3.2 Proposed topology aggregation

The aggregated topology that a domain exposes is classically composed of a list of border nodes with a list of links connecting them. These latter are typically abstract links and correspond to one or multiple data paths (with multiple physical hops) followed by the traffic flowing from one border node to the pair node. Each has an SLA associated with it, which specifies the expected performance in terms of maximum and available capacity, maximum transfer delay, maximum packet loss rate, etc. Links may also be labeled with cost information or any other information that can be used to enforce some policy-based decisions.

The proposed aggregation paves the way for including additional (non-border) network nodes with their associated links to capture some topological specificities. Even if they are typically not physical (i.e., abstracted), non-border nodes are considered as switching nodes and share the same characteristics as border nodes. The aggregation may also include the network slices supported and made available by the domain. We consider the following attributes to characterize the exposed slices:

- Entry & Exit points: It lists the exposed nodes' interfaces that are connected to the slice, i.e., from where packets will benefit from the service provided by the slice; Slice entry or exit points are characterized by:
  - Maximum flow rate defines the maximum packet flow rate allowed when entering or leaving the slice at the considered point.
  - Access delay, when needed. It reflects that an excess delay may be needed to capture the time required to reach the slice from the entry/exit point.
- Slice type: Specifies the type of network services the slice provides. Slice types are shared and agreed upon between all domains. We assume two different families of slice types: (1) Transport slices, which provide raw network connectivity services with a predefined SLA; (2) SFC (Service Function Chaining) based slices, which provide network services that include network capabilities defined in the SFC with a predefined SLA. Without losing generality, we only consider transport slices in the mathematical formulation below.
- SLA: Expressed, as classically, in terms of max transfer delay, packet loss rate, etc.
- Maximum throughput: Specifies the maximum amount of incoming traffic (whatever the entry point) the slice can handle, i.e., the sum of the rates of all packet flows entering the slice must not exceed this capacity.
- Provisioning time: Indicates the time required to provision the slice on the infrastructure of the domain.

## 3.3 Abstracted multi-domain graph

Aggregated topologies are collected and gathered by each domain, then used as input to build their own view of the aggregated multi-domain topology, derive their multi-domain resource graph, which is then used as input for their resource allocation algorithm.

We adopt a classical representation of the abstracted multi-domain resource graph. Border and non-border nodes are represented as vertexes with switching resources expressed as a forwarding table size and a forwarding rate. We also model each exposed slice as a graph vertex that inherits the forwarding rate of the slice. Links connect the different types of vertices. They are classically characterized with classical performance metrics, e.g., delay, capacity, etc. For the links that enter or leave a slice, their bandwidth is respectively set to the maximum flow rate of the corresponding entry or exit point of the slice. The max transfer delay is reflected by the incoming links of the slice. Access delays are assigned to incoming and outgoing links of the slice. For illustration, let us reconsider the network of Figure 1 and assume that the domain is also providing a low latency slice with a max throughput of 2 units of bandwidth and a max transfer delay of 50 ms. We also assume that all border nodes are connected to the slice, and that node E3 requires an extra delay of 10 ms to reach the slice. Figure 7 depicts the resulting topology aggregation with the labels assigned as explained above (for clarity, delays of edge-to-edge links are hidden).

# 4 MATHEMATICAL FORMULATION

The abstracted multi-domain resource graph used for resource allocation does not exhibit any peculiarities that would prevent applying any algorithm devised for the single-domain case. Below, we revisit the resource allocation algorithm of [3] by including abstracted non-border node, slice node and frame switching rate. Requests arrive and are treated in sequence with no information on future requests (i.e., online). Each demand gathers multiple

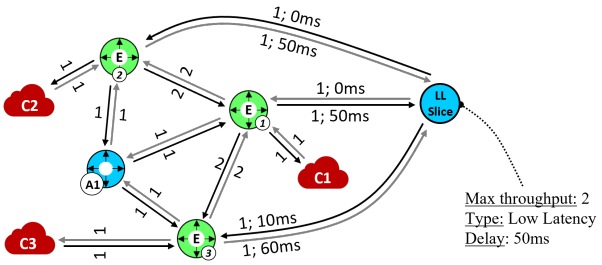


**Figure 7: Abstracted topology with abstracted slice.**

concurrent end-to-end virtual link requests concerning a point-to-point or point-to-multipoint virtual link with bandwidth and delay requirements. By default, the objective of resource allocation is to distribute network traffic fairly, use network resources efficiently, and, when needed, favor the use of slices at the expense of border node-to-border node links. Below, the multi-domain resource graph and virtual links demand models are described. Then, the variables and problem constraints are listed. Lastly, the considered objective function is defined.

## 4.1 Multi-domain resource graph model

It is a unidirectional graph $G' = (V', A')$ where $V' = N_t \cup N_s$ is the set of disclosed network nodes (border nodes + non-border nodes $\in N_t$) and disclosed network slices ($\in N_s$), and $A'$ ($|A'|, A' \subseteq V' \times V'$) is the set of exposed abstract links connecting exposed network nodes to each other or to exposed slices, and, lastly, inter-domain links. For each network node or slice $i \in V'$, we respectively denote by $\mathrm{FR}_i^{\max}$ and $\mathrm{FR}_i$ the maximum and remaining forwarding rate. For each network node $i \in N_t$, we respectively denote by $\mathrm{TE}_i^{\max}$ and $\mathrm{TE}_i$ the maximum number of flow table entries (or flow table size) and the available number of table entries. Lastly, for each link $(i, j) \in A'$, we respectively denote by $\gamma_{i,j}^{\max}$, $\gamma_{i,j}$ and $l_{i,j}$, the maximum capacity of the link, its remaining bandwidth and the maximum packet transfer delay along the link.

## 4.2 Demand model

Each demand is composed of a set of $D$ end-to-end virtual unidirectional links. Each virtual link k is characterized by:

- A source node $o_k \in N_t$, and a set of destination nodes $T_k \subseteq N_t - \{o_k\}$ (when $|T_k| = 1$, the virtual link is point-to-point; otherwise, it is point-to-multipoint).
- A bandwidth requirement of $b_k$, a maximum transfer delay of $L_k$, and a maximum packet size of $p_k$.

## 4.3 Resource-related assignment variables

The output of the resource allocation considered in this section is the set of data paths (with the bandwidth allocations at each supporting substrate link) that support each of the virtual links composing the demand. Since virtual links may be point-to-multipoint variables are related to a specific destination of a virtual link as follows:

- $\phi_{i,j}^{k,t}$: represents the bandwidth assigned to the packets of virtual link $k$ that flow from the source node $o_k$ to a destination node $t$ from $T_k$ at link $(i, j) \in A'$. Also, $\phi_{i,)}^{k}$ refers to the amount of bandwidth used on link $(i, j)$ by the virtual link $k$ regardless of the destination. It is set to the maximum of $\phi_{i,j}^{k,t}$ for all destinations $t \in T_k$.
- $x_{i,j}^{k,t}$: is a Boolean variable which reflects whether the flow of packets of virtual link $k$ destined to $t$ is supported by link $(i, j) \in A'$ ($x_{i,j}^{k,t} = 0$ if $\phi_{i,j}^{k,t} = 0$ and $x_{i,j}^{k,t} = 1$ otherwise). Variables $x_{i,j}^{k,t}$, $t \in T_k$, are used to derive the Boolean variable $x_{i,j}^{k}$, which indicates if some bandwidth from link $(i, j) \in A'$ is assigned to $k$.
- $\mathrm{TE}_i^k$ : specifies the number of entries that are installed in node $i$ forwarding table to support virtual link $k$ with the assumption that all entries consume the same amount of resources regardless of the complexity of the fetch operation and the related forwarding action to perform (forwarding with or without path splitting). The required number of entries is, in fact, dependent from the type of device (IP router, Openflow switch, etc.) Many alternatives can be considered. By default, we assume that the number of table entries needed by virtual link k equals the number of node $i$ ports that are sending traffic from $k$. More formally, $\mathrm{TE}_i^k$ is derived from the following equation (1).

$$\forall k \in D, \forall j \in V', \mathrm{TE}_i^k = \sum_{\substack{j \in V' \\ (i,j) \in A'}} x_{i,j}^k \tag{1}$$

Another option is to have the number of entries corresponding to the number of node $i$ ports that are receiving traffic from virtual link $k$ (which assumes that we have a forwarding table per input port) and is derived from equation 2.

$$\forall k \in D, \forall j \in V', \mathrm{TE}_i^k = \sum_{\substack{j \in V' \\ (j,i) \in A'}} x_{j,i}^k \tag{2}$$

A last option is to consider $\mathrm{TE}_i^k$ as a Boolean variable and add one flow table entry if at least one node $i$ port is receiving traffic from k (i.e. handling traffic from k). It is derived as shown in inequalities (3a and 3b).

$$\forall j \in E', \forall (j, i) \in A' : x_{j,i}^k \leq \mathrm{TE}_i^k \tag{3a}$$

$$\mathrm{TE}_i^k \leq \sum_{\substack{j \in V' \\ (j,i) \in A'}} x_{j,i}^k \tag{3b}$$

- $L_i^{k,t}$: represents an upper bound on the time needed for a packet of virtual link k destined to t to reach node $i$ from its originating node $o_k$.

## 4.4 Problem constraints

The constraints on bandwidth allocations are described in equations (4) to (9). Inequalities (4) reflect the linearization of the Max operator

applied to variables $\phi^{k,t}_{i,j}$ to get $\phi^{k}_{i,j}$.

$$\forall k \in D, \forall (i,j) \in A', \forall t \in T_k: \quad \phi^{k,t}_{i,j} \leq \phi^{k}_{i,j}$$
$$\forall k \in D, \forall (i,j) \in A': \quad \phi^{k}_{i,j} \leq \sum_{t \in T_k} \phi^{k,t}_{i,j} \tag{4}$$

Inequalities (5) ensure that the bandwidth assigned to each virtual link k at link $(i, j)$ does not exceed the remaining bandwidth. Equation (6) represents the usual flow conservation constraints.

$$\forall (i,j) \in A': \quad \sum_{k \in D} \phi^{k}_{i,j} \leq \gamma_{i,j} \tag{5}$$

$\forall k \in D, \forall t \in T_k, \forall i \in V'$:

$$\sum_{\substack{j \in V' \\ (i,j) \in A'}} \phi^{k,t}_{i,j} - \sum_{\substack{j \in V' \\ (j,i) \in A'}} \phi^{k,t}_{j,i} = \begin{cases} b_k & \text{if } i = o_k \\ -b_k & \text{if } i = t \\ 0 & \text{else} \end{cases} \tag{6}$$

Inequality (7) is a channeling constraint between integer and Boolean variables: $\phi^{k}_{i,j}$ and $x^{k}_{i,j}$. Additionally, they impose a constraint on the bandwidth assignment of virtual link $k$ at a substrate link, restricting it to the requested bandwidth $b_k$.

$$\forall k \in D, \forall (i,j) \in A': \quad x^{k}_{i,j} \leq \phi^{k}_{i,j} \quad \text{and} \quad \phi^{k}_{i,j} \leq b_k \cdot x^{k}_{i,j} \tag{7}$$

The constraints related to switching resource allocations are described in equations 8. They simply ensure that with the addition of flow table entries needed by the virtual links composing the demand D, the size of network nodes' forwarding tables remains below their maximum size.

$$\forall i \in V': \sum_{k \in D} \mathrm{TE}^{k}_{i} \leq \mathrm{TE}_i \tag{8}$$

As part of the aggregated topology exposed by a domain is the maximum forwarding rate of incoming packets, which holds for network nodes and slices. Inequality (9) ensures that bandwidth allocations respect these max throughputs.

$$\forall i \in V': \quad \sum_{k \in D} \sum_{(j,i) \in A'} \phi^{k}_{j,i} \leq FR_i \tag{9}$$

Virtual links have end-to-end delay requirements. For point-to-multipoint links, they must be met for all of their end destinations. These requirements are considered by inequalities (10). As path-splitting is not considered in this paper, the transfer delay that a packet from virtual link k destined to t experiences is the accumulation of the delays experienced at each hop. Each hop, through either a transport node or slice node, induces a packet transmission time, bounded by $\frac{p_k}{b_k}$ and switching and propagation delays bounded by latency $l_{i,j}$ for any link $(i, j) \in A'$. Indeed, as explained in section 3.3, the slice delay is reported on incoming links.

$$\forall k \in D, \forall t \in T_k: \sum_{(i,j) \in A'} x^{k,t}_{i,j} \cdot \left(\frac{p_k}{b_k} + l_{i,j}\right) \leq L_k \tag{10}$$

## 4.5 Objective function

The objective function aims at minimizing link and node resource consumption but also at distributing the consumed resources among nodes and links in order to reduce the creation of bottlenecks. To that end, referring to inequalities (11), variable $\varphi_{\max}$ captures the maximum link utilization (when considering all network links) after demand $D$ acceptance. $\varphi_{\max}$ being minimized by the objective function.

$$\forall (i,j) \in A': \quad 1 - \left(\frac{1}{\gamma^{\max}_{i,j}} \cdot \left(\gamma_{i,j} - \sum_{k \in D} \phi^{k}_{i,j}\right)\right) \leq \varphi^{\max} \tag{11}$$

Similarly, $\mathrm{TE}^{max}$ captures the maximum flow table utilization (when considering all network nodes) after demand D acceptance. It conforms to the constraints of inequalities 12 and is minimized by our objective function to reduce the disparity between the flow table occupancy of nodes.

$$\forall i \in V': \quad 1 - \left(\frac{1}{\mathrm{TE}^{\max}_{i}} \cdot \left(\mathrm{TE}_i - \sum_{k \in D} \mathrm{TE}^{k}_{i}\right)\right) \leq \mathrm{TE}^{\max} \tag{12}$$

Expression (13) defines our general and tunable objective function. It consists of five components, each weighted with a parameter that controls the impact of the component on the resolution process. The first two terms concern bandwidth allocations and the subsequent two others are their analogue for flow table entries allocations. We focus hereafter on the former expressions. In the first term, the aspect that is minimized is the average utilization of network links after allocating bandwidth to the considered demand D (weighted by $\alpha_1$). In the second term, the aspect that is minimized is the maximum network link utilization (resp. $\alpha_2$ then called spreading factor). This means that the allocations devoted to the request are distributed over different links (load balanced) in such a way that link load disparity is reduced. When activated ($\alpha_3 \neq 0$), the last term favors the use of slices at the expense of border node-to-border node links (with constants $c1 < c2$).

$$\begin{aligned} &\alpha_1 \cdot \frac{1}{|A'|} \cdot \sum_{(i,j) \in A'} \left(\frac{1}{\gamma^{\max}_{i,j}} \cdot (\gamma^{\max}_{i,j} - \gamma_{i,j} + \sum_{k \in D} \phi^{k}_{i,j})\right) \\ &+ \alpha_2 \cdot \varphi^{\max} \\ &+ \beta_1 \cdot \frac{1}{|V'|} \sum_{i \in V'} \left(\frac{1}{\mathrm{TE}^{\max}_{i}} \cdot (\mathrm{TE}^{\max}_{i} - \mathrm{TE}_i + \sum_{k \in D} \mathrm{TE}^{k}_{i})\right) \\ &+ \beta_2 \cdot \mathrm{TE}^{\max} \\ &+ \alpha_3 \cdot \left(c_1 \cdot \sum_{\substack{(i,j) \in A' \\ j \in N_s}} x^{k}_{i,j} + c_2 \cdot \sum_{\substack{(i,j) \in A' \\ j \in N_t}} x^{k}_{i,j}\right) \end{aligned} \tag{13}$$

# 5 PERFORMANCE EVALUATION

This section quantitatively evaluates the impact of the proposed topology aggregation on the resource allocation of multi-domain end-to-end virtual links under different topologies and service requests, as described in section 4. For clarity, we focus on optimizing bandwidth allocations. In other words, $\beta_1$ and $\beta_2$ are set to zero, which means that the optimal allocation must only satisfy the switching resources availability.

Two different aspects are primarily evaluated below:

- The benefits of allowing the disclosure of non-border network nodes with respect to the legacy topology aggregation;
- The benefits of disclosing domain-level slices compared to a multi-domain aggregation without slices.

Before, we study the general performance of the algorithm (without considering the specificities of the proposed topology aggregation) in terms of resource utilization efficiency. We particularly analyze the impact of load-distribution on the performance of our method. More precisely, we focus our evaluation on the acceptance (admission) rate of end-to-end virtual link demands, maximum and average link utilization, and the overall bandwidth allocated to a demand. The next section summarizes these evaluations.

## 5.1 Impact of load-distribution

*5.1.1 Sensitive analysis on ($\alpha_1, \alpha_2$).* As mentioned in section 4.5, the objective function of expression 13 introduces two weights ($\alpha_1$,$\alpha_2$). These latter can be tuned to either minimize the overall assigned bandwidth or to more evenly distribute the assigned resources (load) and avoid the early saturation of some crucial links.

The question of the choice of these weights is essential. During network operation, their values can be adjusted according to the load conditions; for instance, at low loads, resource minimization should be favoured, while when a load of some links start exceeding some predefined thresholds, spreading the load should be favoured. In our experiments, $\alpha_1$ and $\alpha_2$ are fixed during a simulation run. As we are focused on assessing the benefits of load-distribution (LD), we need to set a representative value of the pair ($\alpha_1$,$\alpha_2$), which will be considered for the comparison to the case where load-distribution is inhibited ($\alpha_2$=0). To determine the optimal values that should be used to maximize the acceptance rate over time, a sensitivity analysis (not detailed here for space reasons) was conducted. It shows that ($\alpha_1 = 1$, $\alpha_2 = 50$) provides the best average admission ratio computed over the different network loads. As the use of slices is not considered in this evaluation, the value of the third weight $\alpha_3$ is set to 0.

Then, the performance evaluation will be conducted with the combinations of the weights presented in table 2. C1 considers only the minimization of link utilization and, hence, corresponds to a bandwidth-constrained shortest path algorithm. Conversely, C2 provides priority to load distribution.

**Table 2: Objective function's selected weights**

| | without LD | with LD |
|---|---|---|
| $C_i$= [$\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2$] | $C_1$ = [1;0;0;0;0] | $C_2$ = [1;50;0;0;0] |

*5.1.2 Simulation set-up.* We consider a randomly generated topology composed of 100 domains and 50 sources and destinations randomly spread over the multi-domain network. Each domain is composed of 1 to 4 edge nodes with decreasing probability. We generate a random and consistent partial mesh topology at the domain and multi-domain level, where each domain has a random number of neighbours between 1 and 4 with decreasing probability. Internal and federated link capacities are randomly generated between [100,300 Mbps] and [300,400 Mbps]. Without loss of generality, service demands are composed of one single end-to-end link (which can have between 1 and 8 destinations) with bandwidth and delay requirements. We generate up to 1000 requests to ensure overload situations regardless of the generated topologies. Demands

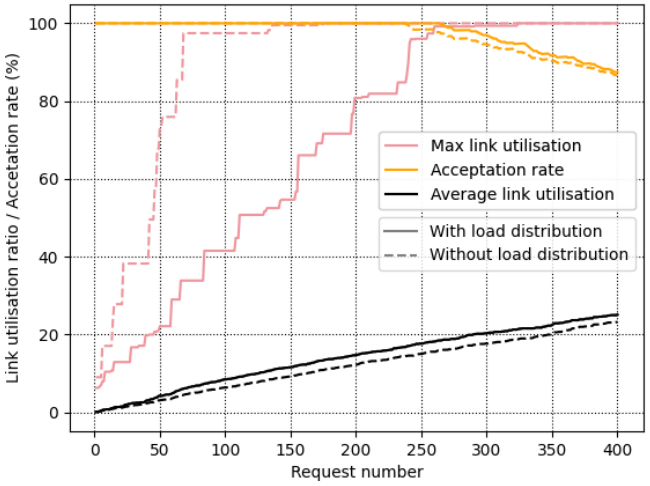


**Figure 8: Global experimentation overview comparison.**

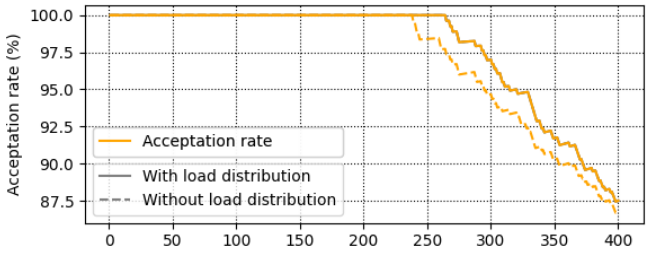


**Figure 9: Admission ratio as a function of time.**

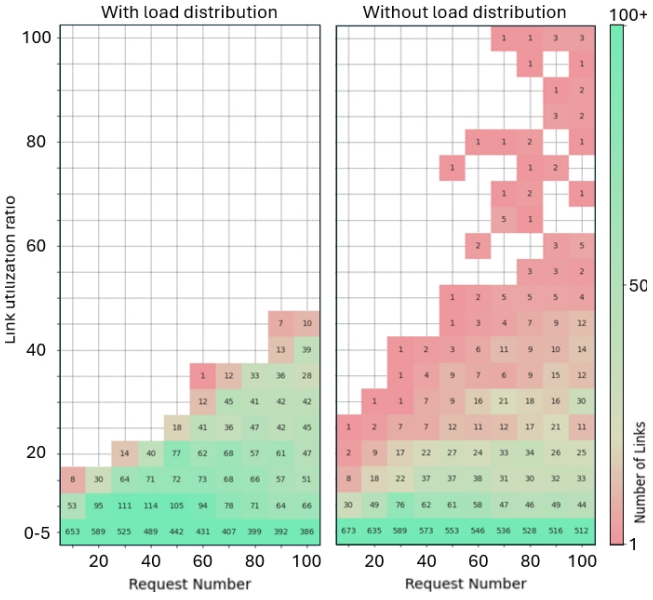


**Figure 10: Comparison of percentage link utilization.**

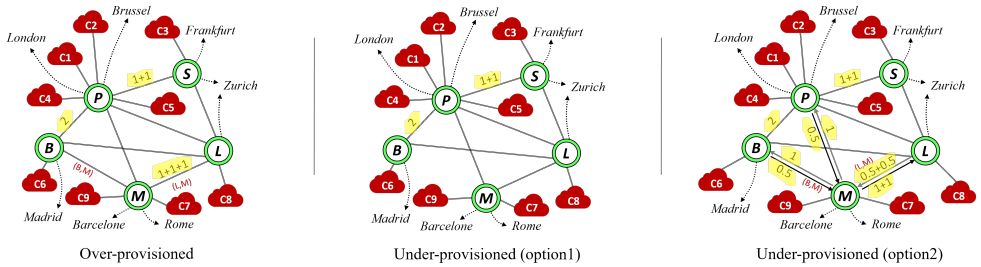


**Figure 11: Over-provisioned and under-provisioned (option 1 and 2) legacy abstraction of the French domain.**

are solved individually, and the available network capacity is updated accordingly. In this way, we can observe a gradual increase in network load and compare the implications of the different resource allocations over time. To assess the smooth operation of the algorithm and highlight the impact of load distribution, each randomly generated request is solved in two different ways: without load distribution and with load distribution (Table 2). Allocations without load distribution consist of identifying the shortest path that fulfils the required bandwidth needs (Dijkstra's algorithm with bandwidth constraints as classically used when dealing with QoS routing [7, 15]). Each resource allocation is then subtracted from the network's available resources to simulate its deployment. The network load increases gradually each time a request is solved and deployed. This process is used for all the evaluations below.

*5.1.3 Performance results.* Figure 8 describes the admission ratio as a function of time for the first 400 demands. When load-distribution is not activated (C1 configuration), some links reach saturation as early as the 50th request. When the load-distribution is activated (C2 configuration), this situation only occurs from the 260th request. This unequal use of links is detailed in Figure 10. This diagram summarizes how links' utilization ratio evolves as requests get accepted, when load-distribution is turned on (left side) or turned off (right side) for the initial 100 requests. More precisely, it shows the number of links (out of more than 700) whose utilization ratio sits within some predefined ranges (y-axis) as demands are accepted.

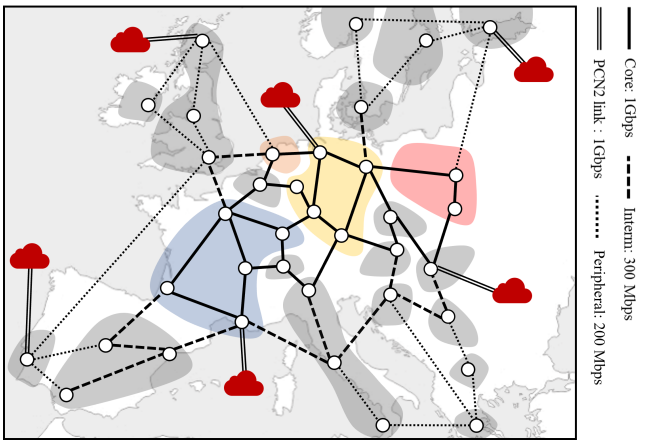


**Figure 12: COST 266 based multi-domain network.**

The positive impact of load-distribution on smoothing the link utilization ratio is easily observable, where few links suffer from a high level of utilization (most of them are below 50% of usage). It is worth to note that load-distribution does not have a noticeable effect on the average link utilization as the black curves in Figure 8 are close. Truly, occasionally distributing the load may involve taking a few longer routes; nevertheless, this leads to just a 4% increase in the average link utilization ratio. Also, this does not negatively impact the acceptance rate. Indeed, as illustrated in Figure 9, the acceptance rate is better when load-distribution is considered and starts to decrease much later. This is due to the fact that load-distribution prevents the early saturation of some crucial core links. By so doing, it allows the acceptance of demands that need to be supported by these links.

## 5.2 Evaluation of the impact of the proposed topology aggregation

*5.2.1 Simulation set-up.* To consider a realistic scenario, we consider the Cost266 topology [2] presented in Figure 12. It is composed of 22 domains from different countries. Apart from the English domain, all other domains expose exclusively border nodes with border-to-border node links. Usually, border nodes from the same domain are not connected in full mesh. The capacities of exposed links are presented in Figure 12. Out of the 22 domains, we vary the Polish and French aggregated topology by considering different legacy aggregations, in addition to the aggregation promoted in this paper. The physical topologies of the Polish and French domains are respectively presented in Figure 1 and on the left-hand Figure 15 and are taken from [2]. As the link information is unavailable, we assume all links have the same capacity of 1 Gbps. Also, we assign 9 clients to the French domain. By connecting the different versions of the Polish and French aggregated topologies to the 20 other domains, four multi-domain topologies are hence considered for the evaluation:

- Topology "proposed": The Cost266 topology with the proposed aggregated topologies of the Polish and French domains as shown in Figure 1 (Polish) and right-hand Figure 15 (French). For both domains, any demand that can be supported by each of the aggregated topologies is admissible by the corresponding physical topology;
- Topology "over-p": The Cost266 topology with the over- provisioned legacy aggregation of the Polish and French as respectively presented in left-hand Figure 6 and Figure 11;

– Topology "under-p1": The Cost266 topology with a first option of the under-provisioned legacy aggregation of the Polish and French domain, as respectively presented in right-hand Figure 6 and Figure 11. This latter is obtained by removing links (L,13) and (B,M) (right-hand Figure 15) before the final contraction;
– Topology "under-p2": The Cost266 topology with a second option of the under-provisioned legacy topology aggregation of the Polish and French domain as respectively presented in right-hand Figure 6 and Figure 11. This latter is obtained by reducing the capacity of links (L,13), (L,M), (P,13) and (B,M).

Service demands are composed of one single end-to-end link (which can have one or multiple destinations) with bandwidth and delay requirements. The number of service demands is varied in order to consider different loads (from low/medium loads up to very high loads (with 1000 demands). The source and destination of the end-to-end link are randomly generated from the set of French and Polish clients.

*5.2.2 Performance results.* Figure 13 describes the admission ratio as a function of time for the four topologies, as well as the corresponding number of admitted demands at the end of the simulation. As expected, the overprovisioned legacy topology aggregation exhibits the best admission ratio, with, at the end of the simulation, 9 more accepted demands when compared to the aggregation proposed in this paper. It is worth to note that these 9 demands are in reality not feasible and even if they are considered by the resource allocation of the multi-domain level (i.e., using the overprovisioned legacy aggregated), the resource allocation running at the domain

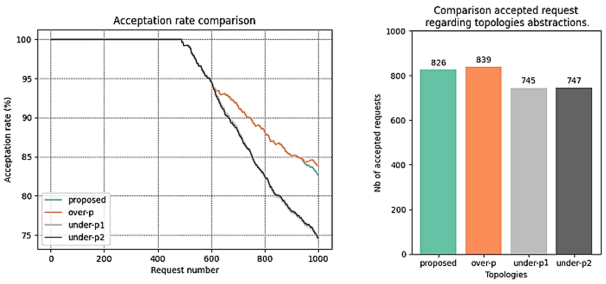


**Figure 13: Service demand acceptance comparison for different Polish and French aggregated topologies.**

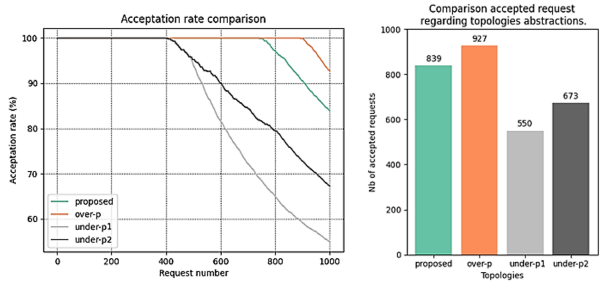


**Figure 14: Demand acceptance for different aggregated topologies, with demands from the French domain**

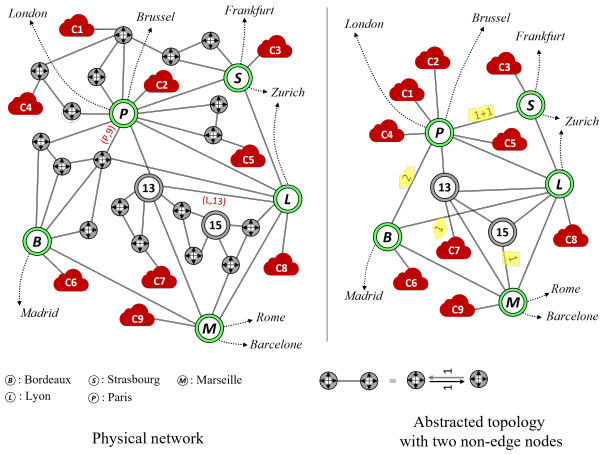


**Figure 15: Physical and aggregated French network.**

level will reject them. Consequently, a new admission must be triggered to compute a new allocation. In other words, the overprovisioned topology aggregation induces computing/network overhead as well as extra delays in addressing a service demand. These could be avoided with the proposed aggregation. Figure 13 also shows that when using an under-provisioned legacy aggregation, at the end of the simulation, more than 10% of the demands are rejected while in fact, they are feasible (as shown with the "proposed" abstraction). Hence, resources are wasted. Indeed, while available, they are not assigned to arriving demands because of an unprecise abstraction. Our simulations show that these observations are also valid when load-distribution is activated.

Next, we assume that all the demands are originating from the French domain (the nine clients associated with the French domain). The resulting performance is described in Figure 14. Compared to the previous results, the difference between the considered topology aggregations is exacerbated. More precisely, the portion of unfeasible demands accepted by the overprovisioned legacy aggregation increases from 1% to more than 10 % of all demands. On the other hand, the portion of erroneously rejected demands with under-provisioned aggregations increases from 10% to more than 30%, which is very significant. Also, for this load model, the second option of the under-provisioned aggregation exhibits better performance than the first one.

## 5.3 Evaluation of the impact of domain-level slicing

In this evaluation, we assume that some domains are implementing and exposing a domain-level slice providing point-to-multipoint (P2M) services within the boundaries of the domain and with some predefined quality of service. The rationale for exposing such slices is to allow, when possible, other domains to adopt these slices (i.e., assign slice resources from different domains) in order to build their end-to-end multi-domain P2M services, with the required QoS. We assume that if a domain exposes a P2M slice, this implicitly means that it considers that using the slice for an end-to-end P2M link is

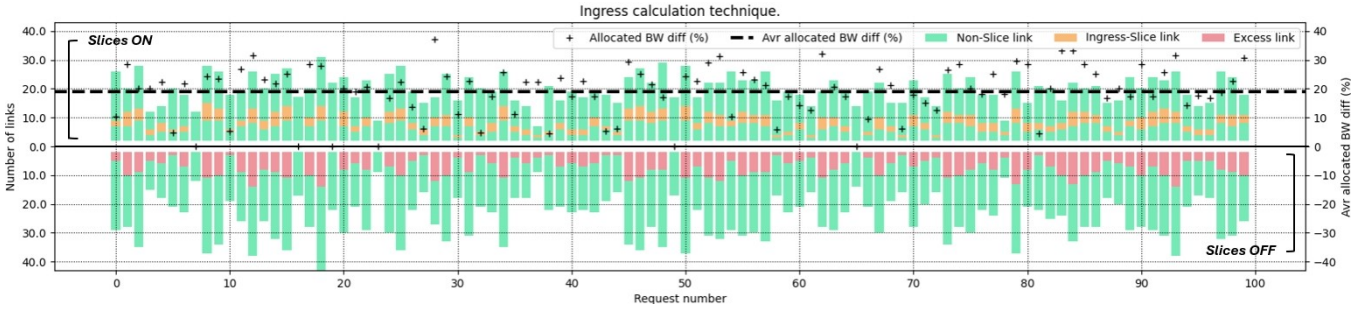


**Figure 16: Overview allocation comparison (75% domains with P2M) .**

much more effective than relying on border node-to-border node links. The objective of this evaluation is to compare the efficiency of the resource algorithm for P2M demands when P2M slices are exposed to the case where they are not.

*5.3.1 Simulation set-up.* We consider randomly generated topologies with 25 domains and 15 sources and destinations randomly spread over the multi-domain network. Each domain is composed of 3 to 7 edge nodes. A random number of non-border nodes, ranging from 0 to 3, completes the network topology of each domain. At the domain and multi-domain level, we generate a random and consistent partial mesh topology. Service demands are composed of one single P2M end-to-end link with a randomly chosen number of destinations between 2 to 8. Their bandwidth and delay requirements are randomly generated as in the previous set of experiments. The number of service demands is varied without overloading the multi-domain network as the objective is to compare the efficiency of the two options when admitting the same demands. Also, when generating the random topologies, we vary the portion of domains that are exposing a P2M slice from 25% to 100%. Also, by default, we force the resource allocation algorithm to assign slice resources only if the considered end-to-end link is duplicated when crossing the slice. This avoids the situation where slice resources are used to support point-to-point transmissions. This is achieved by defining a new set of binary variables $s_i^k, \forall i \in N_s$ and $\forall k \in D$ which specifies whether slice $i$ is being used by link $k$, more formally :

$$\forall k \in D, \forall i \in N_s, \forall (j,i) \in A' : \quad s_i^k \geq x_{j,i}^k \tag{14}$$

The above-cited constraint is enforced as follows:

$$\forall k \in D, \forall i \in N_s : \quad \sum_{(i,j) \in A'} x_{i,j}^k \geq 2 \cdot s_i^k \tag{15}$$

Finally, parameter $\alpha_1$ of the objective function is set to 0 and $\alpha_3$ is given the highest weight. This means that, whenever possible, P2M slices are favoured compared to edge-to-edge links.

*5.3.2 Performance results.* The main performance metric used to compare the efficiency of the resource allocation in the presence of exposed P2M slices is the percent decrease in the overall bandwidth that was allocated to an end-to-end link $k$ when compared to the resources that were needed with no P2M slices. More precisely, when P2M slices are exposed, the overall amount of bandwidth assigned to an end-to-end link k is computed as follows:

$$\forall k \in D : \sum_{\substack{(j,i) \in A' \\ i \in N_s}} \phi_{j,i}^k + \sum_{\substack{(i,j) \in A' \\ i,j \notin N_s}} \phi_{i,j}^k \tag{16}$$

It is worth to note that equation 16 only includes ingress slices' links. With this solution, we implicitly assume that the P2M slices consume one single transmission to reach many destinations. This assumption holds for many physical networks, notably wireless networks, including satellite communication networks. Alternatively, the number of resources could be computed by only considering slices' egress links (equation 17), which is much more resource-consuming.

$$\forall k \in D : \sum_{\substack{(i,j) \in A' \\ i \in N_s}} \phi_{i,j}^k + \sum_{\substack{(i,j) \in A' \\ i,j \notin N_s}} \phi_{i,j}^k \tag{17}$$

Finally, when P2M slices are not exposed, the overall amount of assigned resources equals: $\sum_{(i,j) \in A'} \phi_{i,j}^k$

Figure 16 illustrates the resource allocation outcomes for the initial 100 randomly generated requests, performed with a federation of networks where 75% of the domains support and expose P2M slices. Time on the X-axis represents the consecutive arrivals and acceptances of requests. Every request is solved two times: first, by prioritizing the use of slices (depicted on the top part of the figure) and then by restricting access to slices (bottom part of the figure). The figure highlights the number of links crossed and shows, for every request, the number of crossed slice ingress links and normal node-to-node links. Therefore, Figure 16 enables a straightforward comparison of allocations and highlights how the results are articulated through the network. Each bar plot represents how many links are crossed to compute the resource allocation solution. The number of crossed slice ingress links is depicted in orange. The difference in the number of crossed links between the two options (without slices minus with exposed slices, for each request) is depicted in red. Finally, the average allocation bandwidth difference between the two approaches is represented for each request by a black cross.

The results show that using P2M slices significantly reduces the number of hops (left scale). Furthermore, as we can see on the right scale, the allocated bandwidth gains fluctuate considerably, ranging

from 0% to almost 40%, with an average of 20% (black dotted line) over the first 100 requests. In fact, the gains depend on the presence of slices that can be used to duplicate the flow in order to reach multiple destinations. In other words, they are dependent of the number of destinations that compose the requested P2M virtual link as well as their locations in the network.

Figure 17 and Figure 18 present the results when P2M slices are respectively available at 25% of the domains and 75%. Each point reflects the percent decrease of assigned bandwidth for an end-to-end P2M link; links are classified according to the number of destinations (from 2 to 8 on the x-axis). The first observation is that when the portion of P2M slices is low, many demands (which are randomly generated) do not take advantage of the presence of P2M slices because of the location of their endpoints. This is particularly the case of end-to-end links with 2-3 destinations and this is normal. Indeed, the greater the number of destinations is, the more the probability of crossing a nation well located which provides a P2M slice is important. When the number of destinations increases, many more links take advantage of P2M slices. However, an average gain around 2,3% is visible. By increasing the portion of domains that host P2M slices, the number of links that benefit from P2M slices increases (Figure 18). Results are much more significant than in the previous case. For some links, the gain reaches 40% with the ingress calculation method. Even more interesting, the average benefit of crossing slices can reach 25% of gain with 8 destinations. As a general conclusion, exposing such efficient P2M slices may lead to high resource savings.

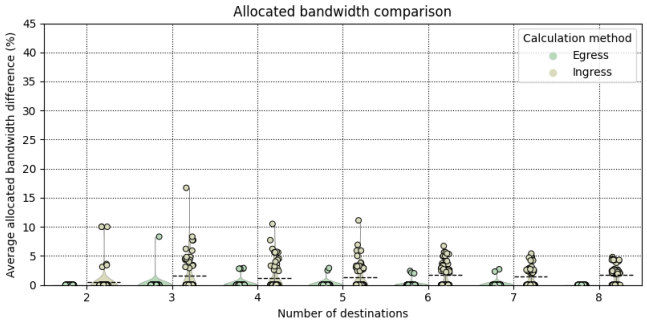


**Figure 17: Allocation comparison (25% domains with P2M).**

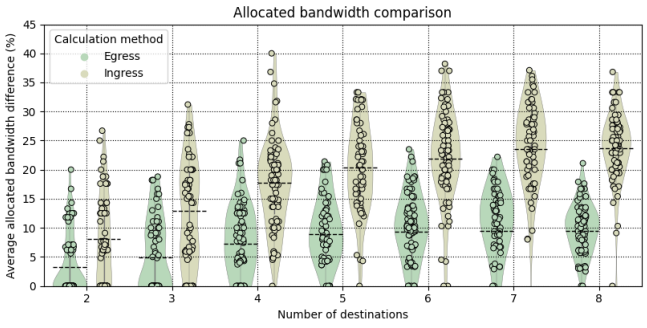


**Figure 18: Allocation comparison (75% domains with P2M).**

## 6 CONCLUSIONS

Deriving a legacy domain topology aggregation is not that trivial especially for complex and large domains. We have shown that even if the abstractions are derived with care, they may lead to a non-negligible degradation of demand admissibility (with an under-provisioned aggregation) or an increase in demand admission delays and network and computing overhead due to erroneously accepted demands at the multi-domain level. This clearly justifies the topology aggregation that we propose, which allows the inclusion of abstracted non-border nodes in order to build more precise and effective abstractions. We have also shown that introducing appropriate domain-level slices into the topology aggregation of domains offers many advantages, notably a more efficient resource usage. In addition, it leads to simplified and more scalable resource allocation methods. Perspectives of this work are considering the impact of dynamic federated multi-domain networks, with the opportunistic inclusion of new domains or, alternatively, networks and computing resources within existing domains.

## ACKNOWLEDGMENTS

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

## REFERENCES

[1] Ole Ingar Bentstuen and Petter Kristiansen. 2015. Towards a transport network for FMN based on the PCN concept. In *2015 International Conference on Military Communications and Information Systems (ICMCIS)*. IEEE, Cracow, Poland, 1–6. https://doi.org/10.1109/ICMCIS.2015.7158718

[2] Zuse-Institute Berlin. 2006. SNDlib. http://sndlib.zib.de/home.action

[3] Mikael Capelle, Slim Abdellatif, Marie-José Huguet, and Pascal Berthou. 2015. Online virtual links resource allocation in Software-Defined Networks. In *2015 IFIP Networking Conference (IFIP Networking)*. IEEE, Toulouse, France, 1–9. https://doi.org/10.1109/IFIPNetworking.2015.7145320

[4] Daniele Ceccarelli and Young Lee. 2018. Framework for Abstraction and Control of TE Networks (ACTN). RFC 8453. https://doi.org/10.17487/RFC8453

[5] Chen Chen, Lars Nagel, Lin Cui, and Fung Po Tso. 2021. Distributed federated service chaining for heterogeneous network environments. In *Proceedings of the 14th IEEE/ACM International Conference on Utility and Cloud Computing*. ACM, Leicester United Kingdom, 1–10. https://doi.org/10.1145/3468737.3494091

[6] Xiaomin Chen, André C. Drummond, Admela Jukan, and Nelson L.S. Da Fonseca. 2012. Multipath routing with topology aggregation for scalable inter-domain service provisioning in optical networks. *Optical Switching and Networking* 9, 4 (Nov. 2012), 314–322. https://doi.org/10.1016/j.osn.2012.01.002

[7] Tommy Chin, Mohamed Rahouti, and Kaiqi Xiong. 2018. Applying software-defined networking to minimize the end-to-end delay of network services. *ACM SIGAPP Applied Computing Review* 18, 1 (April 2018), 30–40. https://doi.org/10.1145/3212069.3212072

[8] David Dietrich, Amr Rizk, and Panagiotis Papadimitriou. 2015. Multi-Provider Virtual Network Embedding With Limited Information Disclosure. *IEEE Transactions on Network and Service Management* 12, 2 (June 2015), 188–201. https://doi.org/10.1109/TNSM.2015.2417652 Conference Name: IEEE Transactions on Network and Service Management.

[9] Hamza Drid, Samer Lahoud, Bernard Cousin, and Miklos Molnar. 2009. A Topology Aggregation Model for Survivability in Multi-Domain Optical Networks Using p-Cycles. In *2009 Sixth IFIP International Conference on Network and Parallel Computing*. IEEE, Gold Coast, Australia, 211–218. https://doi.org/10.1109/NPC.2009.21

[10] Ali El-Amine, Olivier Brun, Slim Abdellatif, and Pascal Berthou. 2021. 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, Madrid, Spain, 1–6. https://doi.org/10.1109/GLOBECOM46510.2021.9685356

[11] Andreas Fischer, Juan Felipe Botero, Michael Till Beck, Hermann de Meer, and Xavier Hesselbach. 2013. Virtual Network Embedding: A Survey. *IEEE Communications Surveys & Tutorials* 15, 4 (2013), 1888–1906. https://doi.org/10.1109/SURV.2013.013013.00155 Conference Name: IEEE Communications Surveys & Tutorials.

[12] Kai Gao, Qiao Xiang, Xin Wang, Yang Richard Yang, and Jun Bi. 2019. An Objective-Driven On-Demand Network Abstraction for Adaptive Applications. *IEEE/ACM Transactions on Networking* 27, 2 (April 2019), 805–818. https://doi.org/10.1109/tnet.2019.2899905 Publisher: Institute of Electrical and Electronics Engineers (IEEE).

[13] M. P. Gilesh, Sanjay Satheesh, S. D. Madhu Kumar, and Lillykutty Jacob. 2018. Selecting suitable virtual machine migrations for optimal provisioning of virtual data centers. *SIGAPP Applied Computing Review* 18, 2 (July 2018), 22–32. https://doi.org/10.1145/3243064.3243066

[14] Ines Houidi, Wajdi Louati, Walid Ben Ameur, and Djamal Zeghlache. 2011. Virtual network provisioning across multiple substrate networks. *Computer Networks* 55, 4 (March 2011), 1011–1023. https://doi.org/10.1016/j.comnet.2010.12.011

[15] Jun Huang, Qiang Duan, Qianbin Chen, Yu Sun, Yoshiaki Tanaka, and Wei Wang. 2014. Guaranteeing end-to-end quality-of-service with a generic routing approach. *ACM SIGAPP Applied Computing Review* 14 (June 2014), 8–22. https://doi.org/10.1145/2656864.2656865

[16] Godfrey Kibalya, Joan Serrat-Fernandez, Juan-Luis Gorricho, Doreen Gift Bujjingo, and Jonathan Serugunda. 2021. A Multi-Stage Graph Aided Algorithm for Distributed Service Function Chain Provisioning Across Multiple Domains. *IEEE Access* 9 (2021), 114884–114904. https://doi.org/10.1109/ACCESS.2021.3104841

[17] Whay C. Lee. 1995. Topology aggregation for hierarchical routing in ATM networks. *ACM SIGCOMM Computer Communication Review* 25, 2 (April 1995), 82–92. https://doi.org/10.1145/210613.210625 Publisher: Association for Computing Machinery (ACM).

[18] Guoyan Li, Yajuan Ren, and Yi Liu. 2021. A Cross-Domain Service Function Chain Deployment Scheme Based on Bargaining Game. *Mathematical Problems in Engineering* 2021 (Jan. 2021), 1–12. https://doi.org/10.1155/2021/6669917

[19] Rongping Lin, Song Yu, Shan Luo, Xiaoning Zhang, Jingyu Wang, and Moshe Zukerman. 2023. Column Generation Based Service Function Chaining Embedding in Multi-Domain Networks. *IEEE Transactions on Cloud Computing* 11, 1 (Jan. 2023), 185–199. https://doi.org/10.1109/TCC.2021.3084999 Conference Name: IEEE Transactions on Cloud Computing.

[20] Toru Mano, Takeru Inoue, Dai Ikarashi, Koki Hamada, Kimihiro Mizutani, and Osamu Akashi. 2014. Efficient virtual network optimization across multiple domains without revealing private information. In *2014 23rd International Conference on Computer Communication and Networks (ICCCN)*. IEEE, China, 1–8. https://doi.org/10.1109/ICCCN.2014.6911811

[21] Balazs Nemeth, Balazs Sonkoly, Matthias Rost, and Stefan Schmid. 2016. Efficient service graph embedding: A practical approach. In *2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. IEEE, Palo Alto, CA, 19–25. https://doi.org/10.1109/NFV-SDN.2016.7919470

[22] Sergey I. Nikolenko, Kirill Kogan, and Antonio Fernandez Anta. 2017. Network simplification preserving bandwidth and routing capabilities. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*. IEEE, Atlanta, GA, USA, 1–9. https://doi.org/10.1109/INFOCOM.2017.8057092

[23] Meng Shen, Ke Xu, Kun Yang, and Hsiao-Hwa Chen. 2014. Towards efficient virtual network embedding across multiple network domains. In *2014 IEEE 22nd International Symposium of Quality of Service (IWQoS)*. IEEE, Hong Kong, China, 61–70. https://doi.org/10.1109/IWQoS.2014.6914301 ISSN: 1548-615X.

[24] Balazs Sonkoly, Robert Szabo, David Jocha, Janos Czentye, Mario Kind, and Fritz-Joachim Westphal. 2015. UNIFYing Cloud and Carrier Network Resources: An Architectural View. In *2015 IEEE Global Communications Conference (GLOBECOM)*, Vol. 7. IEEE, San Diego, CA, USA, 1–7. https://doi.org/10.1109/glocom.2015.7417869

[25] Tarik Taleb, Ibrahim Afolabi, Konstantinos Samdanis, and Faqir Zarrar Yousaf. 2019. On Multi-Domain Network Slicing Orchestration Architecture and Federated Resource Control. *IEEE Network* 33, 5 (Sept. 2019), 242–252. https://doi.org/10.1109/MNET.2018.1800267 Conference Name: IEEE Network.

[26] Nassima Toumi, Djamal-Eddine Meddour, and Adlen Ksentini. 2019. A multi-objective SFC placement scheme over multiple domains. In *ICC 2019, IEEE International Conference on Communications 20-24 May 2019, Shanghai, China*, IEEE (Ed.). IEEE, Shanghai, 53–59.

[27] Suleyman Uludag, King-Shan Lui, Klara Nahrstedt, and Gregory Brewster. 2007. Analysis of Topology Aggregation techniques for QoS routing. *Comput. Surveys* 39, 3 (Sept. 2007), 7. https://doi.org/10.1145/1267070.1267071 Publisher: Association for Computing Machinery (ACM).

[28] Qiao Xiang, J. Jensen Zhang, X. Tony Wang, Y. Jace Liu, Chin Guok, Franck Le, John MacAuley, Harvey Newman, and Y. Richard Yang. 2018. Fine-Grained, Multi-Domain Network Resource Abstraction as a Fundamental Primitive to Enable High-Performance, Collaborative Data Sciences. In *Proceedings of the ACM SIGCOMM 2018 Conference on Posters and Demos*. ACM, Budapest Hungary, 27–29. https://doi.org/10.1145/3234200.3234208