Dynamic Slicing Approach for Multi-Tenant 5G Transport Networks [Invited]

Muhammad Rehan Raza, Matteo Fiorani, Ahmad Rostami, Peter Öhlen, Lena Wosinska, and Paolo Monti

***Abstract*—Software defined networking allows network providers to share their physical network (PN) among multiple tenants by means of network *slicing*, where several virtual networks (VNs) are provisioned on top of the physical one. In this scenario, PN resource utilization can be improved by introducing advanced orchestration functionalities that can intelligently assign and redistribute resources among the slices of different tenants according to the temporal variation of the VN resource requirements. This is a concept known as *dynamic slicing*. This paper presents a solution for the dynamic slicing problem in terms of both mixed integer linear programming formulations and heuristic algorithms. The benefits of dynamic slicing are compared against *static slicing*, i.e., an approach without intelligent adaptation of the amount of resources allocated to each VN. Simulation results show that dynamic slicing can reduce the VN rejection probability by more than 1 order of magnitude compared to static slicing. This can help network providers accept more VNs into their infrastructure and potentially increase their revenues. The benefits of dynamic slicing come at a cost in terms of service degradation (i.e., when not all the resources required by a VN can be provided), but the paper shows that the service degradation level introduced by the proposed solutions is very small.**

***Index Terms*—5G transport; Dynamic slicing; IP over WDM; Multi-tenant networks; Network virtualization; Software defined networking.**

I. Introduction

The fifth generation of mobile networks (5G) is expected to support a variety of services (e.g., enhanced mobile broadband, media delivery, industrial applications) with their own specific requirements in terms of, e.g., latency, capacity, and reliability. These services will be provisioned over the same physical network (PN) infrastructure equipped with a combination of radio, connectivity and cloud resources. This translates into the need for a multi-purpose, flexible, and programmable transport network with the capability to dynamically orchestrate resources in an end-to-end fashion [1,2].

Manuscript received July 12, 2017; revised October 19, 2017; accepted October 22, 2017; published December 13, 2017 (Doc. ID 302237).

M. R. Raza (e-mail: mrraza@kth.se), L. Wosinska, and P. Monti are with KTH Royal Institute of Technology, Department of Communication Systems, Electrum 229, SE-164 40 Kista, Sweden.

M. Fiorani, A. Rostami, and P. Öhlen are with Ericsson AB, Färögatan 6, SE-164 80 Kista, Sweden.

https://doi.org/10.1364/JOCN.10.000A77

Software defined networking (SDN) and network function virtualization (NFV) are promising enablers for having a programmable and flexible 5G transport network. NFV [thanks to its ability to instantiate virtualized network functions (VNFs)] provides flexibility by dynamically creating virtual network infrastructures in different segments of the transport network. Moreover, SDN introduces network programmability by allowing providers to control their resources through application programming interfaces (APIs). In addition, with SDN it is possible to share the same PN resources among different tenants through the concept of *network slicing* [3], which, in turn, allows realization of a multi-purpose transport platform.

With network slicing, several virtual networks (VNs) can be created on top of the same PN and assigned to different services (i.e., usually one VN per service). Each VN is created based on the specific needs of the corresponding service, and it is allocated a portion (i.e., a slice) of the end-to-end PN resources. The allocation of resources to each VN can be done either statically or dynamically. With *static slicing*, each VN is assigned a fixed portion of the PN resources for its entire service time. This amount of resources usually corresponds to the peak requirement of the service requesting the VN. Despite its simplicity, static slicing does not result in an efficient utilization of the PN resources. If the resource requirements of a service vary over time, a static slicing approach may result in the overprovisioning of PN resources. Therefore, a more promising solution is *dynamic slicing*. With this approach, during the *VN mapping* phase, a VN is assigned a resource slice with just enough resources to match the current service needs. In the presence of time-varying service requirements, the orchestration layer may decide (i.e., by monitoring the resource requirements of each VN) which VNs need more resources (hence a slice scales up) and which VNs do not need all the resources currently assigned to them (hence they can be momentarily scaled down). This process of resource "redistribution" among VNs is confined at the infrastructure level. Tenants are oblivious to the fact that their VNs have been scaled up/down and they still manage their resources as if they were allocated to them in a static way. This, of course, would be the ideal case. In reality, there might be instances in which the VN scaling operations are not 100% successful. As a result, some tenants will experience service degradation. This sharing of resources among different tenants is expected to improve the PN resource utilization, and potentially increases

the revenue of the network provider. Although dynamic slicing is a more flexible and resource efficient approach, it requires more complex orchestration functionalities for the dynamic reconfiguration of the PN resources. However, such mechanisms can be realized by using a SDN-based control plane.

The benefits of dynamic versus static slicing were evaluated for the first time in our previous work, presented in Ref. [4]. The dynamic slicing problem (i.e., VN mapping plus VN reconfiguration) was presented and formulated as a mixed integer linear programming (MILP) problem, although only the MILP formulation for the VN reconfiguration part of the problem was presented. In addition, the performance was evaluated using a small (i.e., with 6 nodes) network instance. Very recently, the work in Ref. [5] assessed experimentally the benefits of jointly orchestrating (i.e., among different technology domains) the scaling up and down of resource slices assigned to different VNs. However, in the demo setting, the decision to scale up/down slices was triggered by the tenants, and no orchestration functionalities in this regard were demonstrated.

This paper builds on the initial contribution proposed in Ref. [4] and presents a comprehensive assessment of the benefits of dynamic slicing in a 5G transport network. More precisely, the paper proposes MILP formulations for the complete dynamic slicing problem (i.e., VN mapping plus VN reconfiguration). In addition, the paper presents a solution for the dynamic slicing problem based on heuristic algorithms in order to evaluate the performance of dynamic slicing using large network instances. After validating the performance of the proposed heuristic algorithms in a small network scenario, the benefits of dynamic slicing are evaluated using a more realistic 38-node transport network. The simulation results show that with dynamic slicing, it is possible to achieve more than 1 order of magnitude better resource utilization than with a static slicing approach. It was also found that the level of service degradation is very low, and within acceptable levels for both the network providers and the tenants.

II. Related Work

Much research has been carried out on network slicing, especially in Internet Protocol (IP)-based networks, where most of the literature is focused on how to map a VN request into a PN, e.g., [6–8]. As of now, to the best of our knowledge, there is no universal definition of network slicing. In our view, a slice is a set of resources (either physical or virtual) assigned to a tenant by the network provider that, in turn, owns the corresponding physical resources. When a new tenant asks for the control over a set of resources, it queries the network provider with a description of what is needed. The network provider, in turn, tries to meet these requirements by checking the status of its physical resources. This process of checking whether the requirements can be met is equivalent to solving the VN mapping/embedding problem, where a VN request describes the set of resources required by a tenant. In this respect, the provisioning of a network slice and the VN mapping/embedding problem are very similar problems. On the other hand, a dynamic slicing solution entitles more than just solving a VN mapping/reconfiguration problem. With dynamic slicing, tenants are provided with programmatic APIs to control the provisioned slice(s). In addition to this, dynamic slicing includes a number of enhancements from the architectural and functional point of view that enable the orchestrator of a network provider to intelligently assign and redistribute resources among the slices of different tenants in such a way that *(i)* the infrastructure resources are used more efficiently, and *(ii)* the degradation level experienced by each tenant is contained to acceptable levels.

Network slicing in optical transport networks is still a hot research topic. Most of the works in this area consider the design problem, i.e., all the VN requests are assumed to be known in advance, and the objective is to dimension a PN capable of embedding all of them using as few PN resources as possible [9,10]. The authors of Ref. [9] propose a MILP formulation and a heuristic algorithm for VN mapping over both wavelength-division multiplexing (WDM) and flexible-grid optical PNs. The work in Ref. [10] presents a strategy for the mapping of VNs over an opaque optical PN, where the optical signal can be terminated and converted to the electrical domain at any intermediate node. As a result, for a given VN, there is no need to allocate the same (set of) wavelength(s) for each virtual link.

The problem of network slicing in the presence of dynamic VN mapping requests has recently gained attention from standardization bodies, vendors, and the research community in general. Most of the recent studies in this direction consider a scenario in which the VN requests to be provisioned are not known *a priori* and they arrive/leave the network dynamically. In this respect, the work in Ref. [11] introduces the concept of "Slice-as-a-Service," showing how this feature is crucial in most of the 5G use cases. The authors of Refs. [12,13] propose algorithms for dynamic VN provisioning in different scenarios. The work in Ref. [12] studies how to dynamically serve VN requests over an orthogonal frequency-division multiplexing (OFDM)-based optical PN. A heuristic algorithm for dynamic VN mapping in multilayer optical PN is also proposed in Ref. [13].

On the other hand, the works in Refs. [14,15] present an experimental demonstration of the dynamic VN provisioning concept. The authors of Ref. [14] experimentally demonstrate a multi-tenant network architecture that dynamically provides a slice whose resources are under full control of the tenant requesting it. The work in Ref. [15] proposes a multi-domain resource broker able to dynamically provision virtual optical networks across heterogeneous control domains [i.e., generalized multi-protocol label switching (GMPLS) and OpenFlow] and transport technologies (i.e., optical packet switching and elastic optical networks).

The above-mentioned works do not account for the possibility to change the mapping of a VN after it has been provisioned into the PN, i.e., the VN reconfiguration concept. On the other hand, the works in Refs. [16–18]

highlight the need for VN reconfiguration in order to efficiently utilize the PN resources, i.e., VNs might need to be reconfigured either after a new VN request cannot be provisioned (i.e., because of lack of resources in the PN), or after a failure occurs in the PN. The work in Ref. [16] studies the benefits of VN reconfiguration in the case of elastic optical networks when new VN requests are blocked due to lack of sub-carriers in fiber links. The authors of Ref. [17] propose a VN reconfiguration algorithm that minimizes the number of virtual node migrations to minimize disruptions in the traffic carried by the VN. Moreover, the work in Ref. [18] presents an experimental demonstration of dynamic virtual optical network mapping and reconfiguration, where the latter is performed after node or link failures in the PN.

The authors of Ref. [19] present MILP formulations and heuristic algorithms for on-demand reconfiguration of VNs of mobile operators that provide centralized radio access network (C-RAN) services. In this work, upon request from a tenant, the network provider tries to reallocate the VNs assigned to that specific tenant. VNs are reconfigured by leveraging the spare resources available in the PN at that particular point in time. In this process, the orchestrator does not have any functionality to free resources from other tenants (that at the moment might not need them) to further improve the infrastructure utilization. On the other hand, in the architecture envisioned in this paper, the decision to scale up/down the slice of a tenant is taken by the orchestrator, which, in turn, is also able to redistribute resources among tenants as needed to improve the infrastructure resource efficiency. The work in Ref. [20] is a proof of concept of an architecture that works in a similar way to the one envisioned in our paper. In Ref. [20], the orchestrator (and not a tenant) makes the decision to reconfigure a slice (either because of the presence of congestion or of a failure in the PN). On the other hand, one of the main features of the proposed solution, i.e., the possibility to redistribute resources among slices of different tenants during the reconfiguration process, is not considered. Finally, the work in Ref. [21] highlights the importance of thoroughly investigating the benefits of dynamic slicing as opposed to conventional (i.e., static) approaches (i.e., without dynamic scaling up/down of the slices). Such an investigation is missing in the existing literature, and this paper is the first one that evaluates the benefits of dynamic slicing. More specifically, the objective is to assess whether the benefits brought by dynamic slicing are significant enough to justify the necessary increase in control plane complexity. In particular, this work focuses on strategies for provisioning and maintaining network slices considering *(i)* dynamic arrival/departure of tenants' VN requests, and *(ii)* temporal variation of VN resource requirements.

III. System Architecture

Figure 1 presents the system architecture considered in this paper, which consists of three interconnected technology domains. The first domain is an IP-over-WDM transport network [22], which provides connectivity services to the other two domains. This transport network is

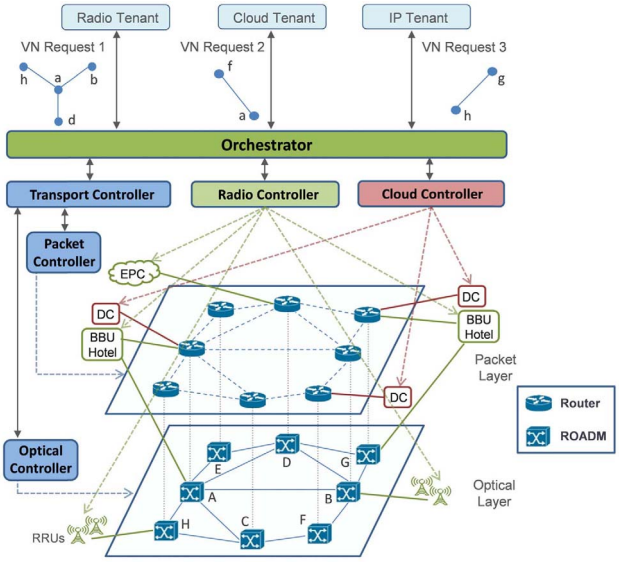

Fig. 1. System architecture. DC, data center; RRU, remote radio unit; BBU, baseband unit; EPC, evolved packet core; VN, virtual network; ROADM, reconfigurable optical add–drop multiplexer.

composed of two layers, i.e., packet and optical, each one managed by its respective controller. The packet layer comprises IP routers, while the optical layer consists of reconfigurable optical add–drop multiplexers (ROADMs) connected by optical fiber links. It is assumed that each IP router is connected with a ROADM to offer traffic grooming [23] capability at each node. The transport controller manages the operations in the packet and optical layers by jointly interacting with the packet and optical controllers.

The radio domain provides mobile broadband services using a number of long-term evolution (LTE) access points. They are deployed according to the C-RAN concept [24], i.e., the RAN functionalities are split between remote radio units (RRUs) and baseband processing units (BBUs) placed at BBU hotels and connected to the RRUs using the common public radio interface (CPRI). The radio domain relies on the connectivity services of the transport network for provisioning of CPRI flows. Moreover, the radio controller is in charge of performing RAN functions, which include *(i)* traffic monitoring at the RRUs, *(ii)* their activation/configuration, *(iii)* the monitoring of the BBU resources, and *(iv)* how BBUs are assigned to RRUs.

The cloud domain comprises a number of datacenters (DCs), which provide compute/storage services. The cloud controller is responsible for the management of the DC resources and for operating the intra-DC network in each DC. The cloud domain is also dependent on the transport network for interconnecting DCs, e.g., for creating replicas or for synchronization among DCs.

In the envisioned architecture, the network provider is in charge of *(i)* slice provisioning and *(ii)* slice reconfiguration decisions. This is done with the help of an orchestration layer that performs not only a cross-domain management of radio (i.e., BBU hotels), transport

(i.e., packet/optical), and cloud (compute/storage) resources,[1] but also life-cycle orchestration of services. This latter aspect is outside the scope of this paper. The orchestrator also aggregates the information about the PN resources into a unified and abstracted representation, which is then directly exposed (i.e., using the orchestrator's north-bound interface) toward the tenant applications. In turn, tenants can ask for the provisioning of resource slices. This is done by issuing VN requests, which includes *(i)* topology information (i.e., nodes, links), *(ii)* the requirements for the resources to be provisioned over the virtual links and in the virtual nodes, and *(iii)* some tenant-specific constraints (e.g., a virtual node might need to be mapped to a certain PN node). The orchestrator tries to assign resources in the underlying PN to each VN request, and based on their availability, notifies the tenant whether or not the operation has been successful.

Three different types of tenant are considered in this work, i.e., *radio*, *cloud*, and *IP*. The radio tenant asks for connectivity (i.e., capacity) between RRUs and the evolved packet core (EPC) while traversing a BBU hotel with enough ports available. The cloud tenant requests the allocation of cloud resources (i.e., compute and storage resources) in some DCs, which in turn need to be able to communicate among themselves for data exchange. The IP tenant asks for a certain amount of bandwidth to be provisioned between Internet service provider (ISP) access points.

The radio and IP tenants are assumed to have peak resource requirements during the daytime, i.e., their users are more active during the day. On the other hand, cloud tenants are assumed to experience their peaks during the nighttime, e.g., for synchronization or replication among the DCs. This temporal variation in resource requirements of different tenants can be exploited by the network provider to achieve efficient utilization of PN resources. The amount of resources assigned to the VN of each tenant can be dynamically adjusted according to the variation of the VN resource requirements, i.e., a concept referred to as dynamic slicing. The next section explains in more detail how this concept works.

IV. Concept of Dynamic Slicing

Figure 2 shows an example of the temporal variations of the resource requirements of two VN requests coming from two different tenants. The VN request of tenant 1 requires more PN resources during the day, while the one from tenant 2 needs more resources during the night. A conventional slicing approach would allocate a fixed amount of resources (i.e., corresponding to the peak requirement) to each VN for its entire duration. However, such a scheme would result in poor utilization of PN resources.

[1]The slice provisioning and reconfiguration decisions made by the orchestrator are based on the information about available resources collected from the respective domain controllers. Our envisioned dynamic slicing policy works as long as the corresponding controllers provide the necessary resource information, and it is not limited by the number of technological domains managed by the orchestrator.

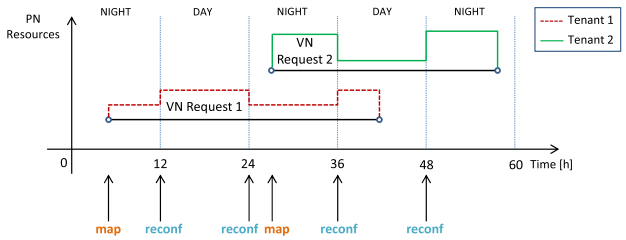

Fig. 2. Dynamic slicing considering VN requests from two different tenants.

A more efficient approach is dynamic slicing, where the size of the slice assigned to each tenant may be varied according to the changes in the VN resource requirements. With dynamic slicing, during any given time period (i.e., day or night), the orchestrator may decide to scale down the slices of those tenants whose VNs do not need all the resources currently allocated to them. At the same time, these newly released resources can be allocated to those VNs that require more. Thanks to dynamic slicing, it becomes possible to share PN resources among different tenants and consequently improve the efficiency in which the overall PN is used.

The problem of dynamic slicing addressed in this paper consists of two parts (Fig. 2): *(i) VN mapping*, when a new VN request needs to be provisioned, and *(ii) VN reconfiguration*, when all the VNs that experience a change in their resource requirements (i.e., during a day/night variation or vice versa) need to have their slices scaled up/down.

The input of the VN mapping phase is described in terms of the VN resource requirements. The objective is to find a set of PN resources (link capacity and compute/storage units) that satisfies these requirements. To minimize the amount of connectivity resources allocated to provision a given VN, the proposed mapping strategy tries to allocate new VNs onto already established lightpaths. If it is not possible, the strategy tries to add a minimum number of new lightpaths to support the VN connectivity requirements. If the PN does not have enough resources to meet the requirements of a new VN request, it is rejected. It is assumed that when a new VN request is rejected, it is not possible to map it at a later time when the PN resources might become available.

The VN reconfiguration phase tries to adapt the mapping of all the provisioned VNs to match their time-varying resource requirements. Resources are released for those VNs needing fewer resources than the ones currently assigned. In turn, these newly freed resources are allocated to those VNs that experience an increase in their resource requirements. The objective of the VN reconfiguration phase is to match as close as possible the variations of VN resource requirements, while limiting as much as possible the number of lightpath reconfigurations, i.e., lightpath re-routing operations, in order to not interrupt the services running over the provisioned VNs. This work assumes that only the virtual link capacity requirements of the VN change over time.

During the VN reconfiguration phase, if the resource slice of a VN needs scaling up, it might happen that not all the required resources can be provided (e.g., if the network is in high load conditions). In this case, the service provided to the tenant becomes degraded. Assuming that a VN request with a service time of T time units is accepted at time t_1, the total VN degradation D is given by the following formula:

$$D = \frac{\int_{t_1}^{t_1+T} C_{\text{req}}(t) - \int_{t_1}^{t_1+T} C_{\text{prov}}(t)}{\int_{t_1}^{t_1+T} C_{\text{req}}(t)}, \quad (1)$$

where $C_{\text{req}}(t)$ denotes the total required capacity and $C_{\text{prov}}(t)$ represents the total provided capacity at time t in all the links of the VN (Fig. 3). The next section describes the MILP formulations for the dynamic slicing procedure.

V. MILP Formulations for Dynamic Slicing

This section presents the MILP formulations for the VN mapping and VN reconfiguration phases described in the previous section.

A. $MILP_{map}$

This formulation is derived from the work presented in Refs. [9,25]. MILP_{map} performs the mapping of a VN request into the PN, without removing or modifying the mapping of any of the VNs already provisioned into the PN. MILP_{map} takes into account the routes and the capacity used by the lightpaths currently established in the PN. The capacity information helps MILP_{map} to decide whether the capacity requirements of an incoming VN request can be met by leveraging onto any previously established lightpath. Otherwise, MILP_{map} adds a minimum number of new lightpaths necessary to accommodate the VN request.

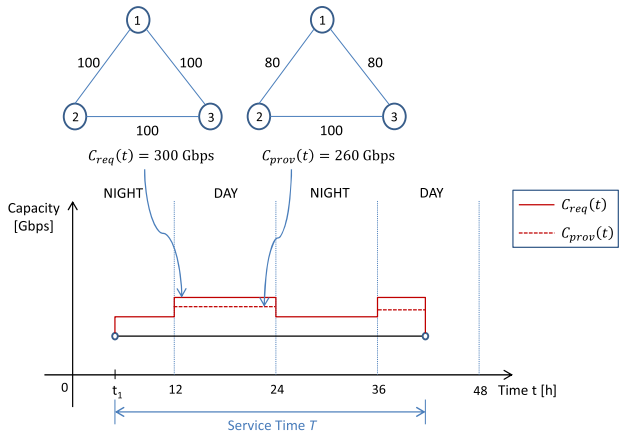

Fig. 3. Total capacity required and provided to a VN with service time T. $C_{\text{req}}(t)$, total required capacity; $C_{\text{prov}}(t)$, total provided capacity; t_1, arrival time of the VN.

The objective function of MILP_{map} is presented in Eq. (2). It aims at minimizing the wavelength resource usage in the PN. Constraint (3) makes sure that the compute/storage capacity required by the cloud tenant does not exceed what is available in the chosen DCs. Constraint (4) ensures that the selected BBU hotel for the radio tenant has enough BBU ports available. Constraint (5) makes sure that a virtual node is mapped to one PN node only. Constraint (6) ensures that two different virtual nodes in the same VN are not mapped to the same PN node. Constraints (7) and (8) compute the total capacity that needs to be provisioned between two nodes in the PN. Constraint (9) ensures flow conservation in the IP layer. Constraint (10) computes the total capacity used on each lightpath after mapping the incoming VN request. Constraint (11) makes sure that the capacity provisioned between two nodes is not exceeding the one provided by the lightpath(s) between them. Constraint (12) ensures flow conservation in the optical layer. Constraint (13) makes sure that the number of established lightpaths on a fiber link is lower than the available wavelengths. Moreover, it is worth mentioning that the wavelength continuity constraint is relaxed, i.e., it is assumed that each PN node is equipped with enough transponders (TPs) to perform as many electrical-to-optical and optical-to-electrical conversions as needed. Constraint (14) ensures that the previously established lightpaths cannot be removed or rerouted to accommodate the incoming VN request.

If MILP_{map} results in an infeasible solution, the VN request is rejected. Otherwise, MILP_{map} generates the following output: *(i)* node mapping of the VN request, i.e., where each virtual node is mapped in the PN; *(ii)* link mapping of the VN request, i.e., how the capacity requirement of each virtual link is mapped into the lightpaths provisioned in the PN; and *(iii)* the routes and the capacity used by all the lightpaths in the PN. The MILP_{map} output information is then used to update the description of the current status of the PN resources. Once the VN service time expires, the corresponding resources in the PN are released and the PN resource usage information is updated. If, after the VN departure, there is no traffic left on a lightpath, the lightpath is torn down.

B. $MILP_{reconf}$

As mentioned earlier, when the link capacity requirements of a VN change over time, its slice needs to be adapted accordingly. Therefore, all the currently mapped VNs need to be reconfigured to match their new requirements. Some VNs might need their slice to be scaled up, while other VNs might require fewer resources. $\text{MILP}_{\text{reconf}}$ is proposed to find an optimal way for scaling up/down the resource slices of all the existing VNs. In the case of an increase in the capacity requirement of a VN, $\text{MILP}_{\text{reconf}}$ first tries to re-size the existing VN mapping. If this is not possible, it tries to re-map the VN over the lightpaths already existing in the PN, or over newly established ones. If both re-sizing and re-mapping are not successful, the VN is degraded and the degradation value is computed using Eq. (1).

The objective function of $\text{MILP}_{\text{reconf}}$ is presented in Eq. (15). It aims at minimizing *(i)* the degradation of VNs currently mapped in the PN, *(ii)* the number of lightpath reconfigurations, and *(iii)* the wavelength resource usage. α, β, and γ are weighting factors, with $\alpha \gg \beta \gg \gamma$. Constraints (16) and (17) compute the total capacity required between two nodes in the PN, considering all the VNs currently mapped into the PN at the time of reconfiguration. Constraint (18) ensures flow conservation in the IP layer, considering that some VNs might be degraded. Constraint (19) computes the total capacity used on each lightpath after reconfiguration. Constraints (20)–(22) are used to compute the number of lightpath reconfigurations. Note that Constraints (21) and (22) can be linearized by using a set of simple linear constraints, as described in Appendix A.

Parameters

V	Set of all VNs currently mapped in the PN
N_s	Set of nodes in the PN
N_v	Set of virtual nodes in the incoming VN request v
Nb_m	Set of neighboring nodes of node m in the PN
SD_v	Set of source-destination pairs of VN v
CN_a	Set of all candidate nodes of virtual node a
ρ_{sd}	Capacity requirement for virtual link s-d
ρ^v_{sd}	Capacity requirement for virtual link s-d of VN v
dc_a	Compute/storage capacity required by virtual node a
bbu_a	No. of BBU ports required by virtual node a
DC_k	Current compute/storage capacity available at PN node k
BBU_k	Current no. of available BBU ports at PN node k
tb_{ij}	Capacity used on lightpaths between PN nodes i and j before invoking MILP_{map}
q^{ij}_{mn}	No. of lightpaths between PN nodes i and j passing through link (m,n) before invoking MILP
r^v_{sb}	Node mapping (i.e., virtual node s is mapped to PN node b) of VN v, used in $\text{MILP}_{\text{reconf}}$
W	No. of wavelengths available in each link in the PN
C	Capacity of each wavelength

Variables

y_{ak}	1 if virtual node a is mapped to PN node k
x_{ij}	No. of lightpaths between PN nodes i and j
ta_{ij}	Capacity used on lightpaths between PN nodes i and j after solving MILP
z^{ij}_{mn}	No. of lightpaths between PN nodes i and j passing through link (m,n) after solving MILP
zq^{ij}_{mn}	Difference between z^{ij}_{mn} and q^{ij}_{mn}
p^{ij}_{mn}	1 if z^{ij}_{mn} is greater than or equal to q^{ij}_{mn}
u_{ij}	No. of reconfigurations of lightpaths between PN nodes i and j
g^{sd}_{be}	XOR of y_{sb} and y_{de} (i.e., 1 if y_{sb} and y_{de} are different, 0 otherwise)
$g^{v,sd}_{be}$	XOR of r^v_{sb} and r^v_{de}
h^{sd}_{be}	1 if end points of virtual link s-d are mapped to PN nodes b-e
$h^{v,sd}_{be}$	1 if end points of virtual link s-d of VN v are mapped to PN nodes b-e
d^v_{be}	Degradation of virtual link of VN v whose source and destination are mapped to PN nodes b-e
l_{be}	Capacity that needs to be provisioned from PN node b to e
l^v_{be}	Capacity that needs to be provisioned from PN node b to e corresponding to VN v
f^{be}_{ij}	Capacity provisioned from PN node b to e through lightpaths i-j
$f^{v,be}_{ij}$	Capacity provisioned from PN node b to e through lightpaths i-j for VN v

MILP_{map}

Objective:

$$\min\left(\sum_{\substack{i,j\in N_s\\ i\neq j}}\sum_{m\in N_s, n\in Nb_m} z^{ij}_{mn}\right). \tag{2}$$

Constraints:

$$\sum_{a\in N_v} y_{ak}\times dc_a \le \text{DC}_k, \quad \forall\, k\in N_s, \tag{3}$$

$$\sum_{a\in N_v} y_{ak}\times \text{bbu}_a \le \text{BBU}_k, \quad \forall\, k\in N_s, \tag{4}$$

$$\sum_{k\in \text{CN}_a} y_{ak} = 1, \quad \forall\, a\in N_v, \tag{5}$$

$$\sum_{a\in N_v} y_{ak} \le 1, \quad \forall k\in N_s, \tag{6}$$

$$y_{sb}+y_{de} = g^{sd}_{be} + 2\cdot h^{sd}_{be}, \quad \forall\,(s,d)\in \text{SD}_v, \quad \forall\, b,e\in N_s: b\neq e, \tag{7}$$

$$\sum_{(s,d)\in \text{SD}_v} (\rho_{sd}\times h^{sd}_{be}) = l_{be}, \quad \forall\, b,e\in N_s: b\neq e, \tag{8}$$

$$\sum_{\substack{j\in N_s\\ j\neq i}} f^{be}_{ij} - \sum_{\substack{j\in N_s\\ j\neq i}} f^{be}_{ji} = \begin{cases} l_{be}, & \text{if } i=b\\ -l_{be}, & \text{if } i=e\\ 0, & \text{otherwise}\end{cases}, \quad \forall\, b,e,i\in N_s: b\neq e, \tag{9}$$

$$\sum_{\substack{b,e\in N_s\\ b\neq e}} f^{be}_{ij} + tb_{ij} = ta_{ij}, \quad \forall\, i,j\in N_s: i\neq j, \tag{10}$$

$$ta_{ij} \le C\times x_{ij}, \quad \forall\, i,j\in N_s: i\neq j, \tag{11}$$

$$\sum_{n\in Nb_m} z^{ij}_{mn} - \sum_{n\in Nb_m} z^{ij}_{nm} = \begin{cases} x_{ij}, & \text{if } m=i\\ -x_{ij}, & \text{if } m=j\\ 0, & \text{otherwise}\end{cases}, \quad \forall\, i,j,m\in N_s: i\neq j, \tag{12}$$

$$\sum_{\substack{i,j\in N_s\\ i\neq j}}(z_{mn}^{ij}+z_{nm}^{ij})\le W,\quad \forall\, m\in N_s, n\in Nb_m, \tag{13}$$

$$z_{mn}^{ij}\ge q_{mn}^{ij},\quad \forall\, i,j\in N_s\colon i\neq j,\quad \forall m\in N_s, n\in Nb_m. \tag{14}$$

MILP$_{\text{reconf}}$

Objective:

$$\min\left(\alpha\sum_{v\in V}\sum_{\substack{b,e\in N_s\\ b\neq e}}d_{be}^{v}+\beta\sum_{\substack{i,j\in N_s\\ i\neq j}}u_{ij}+\gamma\sum_{\substack{i,j\in N_s\\ i\neq j}}\sum_{m\in N_s,n\in Nb_m}z_{mn}^{ij}\right). \tag{15}$$

Constraints:

$$r_{sb}^{v}+r_{de}^{v}=g_{be}^{v,sd}+2\cdot h_{be}^{v,sd},\quad \forall v\in V,\quad \forall (s,d)\in \mathrm{SD}_v,\quad \forall b,e\in N_s\colon b\neq e, \tag{16}$$

$$\sum_{(s,d)\in \mathrm{SD}_v}(\rho_{sd}^{v}\times h_{be}^{v,sd})=l_{be}^{v},\quad \forall v\in V,\quad \forall b,e\in N_s\colon b\neq e, \tag{17}$$

$$\sum_{\substack{j\in N_s\\ j\neq i}}f_{ij}^{v,be}-\sum_{\substack{j\in N_s\\ j\neq i}}f_{ji}^{v,be}=\begin{cases}(l_{be}^{v}-d_{be}^{v}), & \text{if } i=b\\ -(l_{be}^{v}-d_{be}^{v}), & \text{if } i=e\\ 0, & \text{otherwise}\end{cases},\quad \forall v\in V,\quad \forall\, b,e,i\in N_s\colon b\neq e, \tag{18}$$

$$\sum_{v\in V}\sum_{\substack{b,e\in N_s\\ b\neq e}}f_{ij}^{v,be}=ta_{ij},\quad \forall\, i,j\in N_s\colon i\neq j. \tag{19}$$

Constraints (11), (12), *and* (13) *in MILP$_{map}$*

$$zq_{mn}^{ij}=z_{mn}^{ij}-q_{mn}^{ij},\quad \forall\, i,j\in N_s\colon i\neq j,\quad \forall\, m\in N_s, n\in Nb_m. \tag{20}$$

$$p_{mn}^{ij}=\begin{cases}0, & \text{if } zq_{mn}^{ij}<0\\ 1, & \text{if } zq_{mn}^{ij}\ge 0\end{cases},\quad \forall i,j\in N_s\colon i\neq j,\quad \forall m\in N_s, n\in Nb_m \tag{21}$$

$$u_{ij}=\sum_{m\in N_s,n\in Nb_m}zq_{mn}^{ij}\times p_{mn}^{ij},\quad \forall\, i,j\in N_s\colon i\neq j. \tag{22}$$

MILP$_{\text{reconf}}$ always results in a feasible solution, with the following output: *(i)* link mapping of the VNs, i.e., how the capacity of each virtual link is mapped into the lightpaths provisioned in the PN, *(ii)* the routes and capacity used by all the lightpaths, and *(iii)* the degradation (if any) of the virtual links of each VN. The latter output helps to compute the total degradation value of each VN, i.e., using Eq. (1).

Although the MILP formulations presented so far guarantee optimal solutions for both the VN mapping and the VN reconfiguration phases, they might require a long execution time and they might not be suitable to be run with large network instances. Therefore, the next section presents two heuristic algorithms that can be used to implement the proposed dynamic slicing approach in larger scenarios.

VI. Heuristic Algorithms for Dynamic Slicing

This section presents heuristic algorithms to be used for the VN mapping and for the VN reconfiguration phases of the dynamic slicing approach. These algorithms are based on the construction of a grooming graph, as presented in Ref. [9]. The grooming graph has an electrical and an optical layer, with three types of edges: *(i)* lightpath (LP) edge (i.e., if an existing lightpath has sufficient remaining bandwidth to support the capacity requirement of a virtual link); *(ii)* wavelength (WL) edge (i.e., if at least one free wavelength is available in the fiber link); and *(iii)* transponder (TP) edge (i.e., if a transponder is available for electrical-to-optical and optical-to-electrical conversion). The edge weights are selected so that the number of newly established lightpaths to support the VN capacity requirements is minimized. As mentioned earlier, it is assumed that each PN node is equipped with enough TPs to perform as many electrical-to-optical and optical-to-electrical conversions as needed. Hence, there exists a TP edge at each node in the grooming graph connecting the electrical and the optical layers.

A. *Heuristic$_{map}$*

This algorithm is used for mapping a VN request into the PN. Heuristic$_{\text{map}}$ is described in Algorithm 1. When a tenant requests a VN, it specifies the virtual node capacity requirements (i.e., compute/storage resources to be allocated in a DC, or number of BBU ports to be used in a BBU hotel), along with the set of candidate nodes in the PN where each virtual node can be mapped. Based on these specifications, the algorithm generates all possible node mappings (M) for the VN in the PN. For each of these node mapping options (i.e., m), Heuristic$_{\text{map}}$ computes the link mapping as follows. If the capacity requirement of a virtual link v (i.e., c_v) is less than the wavelength capacity (i.e., w), c_v is provisioned in one single try. Otherwise, c_v is split into several chunks, each one with a capacity requirement (i.e., c_p) less than or equal to w. A grooming graph is constructed to provision c_p. The graph is used to compute (in its electrical layer) the shortest path between the source and the destination node of v. By selecting a value for the weight of the TP edges that is much higher than the other edges of the grooming graph, the algorithm tends to find a path that does not use any TP edge. This results in having only LP edges in the path. In turn, this leads to provisioning c_p over already established lightpaths. However, if a TP edge is selected while computing the shortest path, a new lightpath has to be established to provision c_p. If all the virtual links of the VN are provisioned successfully, the algorithm computes the average number of wavelengths (i.e., u_m) in the PN corresponding to node mapping m. Finally, Heuristic$_{\text{map}}$ chooses the node mapping option m that results in the minimum value of u_m. Note that a less complex and faster heuristic algorithm may choose only one node mapping option based on some criterion, and then compute the link mapping only for that considered option. On the other hand, Heuristic$_{\text{map}}$ computes a link mapping

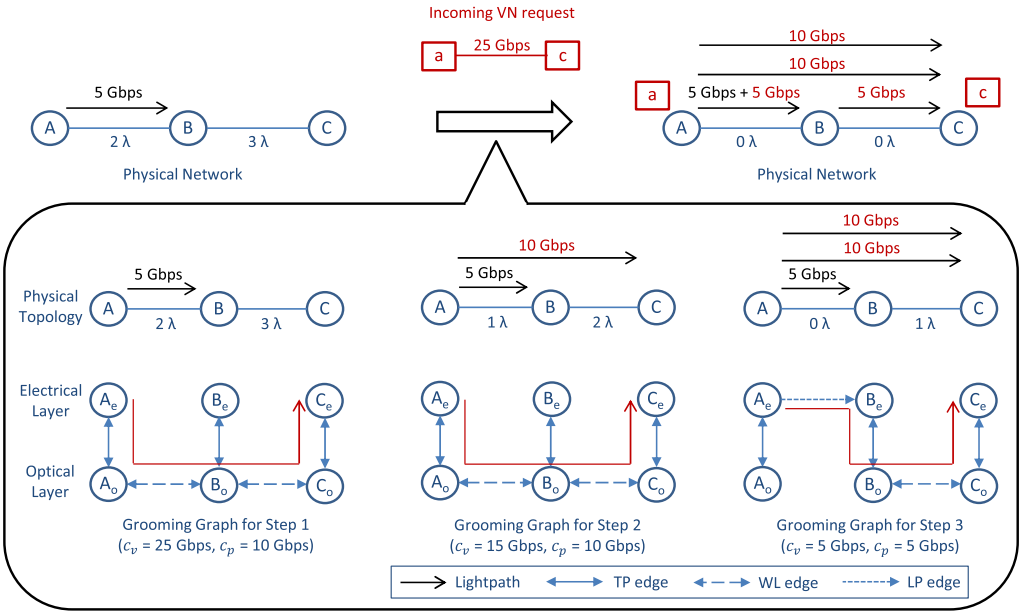

Fig. 4. Example of how to provision a virtual link using the grooming graph concept (assuming wavelength capacity of 10 Gbps).

for each possible node mapping in order to find the one that uses the least resources in the PN.

Figure 4 shows an example of how to provision a capacity $c_v = 25$ Gbps for a virtual link from node A to node C in the PN having three wavelengths in each link with $w = 10$ Gbps. It is assumed that there is already a lightpath established from node A to node B, carrying 5 Gbps of traffic for some other VN. Since c_v is greater than w, the value of c_p is initially set to w, i.e., 10 Gbps. Since there is no previously established lightpath that can accommodate c_p, there is no LP edge in the grooming graph at this step. After finding the shortest path from node A to node C in the electrical layer, i.e., from A_e to C_e (highlighted in red), a new lightpath is established for c_p. In a second step, $c_v = 15$ Gbps and $c_p = 10$ Gbps. This action results in adding another lightpath from node A to node C carrying 10 Gbps. Finally, in a third step, $c_v = 5$ Gbps. Since c_v is less than w, $c_p = 5$ Gbps. In this case, there exists a LP edge from A_e to B_e as the previously established lightpath can accommodate this value of c_p. However, there is no WL edge between corresponding nodes in the optical layer, as there is no free wavelength available in the link. Since the shortest path from A_e to C_e contains the LP edge, c_p is groomed with the existing lightpath from node A to node B. However, a new lightpath is established from node B to node C to accommodate c_p.

Algorithm 1 Heuristic$_{\text{map}}$

Input: current state of PN *init_state*; VN request specifications (with set of virtual nodes N and virtual links V)
Output: updated state of PN *final_state*; mapping status *mapping*

1. compute the set of all possible node mappings M considering resource requirements and candidate lists of each virtual node $n \in N$
2. **for** all node mappings m in M **do**
3. **for** all virtual links v in V **do**
4. **while** capacity requirement c_v of $v \neq 0$ **do**
5. **if** $c_v >$ wavelength capacity w **then**
6. capacity to be provisioned $c_p = w$
7. **else**
8. capacity to be provisioned $c_p = c_v$
9. **end if**
10. construct grooming graph for c_p and compute shortest path between source and destination of v in electrical layer of grooming graph
11. **if** path is not found **then**
12. remove m from set M
13. continue with next node mapping in step 2
14. **else**
15. provision c_p on computed path
16. $c_v = c_v - c_p$
17. **end if**
18. **end while**
19. **end for**
20. compute average wavelength usage u_m for m
21. store information about provisioned lightpaths for m
22. restore PN to *init_state*
23. **end for**
24. **if** M is empty **then**
25. *mapping* = FAILED
26. **else**
27. *mapping* = SUCCEEDED
28. choose the node mapping option m from M with minimum u_m
29. update PN *final_state* according to chosen node mapping and corresponding provisioned lightpaths
30. **end if**

B. Heuristic$_{reconf}$

This algorithm is used for adapting the mapping of an existing VN to the temporal changes of its virtual link capacity requirements. Heuristic$_{reconf}$ is described in Algorithm 2. When switching between daytime and nighttime, the algorithm sets the usage of each currently established lightpath to zero, and stores the information about their routes in the PN. Then, Heuristic$_{reconf}$ selects one VN and tries to provision first the virtual link with the largest capacity requirement. The capacity requirement is split into chunks (i.e., divided in c_p units, as was the case for Heuristic$_{map}$), and a grooming graph is constructed to provision each c_p, in the same way as explained in Fig. 4. While serving the first chunk, a LP edge is added in the grooming graph for each one of the lightpaths already existing in the PN and whose used capacity was set to zero. For the subsequent chunks, the usage of these lightpaths is updated accordingly. By setting the weight of the LP edges to a value that is much smaller than that of the TP edges, the algorithm tries to provision the new virtual link capacity requirements over existing lightpaths. Otherwise, if the path computed from the grooming graph contains a TP edge, a new lightpath needs to be established to serve c_p. If no path is found between the source and the destination, the corresponding virtual link is degraded by an amount equal to the capacity that could not be provided. The algorithm keeps on selecting VNs in a round robin fashion and repeats the same procedure until either all the required capacity is provided or the virtual links that could not be reconfigured are all degraded.

Algorithm 2 Heuristic$_{reconf}$

Input: current state of PN *init_state*; set of currently mapped VNs S; set of virtual links V_s for each VN $s \in S$ and updated capacity requirement $c_{v,s}$ of each virtual link $v \in V_s$; set of virtual nodes N_s for each VN $s \in S$ and mapping of each virtual node $n \in N_s$ in PN
Output: updated state of PN *final_state*; degradation $d_{v,s}$ of virtual link $v \in V_s$ for each VN $s \in S$

1. set usage of each currently established lightpath to zero
2. initialize all degradation variables $d_{v,s}$ to zero
3. pick a VN s from S in round robin fashion
4. take largest capacity requirement $c_{v,s}$ for virtual link v of s, for which the corresponding $d_{v,s}$ is zero
5. **if** $c_{v,s} >$ wavelength capacity w **then**
6. capacity to be provisioned $c_p = w$
7. **else**
8. capacity to be provisioned $c_p = c_{v,s}$
9. **end if**
10. construct grooming graph for c_p and compute shortest path between source and destination of v in electrical layer (using information about where the end points are mapped in PN)
11. **if** path is not found **then**
12. $d_{v,s} = c_{v,s}$
13. **else**
14. provision c_p on computed path
15. $c_{v,s} = c_{v,s} - c_p$
16. **end if**
17. update PN *final_state* according to provisioned lightpaths
18. repeat again from step 3 and pick the next VN until no more capacity can be provisioned

VII. Performance Evaluation

The performance of dynamic slicing is benchmarked against a static slicing approach. As already mentioned in the previous sections, with static slicing, each VN is mapped into the PN using either MILP$_{map}$ (for the comparison using a small-sized network) or Heuristic$_{map}$ (for the comparison using a large-sized network). This is done (for all the virtual links) considering the peak capacity requirements. This mapping then does not change during the whole service time of the VN (i.e., no reconfiguration). On the other hand, with dynamic slicing, a new VN request is mapped in the PN using either MILP$_{map}$ or Heuristic$_{map}$ considering the current capacity requirements of its virtual links (i.e., depending on whether the VN request arrives during the daytime or the nighttime). MILP$_{reconf}$ or Heuristic$_{reconf}$ is used during the reconfiguration process, i.e., when all the VNs mapped in the PN are reconfigured in order to adapt the mapping to the temporal variation of the capacity requirements of the virtual links.

The first part of the section considers a small 6-node PN where dynamic and static slicing are compared using both MILP and heuristic approaches. Then, the same comparison is performed with a large 38-node PN, this time using only the heuristic algorithms, since the MILP formulations do not scale for large network scenarios. The simulation setup used in both cases is described in the following.

A. Simulation Setup

Three different VN requests are considered for the radio, cloud, and IP tenants, as shown in Fig. 5. The requirements for the VNs requested by each tenant are presented in Table I. The table shows the virtual link capacity requirements for the daytime for the different tenants. The corresponding values for the nighttime are obtained by multiplying the daytime capacity requirements by the night traffic variation factor.

The cloud tenant allows the network provider to use any DC in the PN that has enough compute/storage resources. Similarly, the BBU hotel for the radio tenant can be chosen from any of the BBU hotels in the PN with enough BBU ports. However, the locations of RRUs and EPC are specified in the VN request. The same happens for the IP tenant, where ISP nodes that need to be connected are also specified in the VN request. For simplicity, in this work the specific locations for the RRUs, the EPC, and the ISP nodes are randomly chosen within their respective sets while generating a VN request. Moreover, it is assumed that DCs and BBU hotels in the PN have enough compute/storage resources and BBU ports available to guarantee that no VN is rejected during the node mapping process.

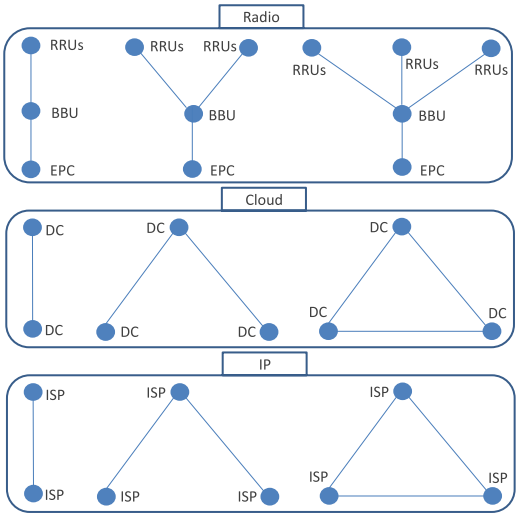

Fig. 5. VN requests considered for the radio, cloud, and IP tenants.

TABLE I
REQUIREMENTS OF RADIO, CLOUD, AND IP TENANTS

Radio
Number of RRUs per node ~ Uniform (5, 15)
Fronthaul (RRUs-BBU) link capacity (Day) = 10 Gbps
Backhaul (BBU-EPC) link capacity (Day) = 10% of fronthaul
Night traffic variation factor = 1/8 [26]
Cloud
DC-DC link capacity (Day) ~ Uniform (50, 100) Gbps
Night traffic variation factor = 25 [27]
IP
ISP-ISP link capacity (Day) ~ Uniform (1200, 1500) Gbps
Night traffic variation factor = 1/8 [27]

A total of 1500 VN requests are randomly generated in each experiment, and the results are averaged over 50 experiments. The experiments are run using an Intel Xeon CPU E5-1660 with 8 cores in a 3.00 GHz machine with 32 GB RAM. The day/night duration is fixed to 12 h, i.e., with dynamic slicing VNs need to be reconfigured every 12 h. The inter-arrival time and the service time of VN requests are exponentially distributed. The mean service time is set to 50 h, while the mean inter-arrival time is varied in order to generate different values of load. The MILP formulations are solved using IBM ILOG CPLEX. The parameters α, β, and γ for $\text{MILP}_{\text{reconf}}$ are set to 10,000, 1, and 0.0001, respectively. For the heuristic algorithms, the weights W_{TP}, W_{LP}, and W_{WL} for the TP, LP, and WL edges are selected as follows [13]:

$$W_{\text{TP}} = 100, \qquad W_{\text{LP}} = 0.01 \times H, \qquad W_{\text{WL}} = 0.01,$$

where H represents the number of hops of a lightpath.

B. Small 6-Node Physical Network

A small 6-node network [9] is used to compare the performance of dynamic versus static slicing and to validate the accuracy of the proposed $\text{Heuristic}_{\text{map}}$ and $\text{Heuristic}_{\text{reconf}}$ algorithms. Each fiber link in the PN has 80 wavelengths, each with 100 Gbps capacity. The 6-node network comprises two BBU hotels, one EPC, and four DCs connected to the PN nodes, as depicted in Fig. 6.

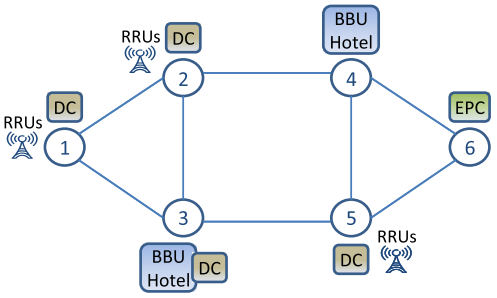

Fig. 6. Small 6-node physical network.

Figure 7(a) shows the value of the average VN rejection probability for different values of load using static and dynamic slicing. It can be seen that dynamic slicing reduces the VN rejection probability by approximately 5 times when the network is in medium to high load conditions (i.e., rejection probability <0.1). Moreover, the results of the MILP formulations and of the heuristic algorithms are very close to each other. This indicates that the proposed heuristic algorithms are well designed and give close to optimal results. It is interesting to note that, at very high load values, the heuristic algorithms for dynamic slicing seem to perform slightly better in terms of overall VN rejection probability than the MILP formulations. This is because the VN requests come with different capacity requirements (Table I). If (at high load conditions) MILP_{map} succeeds in mapping a very demanding VN request (i.e., in terms of capacity requirements), this might result in the inability to accommodate any other VN request for quite some time. On the other hand, $\text{Heuristic}_{\text{map}}$ might not be successful in mapping the same very demanding VN request. As a result, a few extra (and less capacity demanding) VN requests can be mapped in the PN.

The gains in terms of VN rejection probability with dynamic slicing come at the expense of VN degradation. Figure 7(b) presents the value of D [Eq. (1)] averaged over all the accepted VN requests. This value is very small for low loads and tends to increase at high loads. Nonetheless, it can be noticed that the biggest gains in terms of VN rejection come at relatively low degradation values, i.e., at most 0.4% at 12.5 Erlangs using MILP. However, VN degradation increases with the heuristic algorithms due to the sub-optimal performance of $\text{Heuristic}_{\text{reconf}}$.

Figure 7(c) shows how many wavelengths are used on average in each of the eight fiber links of the PN. As expected, by trying to match as closely as possible the VN mapping to the actual capacity requirements, the proposed dynamic slicing approach is able to reduce the number of congested fiber links.

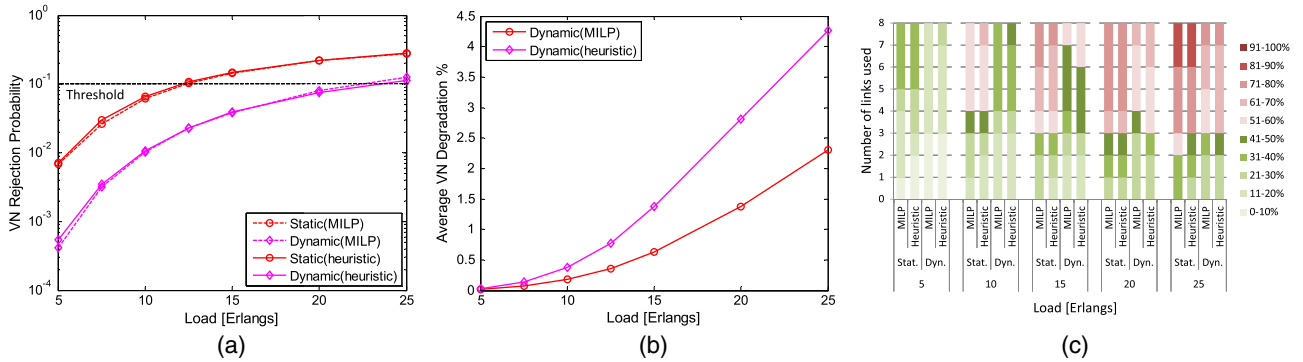

Fig. 7. Results using a 6-node physical network. (a) VN rejection probability for static and dynamic slicing, (b) average VN degradation for dynamic slicing, and (c) average link usage for different values of load.

Using the machine with the specifications described in the previous subsection, the average value of the execution time (i.e., over all the load values and considering 50 experiments for each load) when considering experiments with 1500 VN requests was found to be around 50 min with MILP formulations. However, the heuristic algorithms resulted in an average execution time of around 30 s. This clearly indicates how the heuristic algorithms are more likely to scale with a large-sized network. Note that the above-mentioned times include solving the mapping and reconfiguration problems multiple times, i.e., the MILP_{map}/$\text{Heuristic}_{\text{map}}$ is solved at each arrival out of 1500 VN requests, and the $\text{MILP}_{\text{reconf}}$/$\text{Heuristic}_{\text{reconf}}$ is invoked every 12 h in the experiment, i.e., when switching between day and night.

C. Large 38-Node Physical Network

The previous subsection showed that the heuristic algorithms for dynamic slicing perform very close to the optimum value. It is then safe to use them to assess the benefits of dynamic slicing in a large network scenario. The network under consideration is shown in Fig. 8. It consists of 38 nodes and 60 links [28]. Each fiber link has 80 wavelengths, each with 100 Gbps capacity. Two BBU hotels, one EPC, and five DCs are placed as shown in the figure. Each node not connected to a BBU hotel or an EPC is connected to a number of RRUs (not shown in Fig. 8 for the sake of clarity) following the information provided in Table I. The set of VN requests used in this new set of experiments is the one presented in Fig. 5.

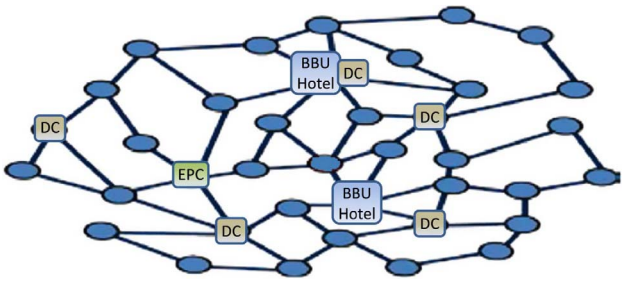

Fig. 8. Large 38-node physical network.

Figure 9(a) shows the average VN rejection probability for dynamic and static slicing. Dynamic slicing presents VN rejection probability values that are more than 1 order of magnitude better than static slicing when the PN is in medium to high load condition (i.e., rejection probability <0.1). Figures 9(b) and 9(c) show the per-VN-type breakdown of the VN rejection probability for static and dynamic slicing, respectively. The figures confirm that, regardless of the tenant type, dynamic slicing brings an improved VN rejection probability value. For the case of dynamic slicing, the cloud tenant has the highest rejection probability because of its high capacity requirements and the inherent inefficiency of $\text{Heuristic}_{\text{reconf}}$. However, for the case of static slicing, the cloud tenant has lower rejection probability than the IP tenant. This is because the location of ISP nodes is pre-defined in the VN request from the IP tenant. In contrast, for the cloud tenant, it is up to the network provider to choose the location of DC nodes, and the $\text{Heuristic}_{\text{map}}$ algorithm always chooses the best nodes, resulting in minimum PN resource usage.

Figure 9(d) shows the average VN degradation for different values of load. It can be observed that for load values corresponding to the highest benefits in terms of VN rejection probability, the degradation value is very small, i.e., it amounts to 0.7% at 35 Erlangs. Moreover, Fig. 9(e) shows the per-VN-type breakdown of the average VN degradation. From the figure, it can be seen that the cloud tenant experiences the highest degradation. This can be expected since the cloud tenant is the one with the highest capacity requirements.

Figure 9(f) presents the values of the average wavelength usage in the 60 fiber links of the PN. In this case, the wavelength usage is more disperse due to the large size of the network, i.e., some links at the edges of the network might not be used up to their maximum capacity. On the other hand, the links at the center of the network are the ones most utilized. However, on average, it can be seen that dynamic slicing can help to decrease the value of the fiber link congestion as compared to static slicing, which in turn helps to accept more VNs into the PN.

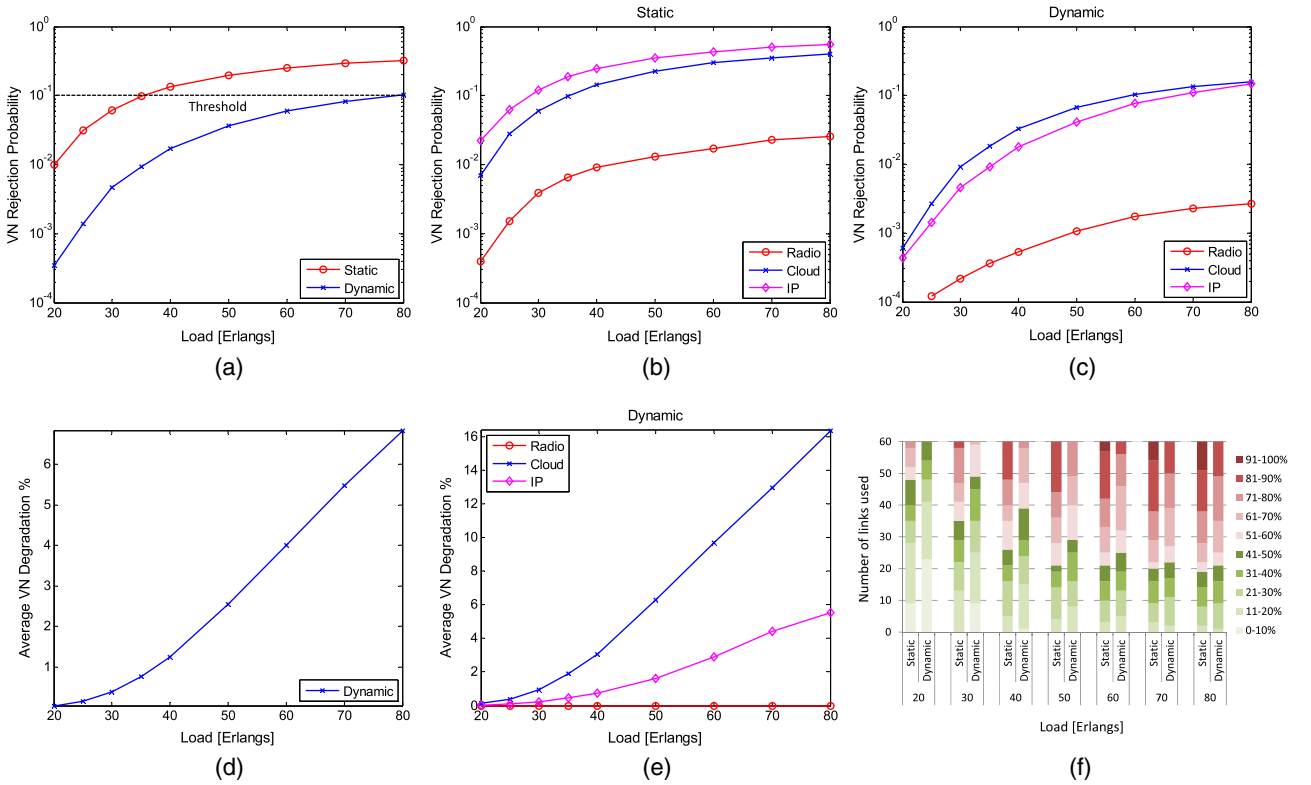

Fig. 9. Results using a 38-node physical network. (a) VN rejection probability and (b), (c) a per-VN-type breakdown of VN rejection probability for static and dynamic slicing; (d) average VN degradation and (e) a per-VN-type breakdown of average VN degradation for dynamic slicing; (f) average link usage for different values of load.

VIII. Conclusion

This paper evaluates the benefits of dynamic slicing, where VNs corresponding to different tenants are reconfigured according to the temporal variations of their capacity requirements. The performance of dynamic slicing is benchmarked against a static approach in which a fixed amount of resources (i.e., corresponding to the peak capacity requirements) is allocated to each slice during the entire VN service time.

Different from existing works, our approach leverages on advanced orchestration functionalities to intelligently assign and redistribute resources among the slices of different tenants. The need for dynamic slicing is motivated by the fact that, in 5G transport networks, with the need to jointly provision resources from different technology domains (e.g., radio, transport, and cloud), the orchestrator can achieve better resource efficiency by taking advantage of the presence of different types of tenants with (possibly) quite different service requirements.

Results based on MILP formulations and heuristic algorithms show that re-sizing and re-mapping resource slices (i.e., to match as closely as possible the requirements of each VN) result in a more efficient utilization of PN resources. In fact, in the considered scenarios, it has been observed that dynamic slicing can improve the VN rejection probability by more than 1 order of magnitude. This improvement can help network providers increase their revenues. On the other hand, using dynamic slicing may result in the degradation of some services. However, the paper shows that service degradation is very small, and within acceptable levels for both the network providers and the tenants.

In future work, we plan to study a more general dynamic slicing problem, i.e., considering variations in both virtual node and link resource requirements. In this case, the network provider might have to re-map a virtual node to some other location in the PN to match the updated requirements in terms of storage and processing. Moreover, it would be very interesting to also have some experimental results from a proof-of-concept demonstration of the dynamic slicing idea, which is left for a future work.

Appendix A

This section describes the procedure used to linearize some of the constraints in the $\text{MILP}_{\text{reconf}}$ formulation.

Constraint (21) is linearized by using the following constraints:

$$zq_{mn}^{ij} \le M \cdot p_{mn}^{ij}, \quad \forall\, i,j \in N_s\colon i \ne j, \quad \forall\, m \in N_s, n \in Nb_m, \tag{A1}$$

$$zq_{mn}^{ij} \geq -M \cdot (1 - p_{mn}^{ij}), \quad \forall i,j \in N_s: i \neq j, \quad \forall m \in N_s, n \in Nb_m, \tag{A2}$$

where M is a very large number.

On the other hand, Constraint (22) is linearized by using the following set of constraints:

$$zqp_{mn}^{ij} \leq M \cdot p_{mn}^{ij}, \quad \forall i,j \in N_s: i \neq j, \quad \forall m \in N_s, n \in Nb_m, \tag{A3}$$

$$zqp_{mn}^{ij} \geq -M \cdot p_{mn}^{ij}, \quad \forall i,j \in N_s: i \neq j, \quad \forall m \in N_s, n \in Nb_m, \tag{A4}$$

$$zqp_{mn}^{ij} \leq zq_{mn}^{ij} + M \cdot (1 - p_{mn}^{ij}), \quad \forall i,j \in N_s: i \neq j, \quad \forall m \in N_s, n \in Nb_m, \tag{A5}$$

$$zqp_{mn}^{ij} \geq zq_{mn}^{ij} - M \cdot (1 - p_{mn}^{ij}), \quad \forall i,j \in N_s: i \neq j, \quad \forall m \in N_s, n \in Nb_m, \tag{A6}$$

$$u_{ij} = \sum_{m \in N_s, n \in Nb_m} zqp_{mn}^{ij}, \quad \forall i,j \in N_s: i \neq j, \tag{A7}$$

where zqp_{mn}^{ij} denotes the product of zq_{mn}^{ij} and p_{mn}^{ij}, while M is a very large number.

Acknowledgment

The work described in this paper was carried out with the support of the Kista 5G Transport Lab (K5) project funded by VINNOVA and Ericsson, and of the H2020-ICT-2014 project 5GEx (Grant Agreement no. 671636).

References

[1] P. Ohlen, B. Skubic, A. Rostami, M. Fiorani, P. Monti, Z. Ghebretensae, J. Mårtensson, K. Wang, and L. Wosinska, "Data plane and control architectures for 5G transport networks," *J. Lightwave Technol.*, vol. 34, no. 6, pp. 1501–1508, Mar. 2016.

[2] A. Rostami, P. Öhlén, K. Wang, Z. Ghebretensaé, B. Skubic, M. Santos, and A. Vidal, "Orchestration of RAN and transport networks resources for 5G: An SDN approach," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 64–70, 2016.

[3] A. Mayoral, R. Munoz, R. Vilalta, R. Casellas, R. Martinez, and V. Lopez, "Need for a transport API in 5G for global orchestration of cloud and networks through a virtualized infrastructure manager and planner," *J. Opt. Commun. Netw.*, vol. 9, no. 1, pp. A55–A62, Jan. 2017.

[4] M. R. Raza, M. Fiorani, A. Rostami, P. Ohlen, L. Wosinska, and P. Monti, "Benefits of programmability in 5G transport networks," in *Optical Fiber Communication Conf. (OFC)*, 2017, paper M2G.3.

[5] A. Rostami, A. Vidal, M. A. Santos, M. R. Raza, F. Moradi, B. Pechenot, Z. Ghebretensae, P. Monti, and P. Ohlen, "An end-to-end programmable platform for dynamic service creation in 5G networks," in *Optical Fiber Communication Conf. (OFC)*, 2017, paper Tu3L.14.

[6] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: Substrate support for path splitting and migration," *Comput. Commun. Rev.*, vol. 38, no. 2, pp. 17–29, Mar. 2008.

[7] N. Chowdhury, M. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," in *IEEE INFOCOM*, Apr. 2009.

[8] A. Blenk and W. Kellerer, "Traffic pattern based virtual network embedding," in *CoNEXT Student Workshop*, ACM, 2013, pp. 23–26.

[9] S. Zhang, L. Shi, C. Vadrevu, and B. Mukherjee, "Network virtualization over WDM and flexible-grid optical networks," *Opt. Switching Netw.*, vol. 10, no. 4, pp. 291–300, Nov. 2013.

[10] A. Pages, J. Perello, S. Spadaro, and G. Junyent, "Strategies for virtual optical network allocation," *IEEE Commun. Lett.*, vol. 16, no. 2, pp. 268–271, Feb. 2012.

[11] Open Networking foundation (ONF), "Applying SDN architecture to 5G slicing," Technical Report TR-526, Issue 1, Apr. 2016 [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/Applying_SDN_Architecture_to_5G_Slicing_TR-526.pdf.

[12] L. Gong, W. Zhao, Y. Wen, and Z. Zhu, "Dynamic transparent virtual network embedding over elastic optical infrastructures," in *IEEE Int. Conf. on Communications (ICC)*, June 2013.

[13] J. Zhang, Y. Ji, M. Song, H. Li, R. Gu, Y. Zhao, and J. Zhang, "Dynamic virtual network embedding over multilayer optical networks," *J. Opt. Commun. Netw.*, vol. 7, no. 9, pp. 918–927, Sept. 2015.

[14] A. Mayoral, R. Vilalta, R. Casellas, R. Martinez, and R. Munoz, "Multi-tenant 5G network slicing architecture with dynamic deployment of virtualized tenant management and orchestration (MANO) instances," in *European Conf. on Optical Communication (ECOC)*, Sept. 2016.

[15] R. Vilalta, R. Muñoz, R. Casellas, R. Martinez, S. Peng, M. Channegowda, T. Vlachogiannis, R. Nejabati, D. Simeonidou, X. Cao, T. Tsuritani, and I. Morita, "Dynamic multi-domain virtual optical network deployment with heterogeneous control domains [Invited]," *J. Opt. Commun. Netw.*, vol. 7, no. 1, pp. A135–A141, 2015.

[16] S. Shakya, N. Pradhan, X. Cao, Z. Ye, and C. Qiao, "Virtual network embedding and reconfiguration in elastic optical networks," in *IEEE Global Communications Conf. (GLOBECOM)*, Dec. 2014.

[17] F. Gu, M. Peng, S. Khan, A. Rayes, and N. Ghani, "Virtual network reconfiguration in optical substrate networks," in *Optical Fiber Communication Conf. (OFC)*, 2013, paper NTh4J.6.

[18] M. A. Siqueira, F. N. Hooft, J. R. Oliveira, E. R. Madeira, and C. E. Rothenberg, "Providing optical network as a service with policy-based transport SDN," *J. Netw. Syst. Manage.*, vol. 23, no. 2, pp. 360–373, Apr. 2015.

[19] A. Asensio, M. Ruiz, L. M. Contreras, and L. Velasco, "Dynamic virtual network connectivity services to support C-RAN backhauling," *J. Opt. Commun. Netw.*, vol. 8, no. 12, pp. B93–B103, Dec. 2016.

[20] A. Aguado, M. Davis, S. Peng, M. V. Alvarez, V. Lopez, T. Szyrkowiec, A. Autenrieth, R. Vilalta, A. Mayoral, R. Munoz, R. Casellas, R. Martinez, N. Yoshikane, T. Tsuritani, R. Nejabati, and D. Simeonidou, "Dynamic virtual network reconfiguration over SDN orchestrated multitechnology optical transport domains," *J. Lightwave Technol.*, vol. 34, no. 8, pp. 1933–1938, Apr. 2016.

[21] "Dynamic end-to-end network slicing for 5G," Nokia White Paper, 2016 [Online]. Available: http://resources.alcatel-lucent.com/asset/200339.

[22] F. Musumeci, M. Tornatore, and A. Pattavina, "A power consumption analysis for IP-over-WDM core network architectures," *J. Opt. Commun. Netw.*, vol. 4, no. 2, pp. 108–117, 2012.

[23] J. Hu and B. Leida, "Traffic grooming, routing, and wavelength assignment in optical WDM mesh networks," in *IEEE INFOCOM*, Mar. 2004.

[24] "C-RAN—The road towards green RAN," China Mobile White Paper, Version 3.0, Dec. 2013 [Online]. Available: http://labs.chinamobile.com/cran/wp-content/uploads/2014/06/20140613-C-RAN-WP-3.0.pdf.

[25] L. Nonde, T. E. H. El-Gorashi, and J. M. H. Elmirghani, "Energy efficient virtual network embedding for cloud networks," *J. Lightwave Technol.*, vol. 33, no. 9, pp. 1828–1849, May 2015.

[26] G. Auer, O. Blume, V. Giannini, I. Godor, M. A. Imran, Y. Jading, E. Katranaras, M. Olsson, D. Sabella, P. Skillermark, and W. Wajda, "Energy efficiency analysis of the reference systems, areas of improvements and target breakdown," EARTH Project Report INFSO-ICT-247733, Deliverable D2.3, 2012.

[27] F. Morales, M. Ruiz, and L. Velasco, "Virtual network topology reconfiguration based on big data analytics for traffic prediction," in *Optical Fiber Communication Conf. (OFC)*, 2016, paper Th3I.5.

[28] S. Zhang, M. Xia, and S. Dahlfort, "Fiber routing, wavelength assignment and multiplexing for DWDM-centric converged metro/aggregation networks," in *European Conf. on Optical Communication (ECOC)*, Sept. 2013.