Optimal Virtual Slice Composition Toward Multi-tenancy over Hybrid OCS/OPS Data Center Networks

Albert Pagès, Miguel Pérez Sanchís, Shuping Peng, Jordi Perelló, Dimitra Simeonidou and Salvatore Spadaro

***Abstract*—Multi-tenancy is a key feature of modern data centers. It allows for the existence of multiple independent virtual infrastructures, called virtual slices, on top of the same physical infrastructure, each one of them specially tailored to the tenants' needs. In such a scenario, an optimal mapping of the virtual slices plays a capital role towards an efficient utilization of the data center network resources, potentially saving costs for the data center owner. However, due to the increasing trend of bringing optics to data center networks, specific virtual slice mapping mechanisms accounting for the particularities of the optical medium (e.g. wavelength continuity constraint) have to be investigated. For this, we present an Integer Linear Programming (ILP) model for optimally mapping a set of virtual slices from different tenants in a hybrid Optical Circuit Switching (OCS)/Optical packet Switching (OPS) data center network with the aim to minimize the necessary optical transponders to be equipped in the network. Additionally, we also present a lightweight heuristic for the cases where the ILP model scalability is compromised. The benefits of the proposals are highlighted by benchmarking them against a pure OCS solution through extensive simulations.**

***Index Terms*—Data centers, virtualization, optimization, multi-tenancy, OCS, OPS.**

I. Introduction

NOWADAYS data centers (DCs) are one of the largest IT systems in the world, consisting of thousands of servers and handling large amounts of traffic in their infrastructures. It is forecast that the traffic handled by DCs will double by 2018, reaching an overall traffic of 6.5 Zettabytes per year [1]. Moreover, it is predicted that the vast majority of such a traffic (around 75%) will remain inside the DCs. This puts a great pressure to existing electronic-based DC networks (DCNs), since they do not scale well in terms of latency, bandwidth and power consumption. Moreover, traditional DCNs have important limitations on the maximum bisection bandwidth that they can provide [2]. For this reason, in order to cope with such an increase on the intra-DC traffic new DCN architectures need to be properly investigated.

A very hot research trend is to bring optical technologies inside the DC so as to replace current electronic-based network fabrics [3]–[7]. In this regard, there are essentially two major trends in research initiatives and projects: the ones that propose hybrid electronic/optical solutions for DCNs (e.g. [4], [5]) as an evolutionary step towards high performance DC infrastructures; and others that plead for a more revolutionary approach, proposing all-optical network fabrics, either based on circuit switching (e.g. [6]) or packet switching (e.g. [7]). All-optical DCNs are promising solutions offering high throughput, low latency and reduced power consumption when compared to electronic-based (e.g. Ethernet, Infiniband) DCNs. [8].

In this context, the FP7 European project LIGHTNESS [9] presents a revolutionary architecture solution for the DCN. It is based on a hybrid OCS/OPS DCN, harnessing the superior flexibility, scalability and bandwidth of the optical transport medium, as well as a unified Software Defined Network (SDN)-based control plane for a fast control and configuration of the DCN infrastructure. The characteristics of intra-DC traffic are very heterogeneous, with connections transmitting large amounts of data (elephant flows) and others only requiring sporadic transmissions of low amounts of data (mice flows) [10]. Moreover, there are also high disparities among the duration of the flows (long-lived and short-lived). Hence, it becomes difficult to efficiently accommodate all the requirements of the connections with a single technology for the DCN. For this reason, LIGHTNESS proposes a novel hybrid OCS/OPS DCN: on one hand, OCS behaves very efficiently when supporting long-lived smooth data flows, for which Quality of Service (QoS) guarantees are ensured; on the other hand, OPS leverages on the statistical multiplexing of optical resources to achieve highly flexible transport services with very low end-to-end latencies for short-lived sporadic data flows. With such an approach, LIGHTNESS seeks to overcome current DCN architectures in order to scale beyond their limitations in terms of flexible traffic handling and allocation as well as limited throughput, latency and energy efficiency.

An additional key feature that modern DCs have to address is the possibility of leasing part of their infrastructures to external entities in order to exploit innovative Infrastructure as a Service (IaaS) solutions and develop their own business models. These entities, hereafter referred as tenants, may request to the DC owner specific virtual infrastructures, called virtual slices, composed of virtual nodes with computational capabilities (e.g. Virtual machines (VMs)) and virtual links, stating the bandwidth requirements for the communication between virtual nodes. Under such circumstances, an optimal mapping of the virtual slices becomes crucial for the overall performance of the DC as well as to fully satisfy the needs of the several tenants while guaranteeing the isolation between them. Thus, it is the responsibility of the DC owner to provide such mapping. In aims to increase the physical resource utilization, several virtual slices of the same tenant may be composed and mapped over the same physical resources, resulting in aggregated synthetic infrastructures, one per tenant. In this regard, a synthetic infrastructure represents the particular slice of the

A. Pagès, M. Pérez Sanchís, J. Perelló and S. Spadaro are with the Advanced Broadband Communications Center (CCABA), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, e-mail: albert-pages@tsc.upc.edu, miguelpsanchis@gmail.com, perello@ac.upc.edu, spadaro@tsc.upc.edu.

S. Peng and D. Simeonidou are with the High-Performance Networks Group, University of Bristol, UK, e-mail: shuping.peng@bristol.ac.uk, dimitra.simeonidou@bristol.ac.uk

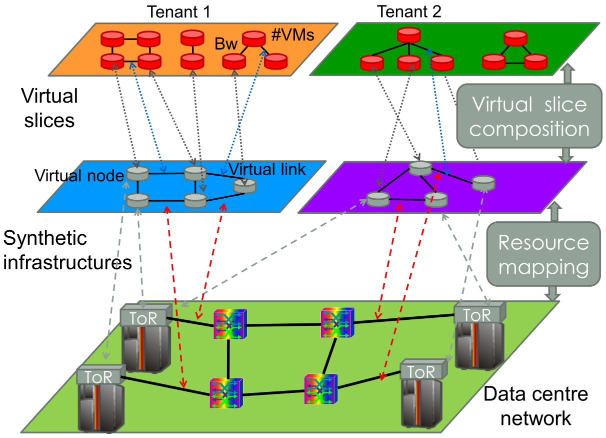

Fig. 1. Multi-tenant scenario.

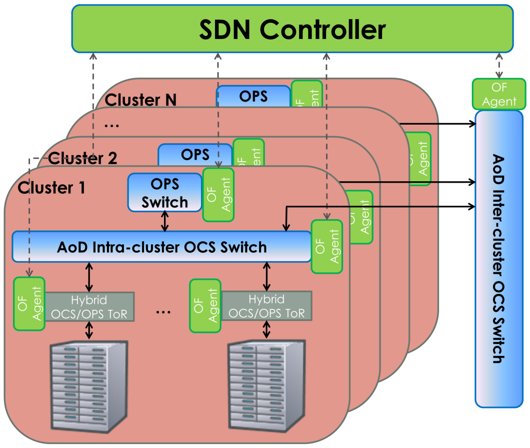

Fig. 2. LIGHTNESS hybrid OCS/OPS optical DCN scenario.

DC infrastructure where all the virtual slices of a tenant have been mapped. Nevertheless, logical independence is still guaranteed and the individual virtual slices are exposed towards the tenant. Figure 1 exemplifies this scenario.

Virtual slice allocation in DC environments has been widely studied and several mapping strategies can be found in the literature (e.g. [11], [12]). Common practices in this regard are to perform the mapping with aims to maximize the energy efficiency of the DC or to achieve high-availability of the virtual slices, e.g. by encouraging rack diversity in the virtual node mapping. Nevertheless, the architecture proposed by LIGHTNESS opens up new challenges on the mapping process. Indeed, each virtual link should be mapped to the best technology according to the link characteristics and the intended goal, while accounting for the particularities of the individual technologies. Authors in [13] showed a virtual slice mapping mechanism for dynamic scenarios in aims to allow multi-tenancy in a hybrid OCS/OPS DCN. They showed that a hybrid OCS/OPS DCN can yield significant benefits on the acceptance rate of the virtual slices when compared to pure OCS DCN solutions.

Following this work, in the current paper we focus on the off-line resource planning case and present novel mechanisms to address the problem of optimally mapping several virtual slice requests in a hybrid OCS/OPS DCN in the presence of several tenants with the aim of minimizing the necessary optical transponders to be equipped at the DCN to allocate them. The next sections are structured as follows: section II details the scenario that we are considering and elaborates on the optimization problem that we are targeting. Next, section III presents the proposed mechanisms to tackle the optimization problem under study. Section IV evaluates the performance of the proposed solutions. Finally, sections V draws up the main conclusions of the present work.

II. Scenario Description

Typical DCs consist on sets of servers organized in racks, which then are grouped in clusters to allow better scalability/manageability of the infrastructure. Communication between servers is achieved thanks to an intra-DCN, with servers accessing to the DCN thanks to a Top of the Rack (ToR) switch, one per rack. The specific solution adopted by the LIGHTNESS project is depicted in Figure 2. There, the ToRs are equipped with a set of optical transponders that allow for establishing both OCS and OPS optical channels whenever needed. All the ToRs of a cluster are plugged thanks to fiber links to an intra-cluster Architecture on Demand (AoD) OCS switch [14]. Moreover, each cluster is provided with an OPS switch node, which is also connected thanks to a fiber link to the AoD OCS switch. The presence of the AoD OCS switch allows for the dynamic reconfiguration of the interconnections between ToRs, allowing for the establishment of OPS channels by transparently connecting the ToRs to the OPS switch or interconnect them through OCS channels. Moreover, it allows for a flexible allocation of the number of OCS or OPS channels between arbitrary ToRs, tunning the capacity according to dynamic traffic needs. Finally, the communication between clusters is enabled thanks to an inter-cluster AoD OCS switch, which can also be employed to establish either OCS or OPS optical channels whenever is needed. Such an architecture allows for a more flat and flexible network fabric, overcoming the limitations of tree-based network topologies utilized on traditional DCs [15].

All the intra-DCN infrastructure is controlled and configured by a centralized SDN controller deployed on top of the DCN. The SDN controller communicates with each switching element through the southbound interface, which implements the Open Flow (OF) protocol. A dedicated OF agent is deployed for each switching device as to offer a standard communication between the controller and the hardware elements. Thanks to this, the SDN controller translates the requirements coming from the application plane to specific configurations of the data plane devices, namely, the activation of optical transponders at the ToRs, the configuration of the switching elements at the AoD and the population of the Look Up Tables (LUT) at the OPS switch. However, some challenges arise in the control aspect. For instance, custom extensions to the OF protocol must be implemented in order to support each one of the optical elements present in the intra-DCN, as current OF version does not support optical devices. Moreover, fast configuration of the data plane must be achieved in order to support the high traffic dynamicities in the DCN. The LIGHTNESS SDN controller implements the corresponding OF extensions, allowing for a millisecond scale configuration of the optical elements.

Hence, we consider a transparent hybrid OCS/OPS optical

DCN to enable the inter-rack communications in the DC, where each ToR is connected thanks to fiber links to a hybrid OCS/OPS-enabled optical DCN. Moreover, we assume that opto-electrical (EO) ToRs are equipped in the racks, with intra-rack communications taking place in the electrical domain while inter-rack communications are established through optical channels. Given this scenario, several tenants request for a set of virtual slices to be allocated in the DC infrastructure. A virtual slice is a logical infrastructure composed of both virtual nodes with computational capabilities (VMs) and virtual links connecting them with a required bandwidth. A particular tenant may ask for several virtual slices, each one of them specifically tailored to cover the necessities of different applications. For instance, a tenant may ask for a virtual slice with very low latencies in their virtual links for the transfer of real-time video. At the same time, the tenant may also ask for a virtual slice whose main purpose is the transfer of very bulky amounts of data, hence, requiring high bandwidths per virtual link, but communication latencies are less critical. In such a case, a single generic virtual slice cannot cover all the tenants necessities efficiently, so multiple specific-purpose virtual slices are requested. In such circumstances, the mission of the DC infrastructure owner is to provide the mapping of these virtual slices onto actual physical resources: virtual nodes onto physical servers and virtual links onto optical connections. Moreover, in the current scenario, a proper switching technology (OCS or OPS) has to be chosen for mapping the virtual links depending on their characteristics (i.e., bandwidth and QoS). In this regard, since optical resources are expensive, particularly optical transponders, we target a planning method where the objective is to allocate an already known set of virtual slices while minimizing the necessary number of both optical transmitters (Tx) and receivers (Rx) to be equipped at the ToRs.

To this end, the virtual slices of a tenant can be composed onto a single synthetic infrastructure (i.e. a single physical slice) potentially saving optical Tx/Rx. For this, virtual links can exploit the grooming capabilities of OCS, which would allow to map several virtual links onto the same lightpath as long as the whole end-to-end physical path is shared. [16] In the case of OPS, virtual links can exploit the statistical multiplexing property of packet switching networks [17], allowing for virtual links with different end-points to share the same wavelength, thus saving some Tx/Rx at the ToRs since a single transponder could be used to transmit different packet flows from the same source to different destinations or from several sources to the same destination. However, in OPS, due to the lack of optical buffers, packet contention may happen for optical packet coming out at the same time from the same port of the OPS switch [18]. In fact, such phenomena increases with the the offered load per port, as the chances of packets coming out at the same time through the same output port are higher. To ensure a proper QoS, the load per port and wavelength in OPS must be kept below certain limits. For this, in general, virtual links also ask for a QoS in terms of a bandwidth limit restriction. If OPS is employed to serve them, the load per wavelength is limited to the most restrictive bandwidth limit of all virtual links mapped over the wavelength. Nevertheless, thanks to the electronic capabilities at the source ToR, it is possible to properly order electronic packets belonging to different virtual links coming from the same destination and going to the same source, before being sent optically, as long as they are mapped over the same end-to-end path. In such situation, no contention is experienced at the OPS switch. Thus, the resulting aggregated OPS flow can be mapped over the same wavelength in the same manner that it would be on the case of grooming in OCS, saving optical Tx/Rx.

Note that, although virtual links of the same tenant may share optical resources, it is important to enforce physical isolation between virtual links of different tenants to avoid any kind of interference. Besides this constraint, it is also important to ensure high availability for the virtual nodes, since it is a desirable feature for virtual slices as commented during the introduction. To this end, we add the restriction that virtual nodes of the same virtual slice must be mapped into different racks to provide resilience against server or rack failures. Nevertheless, different nodes of different virtual slices of the same tenant can be mapped onto the same rack.

With such conditions and scenario, the following section states the optimization problem that we are targeting.

A. Problem Statement

The objective of the optimization problem under study is to find the most suitable technology (OPS or OCS) and physical resources (nodes, paths and wavelengths) to allocate an already known set of virtual slice requests of different tenants with the objective of minimizing the necessary optical Tx/Rx to be equipped at the ToRs of the DCN.

Objective:

- Minimize the necessary number of optical Tx/Rx to be equipped at the ToRs of the DCN.

Given:

1) a transparent hybrid OCS/OPS DCN represented by the graph $G_n = (N_f, E_f)$, being N_f the set of optical nodes either ToR, OCS or OPS switches and E_f the set of physical links.
2) an ordered set of wavelengths per physical link W of enough size to support all virtual slice request. Thus, uncapacitated physical links are assumed. The final capacity of the physical links,which may be different for every physical link, will be determined by the optimization procedure.
3) a set of servers arranged in racks, with the servers in each rack connected to their corresponding ToR switch. We represent with VM_{n_f} the aggregated capacity in terms of VMs of all servers of the rack connected to the ToR $n_f \in N_f$. Hence, we can simplify the virtual node mapping phase, associating the capacity in VMs of a rack to their corresponding ToR, since we do not tackle the specific mapping of a VM inside a particular server of a rack. Thus, the node mapping consist on finding the rack with enough IT resources (i.e., VMs) that allows for the successful allocation of the virtual nodes.
4) a set of virtual slice requests D. Particularly, D is the whole set of virtual slices requested by all tenants, with d_i the subset of D containing all the virtual slice request from tenant i and element $d_{i,j}$ the j^{th} request from tenant i. Each virtual slice is represented by the undirected graph $G_d = (N_v, E_v)$, being N_v the set of virtual nodes and E_v the set of undirected virtual links. Each virtual node requests a capacity in terms of VMs

represented by VM_{n_v}. Additionally, each virtual link requests (in both directions) a bandwidth capacity as a fraction of the total wavelength capacity represented by B_{e_v} and imposes a bandwidth limit in all physical wavelengths that support it (due to QoS restrictions) represented by $B_{e_v}^{max}$.

Find:

- The node and link mapping of virtual nodes and virtual links, respectively, of all virtual slices in D.

Subject to:

1) all virtual slices have to be mapped (no virtual slice blocking is permitted).
2) optical resources assigned to a tenant cannot be shared with other tenants. Nevertheless, optical resources assigned to a virtual slice can be shared with other virtual slices of the same tenant.
3) the wavelength continuity constraint must be ensured along the path onto which a virtual link is mapped (a transparent DCN is considered).
4) a virtual link has to be mapped onto a single technology, either OCS or OPS, but not both at the same time. Nevertheless, different virtual links of the same virtual slice may be mapped over different technologies in aims of saving Tx/Rx at the ToRs.
5) a virtual node can only be mapped to a single physical node.
6) a physical node can only host one virtual node of a certain virtual slice. Nevertheless, physical nodes can host virtual nodes of multiple virtual slices of the same tenant.
7) the aggregated capacity in VMs of virtual nodes mapped in a rack cannot surpass the total capacity of the physical node.
8) the total capacity of a wavelength must not be exceeded.
9) in OPS, the total flow circulating through a wavelength and output port of an OPS switch must not surpass the most restrictive QoS limit imposed by any of the supported flows.
10) the total number of active incoming/outgoing wavelengths from/to an OCS or OPS switch must not surpass its port count.
11) a transponder (hence, a physical wavelength) can only support OCS or OPS flows, not both simultaneously.
12) in OCS, multiple virtual links can share the same circuit (wavelength) in a physical link as long as they share the whole end-to-end path thanks to the grooming capabilities.

In the following, we provide a Mixed Integer Linear Programming (MILP)-based mechanism to attack the stated optimization problem. Additionally, we also provide a purely heuristic mechanism for the scenarios where the scalability of the MILP mechanism can be compromised.

III. Proposed Mechanisms

A. Notation Definition

Before going into the details of the proposed mechanisms, let us define some extra notation:

- P: set of end-to-end paths between ToRs in G_n.
- $\bar{p}$: symmetrical path to $p \in P$.
- P_{e_f}: set of paths that traverse physical link $e_f \in E_f$, with $P_{e_f} \subseteq P$.
- N_c^{OCS}: set of OCS switches in G_n, with $N_c^{OCS} \subseteq N_f$.
- N_c^{OPS}: set of OPS switches in G_n, with $N_c^{OPS} \subseteq N_f$.
- L_{OCS}: port count limit of an OCS switch.
- L_{OPS}: port count limit of an OPS switch.
- $\delta^+(n_f)$: set of outgoing links from node $n_f \in N_f$.
- $\delta^-(n_f)$: set of incoming links to node $n_f \in N_f$.
- $a(\cdot)$: source of a virtual link e_v or physical path p.
- $b(\cdot)$: destination of a virtual link e_v or physical path p.

The definition of P allows us to easily tackle the wavelength continuity constraint of the virtual links as wavelength resources are reserved explicitly along end-to-end paths, hence, they remain the same on all physical links forming the selected path. As for $\bar{p}$, it represents the path composed exactly with the opposite sequence of physical links respect to p. This will allow us to model the bidirectionality of the virtual links. Finally, L_{OCS} and L_{OPS} are used to model the switching capacity limits of an OCS or OPS switch, respectively, that is, a switch can commute simultaneously a number of active wavelength equal to its port count.

Once all these definitions have been introduced, we will proceed with the description of the proposed mechanisms.

B. MILP-based Algorithm

In this section, we propose a novel MILP-based mechanism to optimally address the problem presented in the previous section. Algorithm 1 depicts the pseudo-code of the presented mechanism. Basically, after some pre-processing, the mechanism executes iteratively a MILP formulation for every tenant in the demand set with the aim to allocate all their requested virtual slices in order to obtain the minimum necessary optical Tx/Rx to be equipped at the ToRs. Since we are targeting a dimensioning problem, and because one of the main requirements is to guarantee the physical isolation between tenants, the presented iterative approach is completely valid since optical resources employed for a particular tenant are made unavailable to the rest. Additionally, this iterative approach allows for a better scalability of the mechanism since targeting a joint optimization of all tenants at once would make the optimization problem intractable. For these reasons, we propose the aforementioned iterative approach, where the MILP formulation is applied for one tenant at each step. Nevertheless, since the mechanism still relies on a MILP formulation, its scalability may be compromised when the size of the problem instance grows (e.g., a tenant requests a large number of virtual slices composed of many virtual nodes and links). We will discuss such a limitation later on.

As for the pre-processing phase, its purpose is to manipulate the several virtual slice requests of a tenant in order to compose a single request, which considerably reduces the complexity of the optimization problem. Basically, the process involves composing the graph representations of the several virtual slices into a single graph representation with several components (sub-graphs), one for each particular virtual slice request. Since these components are the representation of the original virtual slice requests of a tenant, they are subject to the restrictions stated during the previous section: virtual nodes of a particular component cannot be mapped onto the same physical resource (node) for reliability reasons; nevertheless, different components may share resources between them, either nodes or lightpaths. In this regard, we define $N_v^t \subseteq N_v$ as the set of virtual nodes belonging to the component t inside the composed graph. This

```
Inputs: $D$, $G_n$, $W$, $L_{OCS}$, $L_{OPS}$; Outputs: $Sol$
Phase 1: Pre-processing
$D \leftarrow$ aggregate all virtual slice requests of a tenant into a single graph for each subset $d_i \in D$
$P \leftarrow$ set of path between all ($s$, $t$) pairs in $G_n$
$Sol \leftarrow \emptyset$
Phase 2: MILP solving
for $d = 1$ to $|D|$ do
    $Sol \leftarrow Sol \cup$output from MILP($d$,$P$,$G_n$,$W$,$L_{OCS}$,$L_{OPS}$)
    Update physical resources availability
Return $Sol$
Demands served
```

Algorithm 1: MILP mechanism pseudo-code.

definition will allow us to account for the potential sharing of physical resources among the different components.

After this discussion, we proceed now on detailing the proposed MILP formulation for obtaining the optimal mapping of a single tenant. We remind the reader that each tenant is mapped independently from the others, thus, forcing physical isolation for the optical resources. All virtual nodes and links considered in the formulation come from the tenant composed graph as explained above. The decision variables of the MILP formulation are:

$t_{e_f,w}$: binary; 1 if any virtual link is mapped through physical link e_f and wavelength w, 0 otherwise.

X_{e_v}: binary; 1 if virtual link e_v is served employing OPS, 0 otherwise.

Z_{e_v}: binary; 1 if virtual link e_v is served employing OCS, 0 otherwise.

$x_{e_v,p,w}$: real; amount of bandwidth from virtual link e_v that circulates through path p and wavelength w if OPS is chosen.

$z_{e_v,p,w}$: real; amount of bandwidth from virtual link e_v that circulates through path p and wavelength w if OCS is chosen.

y_{n_v,n_f}: binary; 1 if virtual node n_v is mapped onto the rack connected to ToR n_f, 0 otherwise.

$A_{p,w}$: real, indicates the aggregated OPS flow circulating through path p and wavelength w.

$C_{e_v,p,w}$: binary; 1 if virtual link e_v is served utilizing OPS through path p and wavelength w, 0 otherwise.

$F_{e_f,p,w}$: binary; 1 if the aggregated OPS traffic circulating through path p and wavelength w is utilizing alone physical link e_f, 0 otherwise.

$S_{e_v,p,w}$: binary; 1 if virtual link e_v is served utilizing OCS through path p and wavelength w, 0 otherwise.

The exact details of the MILP formulation are as follows:

$$\min \sum_{n_f \in N_f \backslash N_c^{OPS}, N_c^{OCS}} \sum_{e_f \in \delta^+(n_f), \delta^-(n_f)} \sum_{w \in W} t_{e_f,w} \quad (1)$$

Objective function (1) has the goal of minimizing the number of wavelengths that are active at the outgoing/incoming links from/to the ToRs, thus effectively minimizing the number of necessary optical Tx/Rx at the DCN, since each active wavelength accounts for a Tx at the source ToR and a Rx at the destination ToR. Next, we will detail the constraints.

$$X_{e_v} + Z_{e_v} = 1, \forall e_v \in E_v \quad (2)$$

Constraint (2) forces that all the virtual links of the request are mapped to either OCS or OPS, but not both at the same time since, although we consider hybrid virtual slices, we do not consider the possibility of splitting traffic of a virtual link across different DCN transport technologies.

$$\sum_{p \in P} \sum_{w \in W} x_{e_v,p,w} = 2 \cdot B_{e_v} \cdot X_{e_v}, \forall e_v \in E_v \quad (3)$$

$$\sum_{p \in P} \sum_{w \in W} z_{e_v,p,w} = 2 \cdot B_{e_v} \cdot Z_{e_v}, \forall e_v \in E_v \quad (4)$$

Constraints (3) and (4) ensure that all the requested bandwidth of every virtual link is served with the chosen technology. Note that these constraints account for twice the requested bandwidth per virtual link. This is due to the bidirectional nature of the virtual links. For simplicity we are considering that the graph representation of the virtual slices is an undirected graph. Therefore, each virtual link in the graph should be provided with twice the requested bandwidth to account for the two directions of the communication. The correct handling of the two directions is done through constraints (8) and (9), which will be explained later. Such approach allows us to reduce the number of binary variables associated to virtual link by a factor of 2, potentially reducing the size of the branch and bound tree and the execution time of the model.

$$\sum_{n_f \in N_f \backslash N_c^{OPS}, N_c^{OCS}} y_{n_v,n_f} = 1, \forall n_v \in N_v \quad (5)$$

$$\sum_{n_v \in N_v^t} y_{n_v,n_f} \leq 1, \forall N_v^t, n_f \in N_f \backslash N_c^{OPS}, N_c^{OCS} \quad (6)$$

$$\sum_{n_v \in N_v} VM_{n_v} \cdot y_{n_v,n_f} \leq VM_{n_f}, \forall n_f \in N_f \backslash N_c^{OPS}, N_c^{OCS} \quad (7)$$

As for the virtual node mapping, constraint (5) ensure that a virtual node is mapped to only one physical node, that is, a single virtual node cannot be mapped to multiple physical nodes. Constraint (6) guarantees that a particular physical node does not host more than one virtual node per component inside the tenant aggregated virtual slice request. Note that such restriction is applied per component, effectively allowing the mapping of virtual nodes belonging to different virtual slices (components) onto the same physical node. Finally, constraint (7) ensures that the aggregated capacity of VMs of all the virtual nodes mapped onto a physical rack does not exceed its capacity.

$$\sum_{w \in W} x_{e_v,p,w} = \sum_{w \in W} x_{e_v,\bar{p},w}, \forall e_v \in E_v, p \in P \quad (8)$$

$$\sum_{w \in W} z_{e_v,p,w} = \sum_{w \in W} z_{e_v,\bar{p},w}, \forall e_v \in E_v, p \in P \quad (9)$$

$$\sum_{w \in W} (x_{e_v,p,w} + x_{e_v,\bar{p},w} + z_{e_v,p,w} + z_{e_v,\bar{p},w}) \leq 2 \cdot (y_{a(e_v),a(p)} + y_{a(e_v),b(p)}), \forall e_v \in E_v, p \in P \quad (10)$$

$$\sum_{w \in W} (x_{e_v,p,w} + x_{e_v,\bar{p},w} + z_{e_v,p,w} + z_{e_v,\bar{p},w}) \leq 2 \cdot (y_{b(e_v),a(p)} + y_{b(e_v),b(p)}), \forall e_v \in E_v, p \in P \quad (11)$$

As said before, due to the bidirectional nature of the virtual links and the undirected graph representation of them, each virtual link is actually provided with twice the requested bandwidth. In order to properly map the virtual link, half of the total bandwidth should be mapped in one direction and the remaining half in the other. Constraints (8) and (9) account for this, forcing that the bandwidth assigned to a virtual link in a particular path p should be equal to the bandwidth assigned to the symmetrical path $\bar{p}$. In this way, the total assigned bandwidth is halved among the two directions of the communication. Constraints (10) and (11) restrict virtual link mappings to physical paths connecting the physical nodes over which the remote endpoints of the virtual links are mapped, accounting for the undirected nature of the virtual links representation.

$$\sum_{e_v \in E_v} \sum_{p \in P_{e_f}} (x_{e_v,p,w} + z_{e_v,p,w}) \leq t_{e_f,w}, \forall e_f \in E_f, w \in W \quad (12)$$

$$\sum_{e_v \in E_v} \sum_{p \in P_{e_f}} (x_{e_v,p,w} + z_{e_v,p,w}) \leq 1, \forall e_f \in E_f, w \in W \quad (13)$$

Constraint (12) is the definition of variables $t_{e_f,w}$, that is, it determines which are the active wavelengths in the physical links. Constraint (13) is the wavelength capacity constraints, limiting the total traffic flow circulating through a wavelength in a physical link to the capacity of the wavelength.

The following collection of constraints (14)–(20) will help us on modeling the QoS restrictions in OPS as explained in section II.

$$A_{p,w} = \sum_{e_v \in E_v} x_{e_v,p,w}, \forall p \in P, w \in W \quad (14)$$

$$\begin{aligned} &\sum_{p \in P_{e_f}} A_{p,w} + (1 - B^{max}_{e_{v_0}}) \cdot C_{e_{v_0},p_0,w} \leq 1 + F_{e_f,p_0,w}, \\ &\forall n_f \in N_c^{OPS}, e_f \in \delta^+(n_f), e_{v_0} \in E_v, p_0 \in P_{e_f}, w \in W \end{aligned} \quad (15)$$

$$x_{e_v,p,w} \leq C_{e_v,p,w}, \forall e_v \in E_v, p \in P, w \in W \quad (16)$$

$$C_{e_v,p,w} \leq M \cdot x_{e_v,p,w}, \forall e_v \in E_v, p \in P, w \in W \quad (17)$$

$$F_{e_f,p,w} \leq M \cdot A_{p,w}, \forall e_f \in E_f, p \in P, w \in W \quad (18)$$

$$\begin{aligned} &A_{p_0,w} - M \cdot \sum_{p \in P_{e_f}, p \neq p_0} A_{p,w} \leq F_{e_f,p_0,w}, \\ &\forall e_f \in E_f, p_0 \in P, w \in W \end{aligned} \quad (19)$$

$$\begin{aligned} &F_{e_f,p_0,w} \leq 1 - m \cdot \sum_{p \in P_{e_f}, p \neq p_0} A_{p,w}, \\ &\forall e_f \in E_f, p_0 \in P, w \in W \end{aligned} \quad (20)$$

In particular, constraint (14) determines the aggregated OPS traffic that circulates through path p and wavelength w (variables $A_{p,w}$). Constraint (15) accounts for the limitations imposed by the QoS to the total outgoing OPS traffic from an OPS switch per output port (link) and wavelength. That is, the most restrictive bandwidth limit of all individual OPS flows circulating through that physical link and wavelength cannot be exceeded. To properly determine which is the imposed bandwidth limit, we have to know which are the virtual links that are currently circulating through a specific output port of an OPS switch and a particular wavelength. In order to do so, we utilize variables $C_{e_v,p,w}$. Another important point is that potential contention between packets in OPS, hence QoS degradation, happens among aggregated flows that share the same output port and wavelength but not the whole end-to-end path. Indeed, packets that go from the same source to the same destination can be serialized through electrical buffering at the source ToR to avoid contention. As long as the aggregate flow circulating through a particular path and wavelength is not sharing the same output port at the OPS switch with other aggregated flows, there will be no bandwidth limits due to QoS restrictions. To model this, we utilize variables $F_{e_f,p,w}$. Constraints (16) and (17) are the definition of variables $C_{e_v,p,w}$ while constraints (18)–(20) are the definition of variables $F_{e_f,p,w}$, with M and m being arbitrarily large and small positive numbers, respectively.

To better illustrate how these constraints work, let us put a small example. For this, let us consider a case with 3 virtual links requesting (B_{e_v}, $B^{max}_{e_v}$): (0.4, 0.85), (0.2, 0.8), (0.1, 0.9), respectively. Additionally, let us consider that the first and second virtual links are mapped over the physical path 1→2→3 while the third virtual link is mapped over the physical path 4→2→3, with node 2 being an OPS switch. For the purpose of the example, all virtual links are assumed to be mapped over the same wavelength. In such scenario, the aggregated flow per path and wavelength (variables $A_{p,w}$), will evaluate to 0.6 and 0.1 for the first and second paths, respectively. The respective variables $C_{e_v,p,w}$ will evaluate to 1, indicating that the particular virtual link employs the specific path and wavelength. As for variables $F_{e_f,p,w}$, they will evaluate to 1 for all paths and physical links except for physical link 2→3, which will evaluate to 0, since the virtual links circulating through that physical link are sharing the same wavelength and link, thus, are not employing alone the stated physical link and wavelength. With all of this, constraint (15), which determines the QoS restrictions according to the total traffic circulating through a particular output link and wavelength, will result in $0.7 \leq 0.8$ for physical link 2→3, which is an output link from an OPS switch, as 0.7 is the total flow circulating through that link and 0.8 is the most restrictive bandwidth limit imposed by the virtual links, in particular, virtual link 2. Thus, the QoS restrictions are properly bounded. In the case that the total flow would surpass the most restrictive bandwidth limit, the constraint would be violated, hence forcing that virtual links that share an output port at an OPS switch must be mapped over different wavelengths.

$$\sum_{e_f \in \delta^+(n_f)} \sum_{w \in W} t_{e_f,w} \leq L_{OCS}, \forall n_f \in N_c^{OCS} \quad (21a)$$

$$\sum_{e_f \in \delta^-(n_f)} \sum_{w \in W} t_{e_f,w} \leq L_{OCS}, \forall n_f \in N_c^{OCS} \quad (21b)$$

$$\sum_{e_f \in \delta^+(n_f)} \sum_{w \in W} t_{e_f,w} \leq L_{OPS}, \forall n_f \in N_c^{OPS} \quad (22a)$$

$$\sum_{e_f \in \delta^-(n_f)} \sum_{w \in W} t_{e_f,w} \leq L_{OPS}, \forall n_f \in N_c^{OPS} \quad (22b)$$

Constraints (21a)–(22b) are the port limit constraints, namely, they avoid having more incoming/outgoing active wavelengths at the switches (either OCS or OPS) than their port count.

$$z_{e_v,p,w} \leq S_{e_v,p,w}, \forall e_v \in E_v, p \in P, w \in W \tag{23}$$

$$S_{e_v,p,w} \leq M \cdot z_{e_v,p,w}, \forall e_v \in E_v, p \in P, w \in W \tag{24}$$

Constraints (23) and (24) are the definition of variables $S_{e_v,p,w}$.

$$\begin{aligned} &S_{e_{v_i},p_m,w} + x_{e_{v_j},p_n,w} \leq 1, \\ &\forall e_{v_i}, e_{v_j} \in E_v, e_{v_i} \neq e_{v_j}, p_m, p_n \in P_{e_f}, e_f \in E_f, w \in W \end{aligned} \tag{25}$$

$$\begin{aligned} &S_{e_{v_i},p_m,w} + S_{e_{v_j},p_n,w} \leq 1, \\ &\forall e_{v_i}, e_{v_j} \in E_v, p_m, p_n \in P_{e_f}, p_m \neq p_n, e_f \in E_f, w \in W \end{aligned} \tag{26}$$

Constraint (25) avoids mapping OCS and OPS flows at the same time over the same physical link and wavelength. Finally, constraint (26) avoids mapping two virtual links employing OCS and the same wavelength into the same physical link unless they share the whole end-to-end path. Such constraint allows us to model the OCS grooming capability that allows the aggregation of multiple virtual links onto the same wavelength (circuit) as long as they share the same physical path thanks to the electronic processing capabilities at both source and destination ToRs.

C. Heuristic Algorithm

Although the iterative approach of the MILP-based mechanism leads to better scalability when serving the requests of multiple tenants, its dependence on MILP may arise as problematic (in terms of execution time) when trying to solve bigger problem instances. For this reason, we also developed an heuristic mechanism in order to provide still accurate enough results at lower computational cost, making it an option when the scalability becomes challenging.

Algorithm 2 depicts a pseudo-code of this heuristic mechanism. Basically, it is structured in 2 phases, where in the second phase a multistart approach is adopted [19], introducing randomization at every iteration and returning at the end the best solution in terms of objective function. The parameter *multistart* controls the number of iterations of the multistart procedure. The first phase serves the same purpose as in the MILP-based mechanism: aggregate the virtual slice requests into one single graph representation for all the requests of a tenant inside the demand set and calculate the path set P. Then, the second phase starts by iteratively mapping the tenants' aggregated virtual slice request, one after another, into the physical DCN. In more detail, the mapping of the tenants is structured in three sub-phases: node mapping, link routing and wavelength assignment. Starting with the node mapping, we first map the virtual nodes of the largest (in terms of number of virtual nodes) virtual slice request (component) of the tenant. For this, a greedy procedure is adopted, with virtual nodes being mapped to the least loaded physical node that has enough room to allocate them. Regarding the virtual nodes of the rest of the components, they are randomly mapped on the subset of physical nodes previously employed to map the largest component. In this way, the intersection of virtual links will be larger, thus favoring the possibilities of saving resources due to flow aggregation or statistical multiplexing. Once the virtual nodes are mapped, virtual links should be mapped as well. For this, as a first step, the physical nodes over which the remote endpoints of every virtual link were mapped are connected through the shortest path in the DCN. Once a route has been assigned to every virtual link, their requested bandwidth should be mapped to actual wavelength channels.

```
Inputs: $D$, $G_n$, $W$, $L_{OCS}$, $L_{OPS}$, $multistart$; Outputs: $Sol$
Phase 1: Pre-processing
$D \leftarrow$ aggregate all virtual slice requests of a tenant into a single graph for each subset $d_i \in D$
$P \leftarrow$ set of path between all ($s$, $t$) pairs in $G_n$
$Sol \leftarrow \emptyset$
Phase 2: Tenant allocation
for $d = 1$ to $|D|$ do
    $G_d(N_v, E_v) \leftarrow$ graph representation of $d$
    $auxSol \leftarrow \emptyset$, $auxBestSol \leftarrow \emptyset$
    for $m = 1$ to $multistart$ do
        2.1: Node mapping
        $N_v^m \leftarrow$ virtual nodes of largest $N_v^t \in G_d$
        Map virtual nodes in $N_v^m$ balancing the load of $N_f$
        for $\forall n_v \in N_v \backslash N_v^m$ do
            Map virtual nodes randomly in subset of physical nodes assigned to $N_v^m$
        2.2: Link routing
        for $i = 1$ to $|N_v|$ do
            for $j = i + 1$ to $|N_v|$ do
                if virtual link ($n_i$,$n_j$) $\in E_d$ exists then
                    Find shortest path $p_{i,j} \in P$ according to physical mapping of virtual nodes
        2.3: Wavelength assignment
        2.3.1: Flow aggregation
        $F \leftarrow \emptyset$
        for $i = 1$ to $|N_f|$ do
            for $j = i + 1$ to $|N_f|$ do
                $R \leftarrow$ set of virtual links in $G_d$ for which their endpoints are mapped in physical nodes $n_i, n_j \in N_f$
                $F \leftarrow F \cup$ output from bin_packing($R$)
        2.3.2: Flow allocation
        for $i = 1$ to $|F|$ do
            $allocated \leftarrow$ false
            for $w = 1$ to $|W|$ and not allocated do
                if $w$ in selected path for $f_i \in F$ is empty then
                    allocate $f_i$ in $w$
                    Update physical resources availability
                    $allocated \leftarrow$ true
                else if enough bandwidth in $w$ and QoS is respected then
                    allocate $f_i$ in $w$
                    Update physical resources availability
                    $allocated \leftarrow$ true
        $auxSol \leftarrow$ tenant $d$ mapping
        if $Obj(auxSol) < Obj(auxBestSol)$ then
            $auxBestSol \leftarrow auxSol$
    $Sol \leftarrow Sol \cup auxBestSol$
Return $Sol$
Demands served
```

Algorithm 2: Heuristic mechanism pseudo-code.

The wavelength assignment phase must account for the possibility of both flow aggregation and QoS restrictions. Therefore, we have divided the process in two steps: one focusing on the flow aggregation aspect, while the other takes care of the wavelength assignment decision. Flow aggregation is a form of the more generic bin packing problem [20],

```
**Inputs:** $R$; **Outputs:** $F$
$binSet \leftarrow$ set of empty bins of size $|R|$
$B_{min} \leftarrow +\infty$; $m \leftarrow |R|$; $aux \leftarrow \emptyset$; $F \leftarrow \emptyset$
**packing**(0)
**for** $i = 1$ ***to*** $|aux|$ **do**
    **if** $aux_i$ *is not empty* **then**
        $BW \leftarrow 0$
        $QoS_{min} \leftarrow +\infty$
        **for** $j = 1$ ***to*** $|aux_i|$ **do**
            $BW \leftarrow BW+$ bandwidth of element $aux_{i,j}$
            **if** *QoS of element* $aux_{i,j} < QoS_{min}$ **then**
                $QoS_{min} \leftarrow$ QoS of element $aux_{i,j}$
        $F \leftarrow F\cup$ new aggregated flow with bandwidth $BW$ and
        QoS $QoS_{min}$
Return $F$
**Flows aggregated**
Function: **packing**($i$)
**if** $i$ *is equal to* $m$ **then**
    **if** *number of not empty bins in* $binSet < B_{min}$ **then**
        $B_{min} \leftarrow$ number of not empty bins in $binSet$
        $aux \leftarrow binSet$
    **End**
**else**
    $bw \leftarrow$ bandwidth of element $R_i$
    $lastEmpty \leftarrow$false
    **for** $k = 0$ ***to*** $m$ **do**
        **if** $binSet_k$ *is empty* **then**
            Add $R_i$ to $binSet_k$
            $lastEmpty \leftarrow$ true
            **packing**($i + 1$)
            Remove last element in $binSet_k$
        **else if** $bw$+*bandwidth allocated in* $binSet_k \leq 100\%$
        **then**
            Add $R_i$ to $binSet_k$
            **packing**($i + 1$)
            Remove last element in $binSet_k$
```

Algorithm 3: Bin_packing procedure pseudo-code.

where multiple elements of different sizes must be allocated into a set of bins, minimizing the number of bins employed. In our scenario, the elements to be allocated are the bandwidth of the virtual links and the bins are the wavelengths. To efficiently address the problem, we propose a specific procedure named **bin_packing** that takes all virtual links that share two endpoints as input, hence, the whole end-to-end route (we remind the reader that aggregation is done at the end-to-end level) and returns the set of aggregated flows that utilize the lowest number of wavelengths. Algorithm 3 depicts a pseudo-code for this procedure.

The mechanics of the procedure are based on applying recursively backtracking [21]. Essentially, it explores all aggregation possibilities and eventually returns a set of aggregated flows that entail the minimum number of required wavelengths (bins). The bandwidth of each aggregated flow results from the summation of the bandwidth of all virtual links aggregated into it and its associated QoS is the most restrictive of all of them.

After obtaining the aggregated flows, the algorithm proceeds with the wavelength assignment of these flows. For this, it employs a sequential first fit procedure. In particular, if the wavelength is empty, the aggregated flow is mapped without restrictions to that wavelength. On the other hand, if a previous aggregated flow has already been mapped into the wavelength under study, the algorithm checks if there is enough remaining capacity on the wavelength to serve the new aggregated flow. This being the case, then the algorithm checks if the mapping of the new aggregated flow will respect the QoS restrictions of the already mapped flows and itself (this is only done in the case that they would share the same output port at the OPS switch). If this condition is met, the aggregated flow is mapped over the wavelength under consideration. If not, or the wavelength does not have enough remaining capacity, the next wavelength is explored. After this process, all the virtual links are assigned a physical path and a wavelength that ensures both their bandwidth and QoS requirements. Note that aggregated flows that do not share wavelength with other aggregated flows will be mapped to OCS, since in such a case OPS does not provide any reduction on the number of employed Tx/Rx. On the other hand, aggregated flows that share the same wavelength (saving Tx/Rx) are mapped to OPS, since is the technology that allows for such situation. At this stage, the algorithm checks if the current mapping of the tenant leads to a better objective function than the best solution found so far. If so, this is registered as the best mapping for the particular tenant and the following iteration of the multistart procedure is executed.

Finally, the algorithm proceeds to repeat the whole mapping process for the virtual slices of the next tenant in the demand set. Particularly, all wavelengths employed are made unavailable for the next tenants in the following iterations of the mechanism, guaranteeing in this way that virtual links belonging to different tenants are physically isolated. The average time complexity of the proposed heuristic, considering that a breadth-first search is utilized for route calculation and an exhaustive search is utilized for the flow aggregation, can be stated as $\Theta(|D| \cdot multistart \cdot |\overline{N_v}|^2 \cdot |E_f| \cdot (\sum_{\forall t} \frac{2|E_v^t|}{|N_v^0|(|N_v^0|-1)})! \cdot |W|)$, where $|\overline{N_v}|$ is the average number of virtual nodes per tenant and the expression in the factorial accounts for the average number of flows that may be aggregated from a source to a destination in the bin packing. Note that, although there is a factorial term (due to the exhaustive search), in practice such term is quite small, even for relatively big-sized instances, resulting in few traffic flows to be aggregated. Hence, in general, the proposed heuristic execution times stay largely below the MILP formulation, as it will be shown in the following section.

IV. Results and Discussion

In this section, we will evaluate the performance of both the MILP-based and heuristic mechanisms. In order to quantify the benefits provided by a hybrid OCS/OPS DCN when optimally mapping virtual slice requests in a multi-tenant scenario, we have run extensive simulations, utilizing as a benchmark the case where only a pure OCS DCN is considered. At this point, it has to be said that we have utilized the pure OCS case as a benchmark since a pure OPS DCN is still highly unlikable in the near-middle future due to its technical complexity and higher cost. The comparison between the thee options (OCS, OPS and hybrid) in terms of performance and cost is left for future studies. In order to perform a fair comparison, we utilized the same optimization mechanisms as in the hybrid case for modeling the pure OCS case. For the MILP, we fix variables Z_{e_v} to 1 so all virtual links are forced to be served employing OCS. As for the heuristic mechanism, we add the restriction that virtual

TABLE I
MILP VS HEURISTIC

		MILP			Heuristic			
Scenario	Slices	Tx	Rx	Time (s.)	Tx	Rx	Time (s.)	Gap (%)
Hybrid	1	5.4	6.3	$> 8.7 \cdot 10^4$	6	6.5	$25 \cdot 10^{-3}$	6.8
	2	9.1	9.8	$> 8.7 \cdot 10^4$	9.4	9.8	$26.6 \cdot 10^{-2}$	1.6
	3	11.8	12.4	$> 8.7 \cdot 10^4$	12.3	12.6	$56.4 \cdot 10^{-2}$	2.9
OCS	1	6.6	6.6	$> 8.7 \cdot 10^4$	6.6	6.6	$48.3 \cdot 10^{-3}$	0
	2	9.8	9.8	$> 8.7 \cdot 10^4$	9.8	9.8	$46.9 \cdot 10^{-2}$	0
	3	12.4	12.4	$> 8.7 \cdot 10^4$	12.6	12.6	$59 \cdot 10^{-2}$	1.6

links can only share a wavelength in the same physical link if they share the whole end-to-end path. Moreover, we set for all virtual links a bandwidth limit imposed by QoS restrictions equal to the entire capacity of a wavelength, so as to recreate the conditions of a pure OCS DCN. For all the experiments through this section, the *multistart* parameter has been set to 1000. Moreover, all the results in this section have been executed in i7 CPUs at 3.4 Ghz with 16 GB of memory, utilizing the solver CPLEX v12.5 [22] for solving the MILP formulations.

First, we will compare the performance of the heuristic against the MILP. For this, we have focused on a limited network scenario consisting on a cluster of 6 racks, with a single intra-cluster AoD OCS switch and a single OPS switch. In this scenario, we consider that the servers present in the racks have enough capacity to host all virtual slice requests. That is, there is no limit on the aggregated VM capacity of the whole rack. Additionally, we consider that both OCS and OPS switches do not have limitations in their port count and can switch an arbitrarily large number of wavelengths. As for the demands, we consider the presence of a single tenant requesting between 1 and 3 virtual slices. The generation of the virtual slices follows a random process structured basically in 2 steps. First, the nodes of the virtual slice and their required capacities are generated. For this, we generate between 2-5 nodes with the same probability with capacities ranging from 1 to 10 VMs. Second, virtual nodes are randomly connected using the Erdős-Rényi algorithm [23], here slightly modified to prevent the generation of non-connected graphs, since virtual slices inside a tenant must be connected graphs. Nevertheless, the tenant aggregated virtual slice may be a non-connected graph when composed. The parameter p of the algorithm is set to 0.5, which leads to the generation on any connectivity matrix with the same probability. The requested bandwidth of the virtual links is uniformly chosen between 10 and 100% of the capacity of a wavelength in steps of 10%. As for the bandwidth limits due to QoS restrictions, they are chosen among the set {60, 64, 70}% to reflect a scenario where different classes of services co-exist.

Table I compares the performance of both MILP-based and heuristic mechanisms in terms of utilized Tx/Rx, execution time and relative gap in the objective function for the hybrid OCS/OPS and pure OCS DCN cases and different numbers of requested virtual slices by the tenant. The obtained results have been averaged over 10 executions, randomly generating a new instance for each execution. To perform a consistent comparison, we utilize the same problem instance for both the MILP-based and heuristic mechanisms as well as for both hybrid OCS/OPS and OCS DCN scenarios. It can be appreciated that the results obtained with the heuristic mechanism are very close to the ones obtained with the MILP-based mechanism, with relative gaps on the objective function ranging from 0 to 7%, hence, highlighting its accuracy. As for the execution times, we can see that, although the MILP-based mechanism requires execution times larger than a day, the execution times of the heuristic remain in the sub-second range. Thus, the heuristic succeeds in providing accurate results in much less time when compared to the MILP-based mechanism.

Once we assessed the accuracy of the heuristic, we will analyze the benefits of the hybrid DCN solution against a pure OCS DCN. Since the potential benefits depend on the characteristics of the requests, we will study how the number of necessary Tx/Rx evolves in both cases according to specific parameters of the virtual slices. All the following results have been extracted utilizing the proposed heuristic mechanism and the same procedure for the generation of the virtual slices explained before, as well as 100 random repetitions per data point in order to obtain statistically relevant average results. The particular details will be noted for each simulation. As for the network topology, we have focused on a scenario with 4 clusters with 8 racks each. Such values have been selected to reflect common DC infrastructures found in the literature (e.g. [24]). Each cluster is connected to an inter-cluster AoD switch enabling the communication between clusters. Like before, we are considering that there are no limits in the number of VMs a server can host nor in the switching capacity of the OCS or OPS switches. Finally, all results have been obtained assuming the presence of 50 tenants, for which every tenant is requesting between 1 and 5 virtual slices with equiprobability. In all cases, the execution times of the heuristic stay lower than 10 s.

To reflect the performance of the proposed solution against different traffic patterns, we have analyzed how the total number of Tx/Rx changes as a function of the share of mice and elephant traffic respect the total traffic. For this, we define as mice and elephant traffic the virtual links that are requesting a bandwidth between 10-40% and 50-100% in terms of wavelength capacity, respectively. Then, we vary the percentage of virtual links of each type. Figure 3 shows the evolution of the total number of Tx/Rx against the share of mice traffic respect the total traffic. It can be appreciated that for low shares of mice traffic, that is, almost all the virtual links correspond to elephant traffic, the differences between the hybrid solution and the pure OCS case are low (around 0.2-5%). This is due to the fact that most of the traffic neither can be aggregated in the same wavelength nor enjoy the multiplexing property of OPS. On the other hand, for high shares of mice traffic, substantial reductions (up to 35%) can be appreciated. This is because more virtual links may share a single physical link thanks to the statistical multiplexing property of OPS, reducing the necessary Tx/Rx to be equipped at the ToRs.

Another important parameter is the bandwidth limit imposed by QoS restrictions in OPS. To analyze this aspect, we have fixed the bandwidth limit per virtual link and obtained the necessary number of resources in the DCN for increasing values of it. In this case, the bandwidth requested per virtual link is chosen between 10-100%. Figure 4 shows the obtained results. The x axis represents the bandwidth limit imposed by QoS restrictions. As expected, higher QoS bandwidth limits allow for further reductions in the necessary number

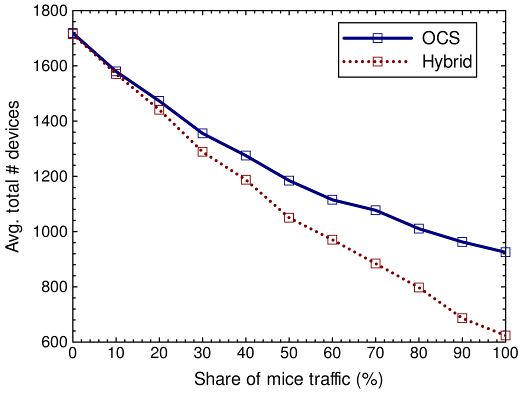

Fig. 3. Evolution of the number of Tx/Rx devices in the DCN as a function of the share of mice traffic.

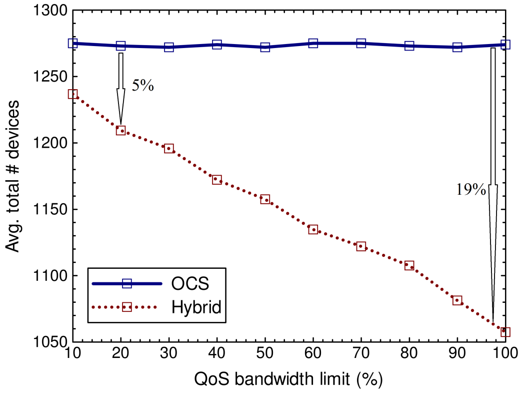

Fig. 4. Evolution of the number of Tx/Rx devices in the DCN as a function of the QoS bandwidth limit.

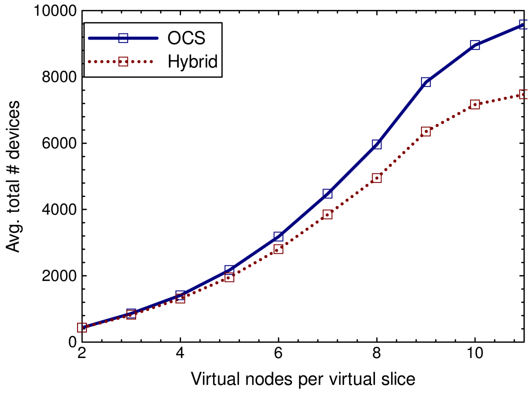

Fig. 5. Evolution of the number of Tx/Rx devices in the DCN as a function of the number of virtual nodes per virtual slice.

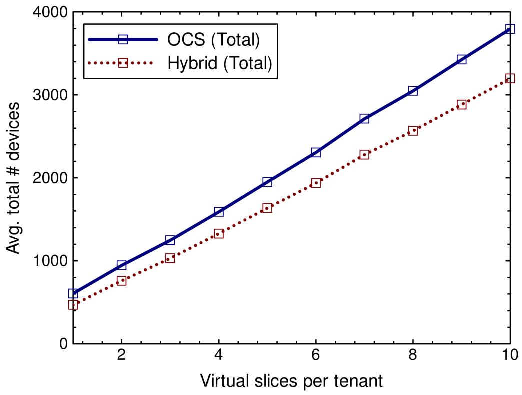

Fig. 6. Evolution of the number of Tx/Rx devices in the DCN as a function of the number of virtual slices per tenant.

of Tx/Rx devices, since more virtual links benefit from the statistical multiplexing properties of OPS as more load can be packed in a wavelength without surpassing the QoS limits. In particular, we can appreciate around 3-5% reductions for low bandwidth limits while the reductions increase up to around 20% for higher limits. In this regard, we can say that a hybrid solution becomes interesting for traffic flows that do not need stringent QoS restrictions, that is, allow a higher load limit per wavelength. Nevertheless, some benefits are also obtained for more restrictive QoS limits when compared to a pure OCS solution.

Next, we also analyzed the influence of the mesh degree of the virtual slices on the necessary number of Tx/Rx devices. This is also particularly relevant, since a more meshed scenario (with more nodes and/or links) means that multiple virtual links would share the same source or the same origin, thus allowing the possibility to reduce the necessary number of Tx/Rx devices due to the statistical multiplexing property of OPS. On the other hand, in a pure OCS solution, the chances of two virtual links sharing the whole end-to-end path are lower so the potential aggregation of virtual links in the same circuit is reduced. For this, we have modified the number of virtual nodes of the virtual slices since when maintaining the probability of interconnection between virtual nodes, the presence of more virtual nodes means that more virtual links will be present, hence, a more meshed virtual slice is realized. With this, Figure 5 shows the evolution of the needed optical devices as a function of the number of virtual nodes per virtual slice, which has been fixed a priori.

It can be seen that for a low number of virtual nodes, the differences between the hybrid solution and the OCS solution are small (around 5-7%) but they grow when increasing the number of virtual nodes per virtual slice, rising up to around 20-25% reductions. This is due to the aforementioned reason: more meshed scenarios benefit from the statistical multiplexing property of OPS. In this regard, we can see that a hybrid solution becomes interesting in scenarios with a large number of nodes and nodal degree, leading to substantial reductions in the necessary optical Tx/Rx devices to be equipped at the DCN.

To conclude our studies, we also analyzed how the necessary number of Tx/Rx devices evolves with the number of virtual slice requests per tenant. A larger amount of virtual slices can allow for more resource sharing between virtual slices of the same tenant, either thanks to grooming in OCS or statistical multiplexing in OPS. For this, we have fixed the number of virtual slice requests per tenant, ranging from 1 to 10. Figure 6 depicts the obtained results. Interestingly, although it can be seen that the absolute differences between the two solutions grow with the number of virtual slices per tenant, the relative gains between the hybrid OCS/OPS and the pure OCS DCN decrease with the number of virtual slices per tenant (from around 20% to around 10%). This mainly happens because, with more slices, more virtual links have to be mapped onto optical channels. In such a situation, the relative gains are less significant when compared to low traffic conditions, where saving few Tx/Rx devices account for substantial reductions on the average number of needed Tx/Rx devices.

V. Conclusions

In this paper, we have shown the importance of optimally allocating virtual slices in a DC given a multi-tenant scenario. It allows for an efficient utilization of the underlying physical infrastructure, saving costs and potentially increasing the revenues of the DC operator. As a case study, we have focused on an all-optical hybrid OCS/OPS solution for the DCN following the proposal of the LIGHTNESS project. Through extensive results, we have shown that such a hybrid DCN can save resources when compared to a pure OCS DCN solution.

To better highlight the benefits of the hybrid solution, we have analyzed different relevant aspects of the virtual slice requests. We have seen that substantial reductions (20-35%) can be achieved in the situations where a significant share of mice traffic flows are present, virtual links do not require very strict QoS or the size of the virtual slice is big, thanks to combining the statistical multiplexing properties of OPS with the grooming capacity of OCS. Nevertheless, it also provides resource savings in less favorable situations, reveling itself as a very versatile and promising solution for future DCs.

Acknowledgment

The work in this paper has been supported by the EC FP7 LIGHTNESS (FP7-318606) and COSIGN (FP7-619572) projects.

References

[1] Cisco, "Cisco Global Cloud Index: Forecast and Methodology 20132018 White Paper", http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.html

[2] T. Wang, Z. Su, Y. Xia, M. Hamdi, "Rethinking the Data Center Networking: Architecture, Network Protocols, and Resource Sharing", IEEE Access, vol. 2, pp. 1481-1496, September 2014.

[3] C. Kachris, I. Tomkos, "A Survey on Optical Interconnects for Data Centers", IEEE Communications Surveys and Tutorials, vol. 14, no. 4, pp. 1021-1036, January 2012.

[4] G. Wang et al., "c-Through: Part-time Optics in Data Centers", Proceedings of ACM Special Interest Group on Data Communication 2010 (SIGCOMM 2010), September 2010.

[5] N. Farrington et al., "Helios: A Hybrid Electrical/Optical Switch Architecture for Modular Data Centers", Proceedings of ACM Special Interest Group on Data Communication 2010 (SIGCOMM 2010), September 2010.

[6] A. Singla et al., "Proteus: a topology malleable data center network", Proceedings of ACM Special Interest Group on Data Communication 2010 (SIGCOMM 2010), September 2010.

[7] Y. Yin et al., "LIONS: An AWGR-Based Low-Latency Optical Switch for High-Performance Computing and Data Centers", IEEE Journal of Selected Topics in Quantum Electronics, vol. 19, no. 2, April 2013.

[8] X. Zhao et al., "The prospect of inter-data-center optical networks", IEEE Communications Magazine, vol. 51, no. 9, pp. 32-38, September 2013.

[9] J. Perelló et al., "All-Optical Packet/Circuit Switching-Based Data Center Network for Enhanced Scalability, Latency, and Throughput", IEEE Networks, vol. 27, no. 6, pp. 14-22, December 2013.

[10] T. Benson, A. Akella, D.A. Maltz, "Network Traffic Characteristics of Data Centers in the Wild", Proceedings of the 10th ACM SIGCOMM conference on Internet measurement (ICM 2010), November 2010.

[11] K.-K. Nguyen, M. Cheriet, Y. Lemieux, "Virtual Slice Assignment in Large-Scale Cloud Interconnects", IEEE Internet Computing, vol. 18, no. 4, pp. 37-46, July 2014.

[12] L. Nonde, T. El-Gorashi, J. Elmirghani, "Energy Efficient Virtual Network Embedding for Cloud Networks", Journal of Lightwave Technology, vol. 33, no. 9, pp. 1828-1849, May 2015.

[13] S. Peng et al., "Enabling Multi-Tenancy in Hybrid Optical Packet/Circuit Switched Data Center Networks", Proceedings of 40th European Conference on Optical Communications, Tu.1.6.4, September 2014.

[14] N. Amaya, G. Zervas, D. Simeonidou, "Architecture on demand for transparent optical networks", Proceedings of 13th International Conference on Transparent Optical Networks (ICTON 2011), June 2011.

[15] S. Peng et al., "A Novel SDN enabled Hybrid Optical Packet/Circuit Switched Data Centre Network: the LIGHTNESS approach", Proceedings of 23rd European Conference on Networks and Communications (EuCNC 2014), September 2014.

[16] Z. Keyao, B. Mukherjee, "Traffic grooming in an optical WDM mesh network", IEEE Journal on Selected Areas in Communications, vol. 20, no. 1, pp. 122-133, January 2002.

[17] A. Borella, F. Chiaraluce, F. Meschini, "Statistical multiplexing of random processes in packet switching networks", IEE Proceedings-Communications, vol. 143, no. 5, pp. 325-334, October 1996.

[18] N. Calabretta, R. Pueyo, S. Di Lucente, H.J.S. Dorren, "On the Performance of a Large-Scale Optical Packet Switch Under Realistic Data Center Traffic", IEEE/OSA Journal of Optical Communications and Networking, vol. 5, no. 6, pp.565-573, June 2013.

[19] C. Tsai, K. Hu, M. Chiang, "A multiple-search multi-start framework for metaheuristics", Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC), October 2014.

[20] B. Chazelle, "The Bottom-Left Bin-Packing Heuristic: An Efficient Implementation", IEEE Transactions on Computers, vol. C-32, no. 8, pp. 697-707, August 1983.

[21] V.N. Rao, V. Kumar, "On the efficiency of parallel backtracking", IEEE Transactions on Parallel and Distributed Systems, vol. 4, no. 4, pp. 427-437, April 1993.

[22] http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

[23] P. Erdős, A. Rényi, "On random graphs", Publicationes Mathematicae, vol. 6, pp. 290-297, 1960.

[24] M.D. Hill et al., "High Performance Datacenter Networks: Architectures, Algorithms, and Opportunities", Ed. Morgan & Claypool, 2011.