

VNF Service Chaining in Optical Data Center Networks

Vikrant Nikam⁽¹⁾, James Gross⁽¹⁾ and Ahmad Rostami⁽²⁾

⁽¹⁾ KTH Royal Institute of Technology, Sweden, ⁽²⁾ Ericsson Research, Sweden
nikam@kth.se, james.gross@ee.kth.se, ahmad.rostami@ericsson.com

Abstract—Virtualized network function (VNF) service chaining in optical datacenter networks (DCN) is a more complex problem than that in packet-switched networks, as it introduces additional constraints related to the optical network. For example, in an optical DCN one needs to make sure that optical network resources are efficiently utilized, which requires multiplexing of several VNF chains to fill the optical pipes. In this paper, we first propose a novel and flexible DCN architecture based on optical circuit switching technology supporting service chaining in the optical domain. Then, we formulate the problem of VNF service chaining in the proposed optical DCN using integer linear problem (ILP) formulation and heuristic methods. We also numerically investigate the performance of the proposed architecture and the service chaining methods using a set of examples.

Keywords—SDN; NFV; Service Chaining; Optical Data Center Network

I. INTRODUCTION

Datacenters (DCs) consist of thousands of servers hosting many rapidly growing cloud services. A large amount of aggregate bandwidth is required to accommodate communication to/from a DC as well as among the servers within the DC. Typically, the required connectivity among servers within a DC is realized through electrical packet switching (EPS) networks with multilayer topologies. However, there are several limitations to existing EPS technology and architectures including the need for a large number of switches and the associated cost and complexity, limited throughput, and high power consumption. Dynamic optical circuit switching (OCS) can play an important role in datacenter networks (DCN) as it can support on-demand high bit-rate connectivity with power consumptions considerably lower than that of EPS. For example, Porter *et al.* [1] proposed to build a hybrid DCN combining EPS and OCS, where large flows are routed through OCS and smaller flows are routed through EPS. The efficient utilization of the OCS network in a DCN requires smart and flexible control mechanisms, among other things, for dynamically aggregating smaller flows into optical pipes.

On the other hand, recent trends in adopting programmable networking technologies (e.g., software defined networking, SDN and network function virtualization, NFV) have enabled the network flexibility in multiple dimensions. Specifically, NFV enables virtualization of functionality in network elements and running them (virtualized network functions, VNFs) on commodity servers in operator's DCs [2]. This, in turn, facilitates dynamic creation, adaptation and termination

of networking services, realized in form of chained graphs of VNFs, i.e., VNF service chaining. Software-defined networking (SDN) further facilitates this novel architecture by dynamic steering of traffic through components of a service [3, 4]. Although lots of research efforts have been put into VNF service chaining in EPS networks, little has been achieved when it comes to optical DCN. The VNF service chaining in optical DCN is a more complex problem than that in packet switched networks, since there are additional constraints that need to be considered. Two such constraints are a) VNF service chain multiplexing to fill up the optical channels, b) optical network control algorithms (i.e. routing and wavelength assignment constraints). In this paper, we try to address these challenges by a) proposing a novel and flexible DCN architecture based on OCS technology that inherently supports service chaining in the optical domain, b) formulating the VNF service chaining problem on top of the proposed network architecture as integer linear program (ILP), and c) presenting a new heuristic algorithm to solve the ILP.

The rest of this paper is organized as follows. Section II presents the system model including proposed optical DCN architecture and problem definition. Section III presents mathematical formulation of optical service chaining together with corresponding heuristic algorithms. In Section IV, we numerically evaluate the models. Section V presents related works, and finally in Section VI we conclude the paper by summarizing the achieved results.

II. SYSTEM ARCHITECTURE

A. Modular Optical Data Center Network Architecture

Fig.1 illustrates an overview of the proposed optical data center network architecture. It is based on optical circuit-switching using the wavelength division multiplexing (WDM) technology. The network can be complemented with an EPS segment, though here we only focus on the OCS segment. The DCN is connected to outside through a packet-based gateway router, which also serves as aggregation point of incoming traffic flows for forwarding them into optical pipes, and vice versa. This requires the gateway router to be equipped with (tunable) WDM transceivers. The DC includes one or more cascaded pods. Fig.2. illustrates the detailed data plane architecture of each pod consisting of two 1:3 multi-directional wavelength selective switches (MD-WSS) [5] and several server racks. Each server rack has a top of rack (ToR) switch, which is equipped with one or more tunable WDM transceivers. The MD-WSS is the main building block of this

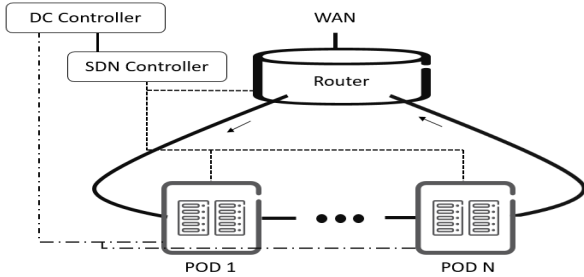


Figure 1. Modular optical data center architecture overview

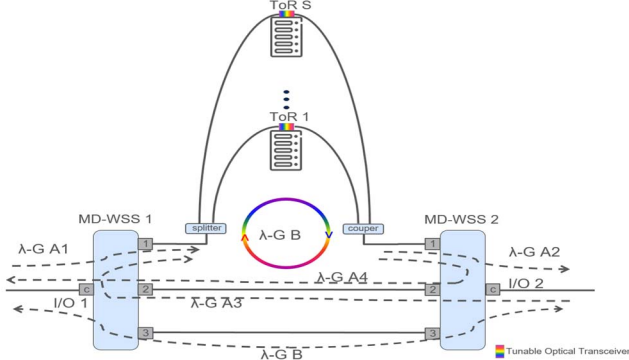


Figure 2. Pod internal architecture

optical DCN architecture, which interconnects the server racks intra and inter-pods. In particular, an MD-WSS allows any group of wavelengths that are routed between the common port (c) and tributary port 3 to be also reused between the tributary ports 1 and 2 [5]. We utilize this unique feature to realize optical interconnections among server racks within a given pod as well as among the same pod and other pods, using the same wavelength(s). The broadcast and select technique is used with one optical coupler and optical splitter to carry the traffic between the MD-WSS and ToR switches. Input/Output (I/O) ports at the common port of both MD-WSSes allow I/O traffic to/from each pod. The wavelength space is divided into two groups A and B, which are used to support different types of traffic. Specifically, wavelengths from group A are used for pod I/O traffic and it is further divided into subgroups A1-A4. Wavelengths belonging to group B are used for bypass traffic (passing through common port & tributary port 3) as well as for intra-pod traffic (passing through tributary ports 1&2). The routing of intra-pod traffic is achieved by a (virtual) ring type connectivity created between ToR switches, the optical splitter and the coupler through tributary ports 1 & 2 of both MD-WSSes. The incoming traffic at port I/O 1 destined to any server in the pod is routed to tributary port 1 of the MD-WSS 1 and then to the splitter. The traffic is broadcasted to all ToR switches. The tunable WDM transceiver at the ToR of a destination server rack is tuned to receive one of the wavelengths from group A1 on which traffic arrives. The modularity of the architecture enables extending the DC capacity by adding new pods, which might also require adding optical amplifiers between pods.

A centralized software defined network (SDN) controller steers traffic within the DC. It interfaces to and dynamically configures network elements including aggregation router and its transceivers, ToR switches, as well as MD-WSSes. The

controller is also responsible for topology computation and dynamic provisioning of optical circuits. Also, a DC controller, on top of the SDN controller, is responsible for compute and storage resource provisioning in a DC, i.e. for server allocation to VNFs.

B. VNF Service Chaining in Optical DCN

We leverage the inherent wavelength routing features in the presented DCN network to realize VNF service chaining directly at the optical layer. The chaining problem can be divided into three sub-problems:

1) *VNF chains multiplexing*: to minimize the required scarce wavelength resources, several VNF chains can be multiplexed over a single WDM channel. The multiplexing can be performed depending on service chains' data rates & CPU¹ requirements as well as on the availability of resources in terms of link bandwidth and system CPU.

2) *VNF placement*: for each service request, corresponding VNFs must be placed on the servers (across one or more racks) and be chained according to the requested order.

3) *Routing and wavelength assignment (RWA)*: to chain VNFs of a service (or a group of multiplexed services), an end-to-end optical path should be established taking into account the DCN architecture and wavelength continuity constraints.

The sub-problems need to be solved such that the processing and networking resources (i.e., available wavelengths and transceivers) are optimally utilized. The challenge here is that the sub-problems are not fully independent from each other. The multiplexing of chains has an impact on the VNF placement as well as on the usage of the number of wavelength resources. While multiplexing chains, one should consider the VNF placement—in addition to the system resources information—as the group of service chain may be distributed across server racks. Similarly, for VNF placement, RWA constraints of DCN architecture need to be considered. Solving VNF placement problem individually may end up in an inefficient use of wavelength resources.

III. PROPOSED VNF SERVICE CHAINING SOLUTION

The complexity of each sub-problem and inter-dependency between them makes it quite difficult to derive a single model that provides an end-to-end solution for the VNF service chaining. Therefore, in this paper we adopt an approach, where we solve the three sub-problems sequentially in three steps while at each step we take into account the available system resources and the next sequential sub-problem.

A. VNF Service Chain Multiplexing

The multiplexing of service chains is a special case of the bin-packing problem, where VNF service chains need to be grouped into the minimum number of groups considering system CPU and link capacities. However, the placement-

¹ For simplicity, we consider only CPU as an indicator of processing resource requirements.

awareness adds new dimensions to the classical bin-packing problem. Accordingly, we argue that the VNF service chain multiplexing is an NP-hard problem [6]. We therefore first formulate a simplified ILP and then present a heuristic to solve the multiplexing problem.

To multiplex service chains while considering the capacity and placement constraints following conditions needs to be satisfied: a) combined data-rate demands of all service chains multiplexed into one group must be less than or equal to the link capacity, b) the total number of server racks required to place a group of multiplexed VNF service chains must be less than or equal to service chain length with least number of VNFs, and c) the combined CPU requirement of first VNF from each service chain in a group must not exceed the server rack's CPU capacity, and the combined CPU requirement of last VNF from each VNF service chain in a group must not exceed the server rack's CPU capacity. The condition 'c' ensures that same input and output connection paths are used for all service chains in a group by placing first VNF from each chain of the group on one server rack, and the last VNFs from each chain of the group on one server rack.

For the ILP formulation we make a simplifying assumption that all chains consist of equal number of VNFs. This partly removes the complexity of placement-awareness related to chains with unequal number of VNFs, which would require complex constraints to maintain order of VNFs and ensure same optical paths for all service chains in a group. As a result, we only consider CPU and link requirements for VNF placement. This simplified model will later allow us to evaluate the performance of the proposed heuristic.

Let S and G be the set of input service chain requests and the set of possible groups of multiplexed chains, respectively. $X_{i,j}$ is a binary variable that is set to one if VNF service chain i belongs to group j , where $i \in S$ and $j \in G$. Y_j is a binary variable that is set to one if a group j is used. Input parameters for the ILP are as follows: the number of VNFs in service chains to be multiplexed (N_f), CPU capacity per server rack (C_r), Link capacity (L), CPU resource demand of chain i (C_i) and data-rate demand of service chain i (D_i). The ILP formulation is presented below,

$$\min \sum_j Y_j \quad (1)$$

$$\forall j \in G \ \& \ i \in S \quad \sum_i X_{i,j} \cdot C_i \leq N_f \cdot C_r \quad (2)$$

$$\forall j \in G \ \& \ i \in S \quad \sum_i X_{i,j} \cdot D_i \leq L \quad (3)$$

$$\forall i \in S \ \& \ j \in G \quad \sum_j X_{i,j} = 1 \quad (4)$$

$$\forall i \in S, \forall j \in G \quad X_{i,j} \leq Y_j \quad (5)$$

The objective in (1) is to multiplex the VNF service chains into a minimum number of groups. Constraint (2) ensures that the sum of CPU requirements of grouped chains does not exceed CPU rack capacity. Constraint (3) ensures that the sum of data-rate demand of VNF service chains to be grouped does not exceed the link capacity. Constraint (4) ensures that each

VNF service chain belongs to only one multiplexed group. And finally, constraint (5) ensures that if service chain i is assigned to group j then group j is used.

A heuristic for multiplexing the chains is presented in Algorithm 1. The algorithm uses decreasing best-fit approach to find the possible chains that can be multiplexed based on the link capacity and data-rate demand per chain. It first sorts the input chains S by descending order of corresponding data rate demands. Then for each chain, the algorithm filters other chains that can be multiplexed based on available link capacity to satisfy condition 'a' (line 5). After identifying the possible chains that can be multiplexed, placement and CPU capacity constraints are verified for these chains. To group two chains, the algorithm verifies the placement criteria 'b' for the number of racks required for placement of the chains (line 10) and checks the CPU requirements 'c' (line 11). If all the conditions are satisfied, the heuristic groups service chains together. For each multiplexed group, it calculates and assigns following metrics: the number of VNF service chains multiplexed, the number of server racks needed to place all chains, link utilization, and CPU utilization (line 13). The algorithm also ensures no two groups share the same set of service chains.

Algorithm 1: Heuristic for multiplexing VNF service chains

Data: S = VNF Service Chain List, L = Link Capacity, C_r = Server Rack CPU Capacity, Z = Max. No. of chains in a group and MPG = Max. permissible groups per chain

```

1. Sort  $S$  in descending order of  $s^d$ ;
2. muxSCList = null
3. foreach  $s \in S$  do
4. |  $l$  = calculateAvailableLinkCapacity( $s, L$ );
5. | tempMuxList = findSCForMultiplexing( $s$  [1...end],  $l$ )
6. | if (! tempMuxList.isEmpty()) then
7. | | Sort tempMuxList in random order
8. | | foreach  $t \in$  tempMuxList do
9. | | | if (! isMPGLimitReached( $t, MPG$ )) then
10. | | | |  $cond1$  = checkNoOfServerRacksCriteria( $s, t$ );
11. | | | |  $cond2$  = checkCPUCriteria( $s, t, C_r$ );
12. | | | | if ( $cond1 \wedge cond2$ ) then
13. | | | | | multiplexedGroup = multiplexSC( $s, t$ );
14. | | | | | add(muxSCList, multiplexedGroup);
15. | | | | else
16. | | | | | dropServiceChain( $t$ );
17. | | foreach  $m \in$  muxSCList do
18. | | | if (! isMuxLimitReached( $m, Z$ )) then
19. | | | |  $l$  = calculateAvailableLinkCapacity( $m, L$ );
20. | | | | tempList = getAvailableChainsForMultiplexing
    | | | | ( $m, s$  [1...end],  $l$ );
21. | | | | foreach  $t \in$  tempList do
22. | | | | | Repeat steps 9 to 16 (replacing  $s$  by  $m$ )
23. | | dropServiceChain( $s$ );

```

The algorithm also iterates over the list of multiplexed service chains to find more possible service chains that can be added to the multiplexed group to achieve maximum link and CPU utilization (line 21-23). To improve the performance of the algorithm, there is a limit on the maximum number of

service chains that can be grouped together (line 18). Also, to limit the size of the result set, a limit is set on the number of occurrences of an individual VNF chain in multiple multiplexed groups (line 9).

B. VNF Placement

Placement of VNF chains in the DC is also a special case of bin-packing problem. If we consider a server rack as a bin, then we need to pack service chains into the minimum number of server racks taking into consideration the total CPU capacity of the server racks. However, after multiplexing several VNF service chains, additional constraints need to be applied to ensure that VNFs are placed such that requested order of VNFs per chain is maintained and the same end-to-end optical connection are used by all service chains from each multiplexed group. Thus, the optimal VNF placement problem is also NP-hard.

The proposed DCN architecture allows to reuse the set of wavelengths carrying intra-pod traffic to carry bypass traffic as well. It also provides flexibility to use same set of wavelengths to carry incoming and outgoing traffic to/from each pod. Wavelengths usage can be optimized if VNF placement algorithm exploits the wavelength reusability feature. Let us consider a simple example where we have 2 pods P1 with server racks R1, R2 and P2 with 2 server racks R3, R4 in the DC and a request to place a multiplexed service chain MC that requires 2 server racks. If we place MC such that it uses 2 server racks from same pod P1 then the connection path would be as follows, InPort→R1→R2→OutPort. This will require two wavelengths (for I/O traffic and intra-pod traffic). On the other hand, if we distribute VNFs of MC across server racks of pod P1 and P2, the connection path would be InPort→R1→R3→OutPort. In this case, the end-to-end flow can be established using only one wavelength as I/O connection from each pod can reuse the same wavelength.

We propose a heuristic for VNF placement in the proposed optical DCN taking into consideration RWA constraints. The basic idea behind our heuristic is to implement a hybrid approach such that VNFs are distributed across the server racks of different pods to minimize the number of wavelengths needed. If VNFs cannot be distributed across the pods due to the limited number of pods, VNFs are placed on different racks within the same pod.

The output of VNF service chain multiplexing algorithm is further used for VNF placement. The result set of multiplexing algorithm provides metrics for each multiplexed group that is used for the optimal multiplexed group selection. VNF placement heuristic sorts the multiplexed chain groups in ascending order of link utilization per group so that number of wavelengths needed to serve chains is minimized. Multiplexed service chain groups are placed sequentially in the DC. While placing the multiplexed group, if it requires more than one server rack for placement then it distributes VNFs across server racks of different pods in the DC. If distribution across pods is not possible then it places them in the same pod. Whenever a multiplexed chain group is placed in the DC, all other groups that contain any of the VNF service chain from placed group are discarded.

Algorithm 2: Placement of VNFs from multiplexed service chain group

Data: M = Multiplexed VNF Service Chain list, P= Pod list

```

1. foreach m ∈ M do
2.   | chains = getChainsFromMultiplexedGroup(m);
3.   | noOfRacksNeeded = getNoOfRackedNeeded(chains)
4.   | p = getPodFromPodList(P);
5.   | serverRacks = getServerRacksFromPod(p);
6.   | foreach c ∈ chains do
7.     | NFList = getVNFList(c);
8.     | if (size(NFList) == noOfRacksNeeded) then
9.       | | foreach r ∈ serverRacks do
10.        | | | placeNFOnServerRack(NFList(0), r);
11.        | | | remove(NFList(0));
12.        | | | remove(c);
13.     | foreach r ∈ serverRacks do
14.       | | foreach c ∈ chains do
15.         | | | NFList = getVNFList(c);
16.         | | | placeNFOnServerRack(NFList(0), r);
17.         | | | remove(NFList(0));
18.         | | | while (getRackCapacity(r) > 0) do
19.           | | | foreach c ∈ chains do
20.             | | | | NFList = getVNFList(c);
21.             | | | | if (size(NFList) > (size(serverRacks)-
22.               | | | | | indexOf(r))) then
23.               | | | | | if (NFList(0).CPUReq <= getRackCapacity(r))
24.                 | | | | | placeNFOnServerRack(NFList(0), r);
25.                 | | | | | remove(NFList(0));

```

The Algorithm 2 explains the distribution of VNFs per chain to meet the following requirements: a) order of VNFs per chain is maintained, and b) each rack contains at least one VNF from each chain (if distributed on multiple racks). The simple scenario for VNF placement across multiple rack is that a service chain in a multiplexed group requires the number of racks that is equal to the number of VNFs in chain (line 8-12). VNFs from a service chain are placed sequentially on racks maintaining ordered sequence. Lines 13-24 of the algorithm ensure that at least one VNF from each service chain of a group is placed on each server rack and order of VNF is followed for each service chain in a group.

C. Routing and Wavelength Assignment

The RWA problem in the proposed DC is challenging due to constraints imposed by MD-WSS and modular nature of the DC. While solving the RWA problem, we need to ensure that wavelength assignment constraints imposed by MD-WSS are satisfied and for each multiplexed group of service chains, wavelengths are assigned to all the lightpaths on their end-to-end path. The wavelength assignment problem using vertex graph coloring is known to be an NP-complete problem [6]. We formulate an ILP and present heuristic to solve the wavelength assignment problem.

To solve the wavelength assignment problem using graph coloring method, a graph is constructed where all the requested lightpaths to be colored are represented by nodes in the graph. We transform the physical topology of the DCN

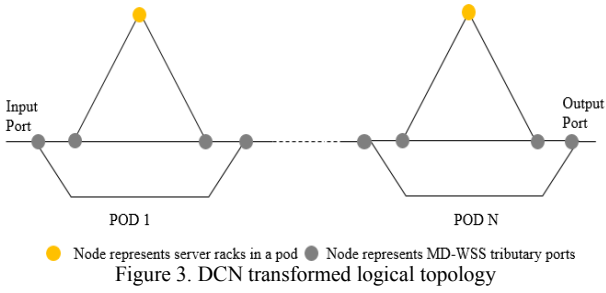


Figure 3. DCN transformed logical topology

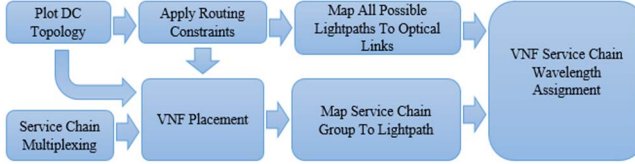


Figure 4. RWA flow diagram

into the simplified logical topology in Fig.3, which in turn allows us to apply graph coloring methods. In the transformation, all server racks in one pod are represented as a single node. MD-WSS tributary ports are represented as a node and tributary ports 1&2 of each MD-WSS are merged into one node. The edge between two nodes represents a link carrying traffic. Except the links used for intra-pod connections, all other links in the DC carry bidirectional traffic. As shown in Fig.4, we use the derived logical topology and apply routing constraints to identify the routes between all possible source-destination pairs within the DC and the links that are used on each route. Based on the VNF placement results, we dynamically compute all lightpath connections needed to setup an end-to-end path for each multiplexed group of VNF service chains by using the knowledge on mapping of multiplexed service chains group to server racks and server racks to the pods. By applying the routing constraints imposed by DCN architecture on the logical topology, we derive lightpaths between all possible end nodes in the DC. Then, we derive the mapping between lightpaths and links used by each lightpath. By combining all gathered information, lightpath connections needed to serve multiplexed service chains and links used by the lightpaths is derived which is further used for developing an ILP and a heuristic solution.

An ILP model is formulated for wavelength assignment with the objective of maximizing the number of VNF service chains served. Let V be the set of nodes in the graph representing lightpaths for each multiplexed VNF service chain request. Also, let L and E be the set of lightpaths, and the set of edges between two adjacent lightpaths, respectively. Let S be the set of multiplexed groups of VNF service chains and W be the set of available wavelengths. $X_{i,j,k}$ is a binary variable that is set to one if wavelength k is assigned to lightpath j used by a multiplexed group of VNF service chain i , where $i \in S, j \in L$ and $k \in W$. $D_{i,j}$ is a binary input variable that is set to one if a lightpath j is used by a multiplexed group of VNF service chain. Y_i is a binary variable that is set to one if a multiplexed group of chains is served. Z_k is a binary variable that is set to one if wavelength k is used.

$$\max \sum_i Y_i \quad (6)$$

$$\forall i \in S \& j \in L, k \in W \quad \sum_k \sum_j X_{i,j,k} = \sum_j D_{i,j} \cdot Y_i \quad (7)$$

$$\forall i \in S, \forall j \in L \& k \in W \quad \sum_k X_{i,j,k} = 1 \quad (8)$$

$$\forall i \in S, \forall j \in L, \forall k \in W \quad X_{i,j,k} \leq Z_k \quad (9)$$

$$\forall k \in W, \quad X_{i,j,k} + X_{m,n,k} \leq 1 \quad (10)$$

$$(i, j), (m, n) \in E$$

The objective in (6) is used to maximize the number of service chains served. Constraint (7) ensures that if wavelengths are assigned to all lightpaths in VNF service chain path, then the service chain is served. Constraint (8) ensures that each lightpath is assigned only one wavelength. Constraint (9) ensures that if lightpath is assigned wavelength k , then wavelength k is used. Finally, Constraint (10) ensures that if two lightpaths share the same edge then those lightpaths are not assigned with same wavelength.

We have also developed a heuristic for solving the wavelength assignment problem—though the details are not presented here due to space limit. Same as in ILP, the objective here is also to serve the maximum number of VNF service chains with available set of wavelengths. The graph consisting nodes representing lightpath requests by all multiplexed VNF service chains is used for wavelength assignment. To solve this, our heuristic first plots sub-graph for each multiplexed service chain request and sorts them in an ascending order. To do that we use the DSATUR (Degree of Saturation), which is a sequential graph algorithm for wavelength assignment [7]. Based on comparative study between several vertex graph coloring algorithms, DSATUR algorithm is efficient in terms of the number of colors used and time taken for coloring the graph [8]. Then, our heuristic assigns wavelengths sequentially to multiplexed groups of VNF service chains according to the sorted list. Wavelengths are assigned using parent graph such that two VNF service chains that use the same route are assigned with different wavelengths.

IV. EVALUATION RESULTS

For the performance evaluation, we consider the following metrics: the number of VNF service chains served, the number of server racks used to serve chains, the number of wavelengths needed to serve chains, system CPU utilization and link utilization. We have performed extensive simulation experiments in MATLAB to numerically investigate these metrics. Table 1 presents the basic parameter choice in our simulations. Considering the optical signal budget, we limit the number of server racks per pod to 16. Also, CPU requirement per VNF is varied depending upon data rate demand as shown in Table 2. We have designed and conducted sets of experiments as detailed below.

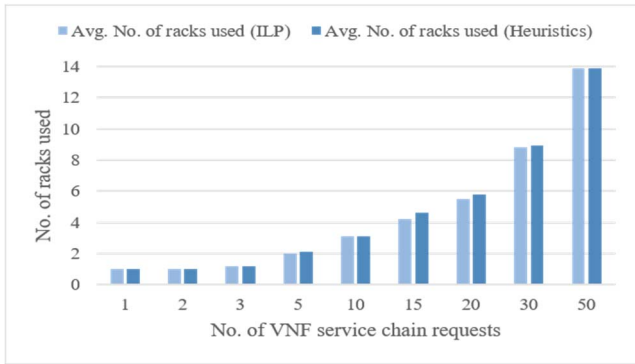


Figure 5. No. of racks used Vs No. of VNF service chain requests

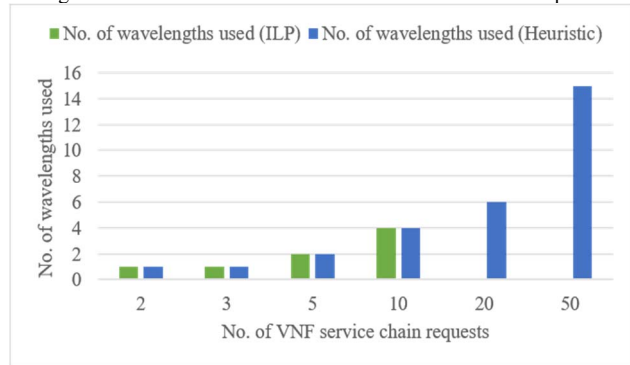


Figure 6. No. of wavelengths used Vs No. of VNF service chains

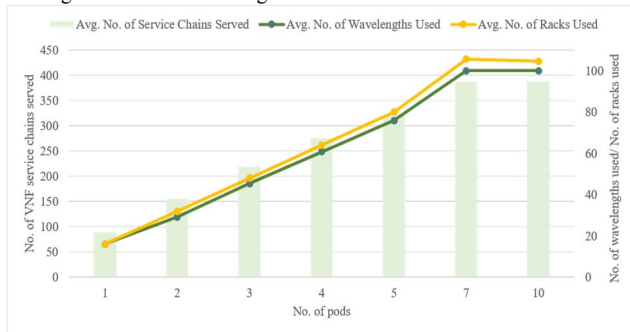


Figure 7. No. of chains served & DC resources usage Vs No. of pods

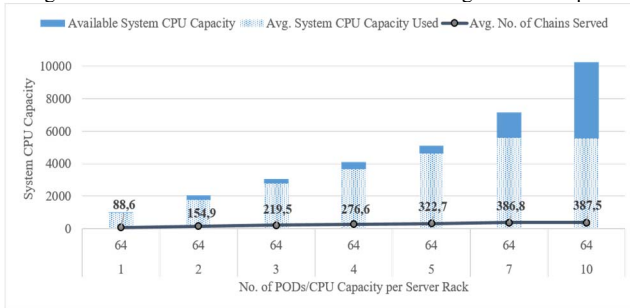


Figure 8. System CPU resource usage Vs no. of pods

TABLE 1

System Resource Parameters	Values
Link capacity	10Gbps
No. of wavelengths per fiber	100
CPU core capacity per server rack	64
Tunable Receiver per ToR switch	1

TABLE 2

Data Rate	≤ 0.5 Gbps	≤ 1.5 Gbps	≤ 3.5 Gbps	> 3 Gbps
CPU cores	1	2	4	8

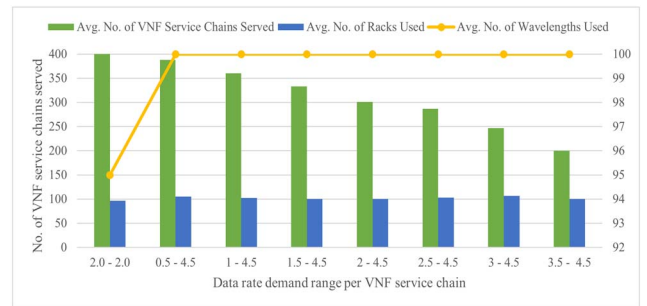


Figure 9. No. of chains server & DC resource used Vs. Data rate demand

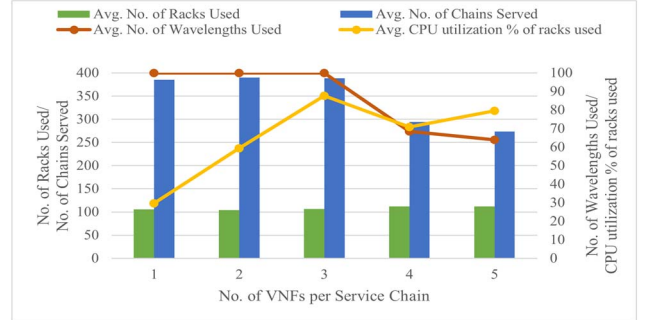


Figure 10. No. of VNF chains served and Resource usage Vs No. of VNFs per service chain

First, we investigate the quality of our heuristics—as compared to optimal solution of the ILP—by varying the number of service chain requests. Note that because of NP-hardness of both ILP problems, they do not scale for large number of VNF chain requests. Figs. 5 and 6 show results which demonstrate that the heuristics for multiplexing the VNF service chains and placement, as well as wavelength assignment all achieve near-optimum results as compared to the ILP solution. The number of VNFs per service chain are kept equal as ILP is developed with this assumption.

In the rest of our evaluations we consider only our heuristics, and experiments are carried out by randomly generating 400 VNF service chain requests. In second experiment, we evaluate the number of VNF service chains served by varying the number of pods (Fig. 7). We vary the number of pods from 1 to 10 and the number of VNFs per service chain is uniformly distributed in [1, 5] and data-rate demand of service chains is uniformly distributed in [0.5, 5] Gbps. Fig. 7 also depicts the number of required wavelengths and used server racks. We observe that with 7 pods in the DC, all 100 wavelengths are used and up to 387 (out of 400) service chains are served. We do not gain from increasing the pods beyond 7 since the network connectivity (the number of wavelengths) becomes the bottleneck. This effect is also reflected in Fig. 8, which shows that the system CPU resource utilization increases with the number of pods in the DC up until the case where we have 7 pods. After that adding extra CPU resources (i.e., more pods) only adds to the unused CPU capacity as all wavelengths are consumed to serve the traffic.

In the third experiment, we investigate the impact of the data-rate per VNF on the number of served chains (Fig. 9). 7 pods are used and the number of VNFs of all service chains is kept equal and data-rate demand of service chains is uniformly distributed in [0.5, 5] Gbps. As one can expect, increasing the

data-rate demand per service chain limits the possibility of multiplexing multiple chains, which in turn reduces the number of served VNF service chains.

In the last experiment, we evaluate the impact of the number of VNFs per service chain on the system performance (Fig. 10). 7 pods are used and the number of VNFs per service chain is uniformly distributed in (1, 5]. The average data-rate demand is varied in the range [2, 4.5] Gbps. We observe that as the number of VNFs per chain goes beyond 3, the total number of served chains decreases because the CPU resources in the DC are kept fixed.

V. RELATED WORKS

There has been significant research works on service chaining using SDN/NFV technologies, though most of them focus only on the packet switched networks. For example, Sherry *et al.* [9] study deployments of middle boxes in the cloud. Z. A. Qazi *et al.* [10] propose a centralized control mechanism to steer traffic to different services using SDN. Also, Y. Zhang *et al.* [11] investigate routing of traffic through any sequence of middle boxes using SDN/OpenFlow. The related VNF placement problem is also studied in the literature. For example, Mehraghdam *et al.* in [12] propose models for specifying and placing VNF service chain requests. Bari *et al.* [13] further propose a model to determine optimal number of VNFs and their placement to optimize operational cost. Moens *et al.* [14] propose VNF placement models for hybrid scenarios, where NFs may be provided as hardwares or virtualized instances.

Xia *et al.* [15] investigate a hybrid electrical/optical architecture DCN and the traffic steering in the optical domain for virtualized network services. The presented model in [15] is, however, limited to the VNF packing in existing DC pods to minimize the number of opto-electrical conversions.

VI. CONCLUSION

We presented a novel optical DCN architecture based on MD-WSS and addressed VNF service chaining in the proposed architecture. The service chaining is formulated as three interconnected sub-problems: VNF service chain multiplexing, VNF placement, as well as routing and wavelength assignments. We presented an ILP formulation as well as fast heuristic solutions to the sub-problems and demonstrated—through numerical examples—the effectiveness of the heuristics in mapping a large group of service chains to the optical and compute resources. For future work we will include on-line mapping of dynamic service chains to available resources.

ACKNOWLEDGEMENT

This work was supported by Celtic-Plus subproject SENDATE-EXTEND funded by Vinnova.

REFERENCES

[1] G. Porter *et al.*, “Integrating microsecond circuit switching into the data center,” in *Proc. of ACM SIGCOMM*, 2013.

[2] B. Han *et al.*, “Network function virtualization: challenges and opportunities for innovations,” *IEEE Communications Magazine*, vol. 53, no. 2, pp. 90–97, 2015.

[3] D. Kreutz *et al.*, “Software-defined networking: a comprehensive survey,” *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[4] A. Rostami *et al.*, “Orchestration of ran and transport networks for 5g: an sdn approach,” *IEEE Communications Magazine*, vol. 55, no. 4, pp. 64–70, 2017.

[5] A. Rostami and B. Skubic, “Multidirectional routing in wavelength selective switches,” in *Proc. of OFC 2015*, paper W3J.3.

[6] M. R. Garey and D. S. Johnson, *Computers and Intractability: a Guide to the Theory of NP-Completeness*, San Francisco, CA, USA: Freeman, 1979.

[7] D. Brelaz, “New methods to color the vertices of a graph,” *Communications of the ACM* 22, 1979.

[8] W. Klotz, “Graph coloring algorithms,” *Mathematik-Bericht*, vol. 5, pp. 1–9, 2002.

[9] Sherry *et al.*, “Making middle boxes someone else’s problem: network processing as a cloud service,” in *Proc. ACM SIGCOMM*, 2012.

[10] Z. A. Qazi *et al.*, “Simple-fying middlebox policy enforcement using SDN,” in *Proc. ACM SIGCOMM*, 2013.

[11] Y. Zhang *et al.*, “Steering: a software-defined networking for inline service chaining,” in *Proc. Int. Conf. Netw. Protocols*, 2013.

[12] S. Mehraghdam, M. Keller, and H. Karl, “Specifying and placing Chains of Virtual Network Functions,” arXiv preprint arXiv:1406.1058, 2014.

[13] M. Bari *et al.*, “On orchestrating virtual network functions in NFV,” arXiv:1503.06377v2 [cs.NI] 25 Mar 2015.

[14] H. Moens and F. De Turck, “VNF-P: a model for efficient placement of virtualized network functions,” *10th CNSM and Workshop*, 2014.

[15] M. Xia *et al.*, “Network function placement for NFV chaining in packet/optical datacenters”, *Journal of Lightwave Technology*, Vol. 33, No. 8, April 15, 2015.