On Establishing and Task Scheduling of Data-Oriented vNF-SCs in an Optical DCI

Zichen Xu and Zuqing Zhu, *Senior Member, IEEE*

***Abstract*—The development of network function virtualization (NFV) enables service providers to provision various network services with virtual network function service chains (vNF-SCs). However, most of the existing studies on the service provisioning of vNF-SCs only addressed the flow-oriented ones, while the provisioning schemes of the data-oriented vNF-SCs, each of which needs to process and transfer bulk data through a series of vNFs before a preset deadline, has not been fully explored yet. Therefore, this paper studies how to jointly optimize the establishing and task scheduling of data-oriented vNF-SCs in an optical datacenter interconnection (DCI), such that the probability of scheduling the data-oriented vNF-SCs to satisfy their deadlines can be maximized. To make the best use of the resources in the optical DCI for meeting each vNF-SC's deadline, we leverage dynamic programming (DP) to propose two time-efficient algorithms with the deadline-prioritized and conflict-aware approaches, respectively. Extensive simulations evaluate the performance of our proposed algorithms in different network scenarios, and confirm their effectiveness. Specifically, compared with a greedy-based benchmark, our DP-based algorithms can reduce the blocking probability of data-oriented vNF-SCs by up to two magnitudes and maintain similar service completion time.**

***Index Terms*—Datacenter interconnection (DCI), Network function virtualization (NFV), Service function chaining, Task scheduling, Elastic optical networks (EONs), Dependency graph.**

I. Introduction

Recently, the increase of network users and the emergence of various applications have put great pressure on network architectures and made network operations much more dynamic [1]. Therefore, network devices might need to be upgraded consistently, especially for those in datacenter interconnections (DCIs) [2]. This is because datacenters (DCs) normally provide the IT resources (*e.g.*, computing, memory and storage resources) needed by emerging network services [3]. Hence, the traditional way of deploying dedicated middleboxes to support network functions will not be suitable anymore. Network function virtualization (NFV) addresses this issue by replacing the middleboxes with virtual network functions (vNFs) [4]. Specifically, vNFs are instantiated on general-purpose network devices (*e.g.*, switches and servers), and by doing so, service providers (SPs) can decouple the software and hardware of network services to make sure that they can be launched/upgraded much more timely and cost-effectively. For instance, an SP can deploy the vNFs for firewall, deep packet inspection, load balancing, *etc.*, dynamically and adaptively on a same set of commodity switches and servers in a DCI. Then, by steering application traffic through the vNFs in specific orders, the SP can provision various network services in the form of vNF service chains (vNF-SCs) [5–7].

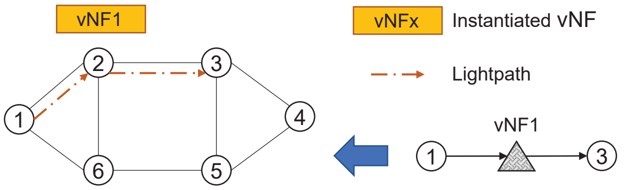
Z. Xu and Z. Zhu are with the School of Information Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, P. R. China (email: zqzhu@ieee.org).

Manuscript received on May 11, 2021.

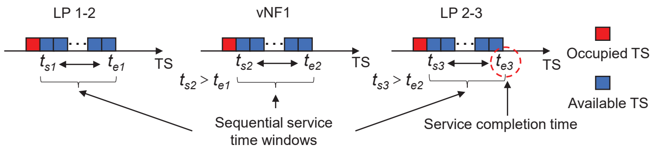
Meanwhile, to ensure the quality-of-service (QoS) of vNF-SCs, people need to rely on optical networking technologies to build DCIs, and the latest study in [3] even suggested to architect regional DCIs directly with optical circuit switching. Note that, with the new optical networking technologies such as flexible-grid elastic optical networking (EON) [8–11], an optical DCI can become more spectrum-efficient, adaptive and application-aware. Therefore, it is relevant to study the service provisioning of vNF-SCs in optical DCIs. Previously, a few studies have considered this problem and proposed various algorithms [6, 12–14]. However, they only addressed the flow-oriented vNF-SCs, which means that for each vNF-SC, the SP needs to first place the required vNFs in the DCs of an optical DCI, and then set up lightpaths to route continuous traffic through the vNFs in sequence. Hence, to provision the flow-oriented vNF-SCs, the existing studies generally optimized vNF placement together with routing and spectrum assignment (RSA) towards different objectives (*e.g.*, resource utilization, request blocking probability, and energy consumption).

Note that, other than the flow-oriented ones, there are also noticeable amounts of data-oriented vNF-SCs in an optical DCI, which process and transfer bulk data from time to time and usually need to satisfy the QoS demand on latency [15]. The data-oriented vNF-SCs are essential to support certain emerging applications such as grid computing in e-Science [16], large-scale and distributed machine learning [17], *etc.*, and their service provisioning is different from that of the flow-oriented ones. Specifically, as setup latency is normally critical for data-oriented vNF-SCs, we can only leverage the existing vNFs and lightpaths to provision them, *i.e.*, on-demand vNF instantiation and lightpath setup are not suitable. Moreover, as end-to-end latency is also important, we should pay attention to the task processing in existing vNFs. Therefore, to provision a data-oriented vNF-SC, we need to establish it with existing vNFs and lightpaths to process and transfer a specific volume of data in sequence, and schedule the task processing in each selected vNF, to ensure that the end-to-end latency of data processing and transferring can meet a preset deadline.

Here, we refer to the time period that a data-oriented vNF-SC is using a vNF/lightpath as its service time window (STW) on the vNF/lightpath, and define its service completion time (SCT) as the end-to-end latency of data processing and transferring for it. Fig. 1 provides an illustrative example on



(a) Selecting existing vNFs and lightpaths



(b) Scheduling of data transferring and processing

Fig. 1. Example on the service provisioning of a data-oriented vNF-SC.

the service provisioning of data-oriented vNF-SCs. We assume that there is a simple data-oriented vNF-SC request, which needs to send a specific volume of data from *Node* 1 to *vNF* 1 for being processed there and then target the data to *Node* 3. As shown in Fig. 1(a), the 6-node optical DCI has a *vNF* 1 running on the DC on *Node* 2, and two established lightpaths, *i.e.*, *LPs* 1-2 and 2-3. Hence, the data-oriented vNF-SC can leverage them for its service provisioning. Specifically, as shown in Fig. 1(b), the service provisioning actually involves reserving sequential STWs, *i.e.*, $[t_{s1}, t_{e1}]$, $[t_{s2}, t_{e1}]$, $[t_{s3}, t_{e3}]$ on *LP* 1-2, *vNF* 1 and *LP* 2-3, respectively.

We assume that the optical DCI is operated as a discrete-time system, *i.e.*, the operations in the time domain are performed according to time slots (TS'). Therefore, for the provisioning scheme in Fig. 1(b), the SCT of the data-oriented vNF-SC is the end time of the STW on *LP* 2-3, *i.e.*, t_{e3}, which should satisfy the preset deadline. To this end, we can see that to provision a data-oriented vNF-SC, we need to schedule its data processing and transferring in a dynamic network environment, which can be more complicated than that of a flow-oriented vNF-SC. More importantly, as the establishing of the data-oriented vNF-SC (*i.e.*, selecting the existing vNFs and lightpaths to set it up) and its task scheduling on the selected vNFs and lightpaths are correlated, we need to optimize them jointly, which makes the problem-solving more challenging.

In this work, we study how to jointly optimize the establishing and task scheduling of data-oriented vNF-SCs in an optical DCI. Each data-oriented vNF-SC request is assumed to have a service deadline, and when the operator finds that its SCT will exceed its deadline, the request will be blocked. Hence, our objective is to minimize the blocking probability of data-oriented vNF-SC requests, *i.e.*, to make the best utilization of the resources in the optical DCI while meeting each request's deadline. We leverage dynamic programming (DP) to propose two time-efficient algorithms for it, with deadline-prioritized and conflict-aware approaches, respectively. Extensive simulations evaluate the performance of our proposed algorithms in different network scenarios, and confirm that both of them can outperform the greedy-based benchmark algorithm.

The rest of the paper is organized as follows. Section II gives a brief survey on the related work. In Section III, we describe the problem and its network model in detail. The DP-based algorithms are proposed in Section IV, and we use numerical simulations to evaluate their performance in Section V. Finally, Section VI summarizes the paper.

II. Related Work

In recent years, NFV has attracted intensive attentions from both academia and industry, for the facts that it can effectively improve the cost-effectiveness of network services and reduce their time-to-market significantly. The specifications in [4] explain the service frameworks of NFV and the typical use-cases, while the standardization efforts in [18] specify the service provisioning and operation of vNF-SCs.

For flow-oriented vNF-SCs, many studies have already tackled the problem of their service provisioning in various networks [5–7, 12–14, 19–26]. For instance, the study in [19] tried to optimize the energy consumption of vNF-SCs in an IP-over-wavelength division multiplexing (WDM) network, while deep reinforcement learning (DRL) was leveraged in [22, 27] for vNF-SCs provisioning. Note that, although the provisioning of flow-oriented vNF-SCs looks similar to the well-known virtual network embedding (VNE) problem [28, 29], there are important differences because many-to-one mapping is normally not allowed in VNE [30].

As we have explained in the previous section, the provisioning of flow-oriented vNF-SCs is fundamentally different from that of data-oriented ones, and even though the studies in [23–26] also considered end-to-end latency in their objectives, they optimized the latency for flows but did not address the task scheduling in the vNFs. Compared with the service provisioning in packet-based networks (*e.g.*, in [5, 7, 21, 23–26]), the one in optical networks (*e.g.*, in [6, 12–14, 20]) needs to solve the RSA problem [31, 32], when setting up the lightpaths to connect vNFs. Note that, in an optical DCI, flow-oriented network services can also be composed in the forms of vNF multicast trees [30] and generic vNF forwarding graphs [33], in addition to vNF-SCs.

By assuming that vNFs can be deployed on substrate nodes (SNs) dynamically and on-demand, the investigations in [34–37] addressed the problem of scheduling vNF-SCs. Nevertheless, this problem is different from ours on the provisioning of data-oriented vNF-SCs in at least three aspects, as follows.

Firstly, the network models are different. For the scheduling of vNF-SCs considered in [34–37], the authors essentially tried to schedule the instantiations of vNFs to satisfy various delay requirements. Specifically, as different types of vNFs can be instantiated on an SN dynamically, they solved the problem of how to deploy the required vNFs spatially (*i.e.*, on SNs) and temporally (*i.e.*, in TS'). On the other hand, our problem assumes that the vNFs have already been instantiated in an optical DCI, and thus we need to schedule the data processing and transferring of data-oriented vNF-SCs on existing vNFs and lightpaths. Therefore, we do not schedule the vNFs but actually schedule the data processing tasks in vNFs. This means that our problem is not only based on a different network model, but also for an optimization with more constraints.

Secondly, the importance of flow routing is different in the two problems. As the network models in [34–37] assumed that different types of vNFs can be instantiated on an SN dynamically, the routing of the application traffic of a vNF-SC for going through the required vNFs in sequence becomes less important or even trivial to the quality of a solution. Hence, the studies in [34, 37] did not even consider the routing problem. However, the routing of data transfers is much more important in our problem. This is because it determines where the bulk data will be processed for its next required vNF, and if the routing path is not properly selected, all the STWs after that of the current vNF will be delayed. Therefore, in our problem, the routing of data transfers on lightpaths has to be optimized jointly with the scheduling of data processing in vNFs, which makes the optimization more challenging. This is because the routing schemes need to be planned on account of one more dimension (*i.e.*, the time schedule of data processing in vNFs).

Lastly, but not least, as latency is critical to the provisioning of data-oriented vNF-SCs, we cannot solve our problem with a mixed integer linear programming (MILP) model as in [34–37], because they can hardly be solved time-efficiently to obtain timely decisions for online provisioning. For the same reason, we cannot leverage the approaches based on tabu search [34], genetic algorithm [35], and column generation [36] either. To the best of our knowledge, this work is the first one that tries to jointly optimize the establishing and task scheduling of data-oriented vNF-SCs in an optical DCI.

III. PROBLEM DESCRIPTION

In this section, we first elaborate on the network model used in this work, and then explain the problem of provisioning data-oriented vNF-SCs in detail.

A. Network Model

We model the optical DCI as an undirected graph $G(V, E)$, where V is the set of substrate nodes (SNs) and E is the set of fiber links to interconnect the SNs. We assume that there are two types of SNs in the optical DCI, *i.e.*, the DC nodes and optical switching nodes (OSNs) [3]. Each DC node consists of both a local DC and an OSN, which is equipped with a bandwidth-variable optical switch (BV-OXC) and several sliceable bandwidth-variable transponders (SBVTs), *i.e.*, EON is assumed to be used in the optical layer [8]. On the other hand, an OSN is only for optical switching and does not have a local DC. On the fiber links in E, a number of lightpaths have already been established for inter-DC communications.

In the DCs, F types of vNFs can be instantiated. We use $\{f_{v,1}, \cdots, f_{v,n}\}$ to represent the existing vNFs that have been deployed on a DC node $v \in V$. For a type-f vNF ($f \in F$), the volume of the data that it can process within a TS is α_f, and it has an output-to-input data volume ratio of β_f, *i.e.*, after receiving and processing one unit of data, the volume of the data that it generates is β_f. A time table is maintained for each existing vNF on a DC, which classifies future TS' on the vNF into occupied and spare ones, and only the spare TS' can be allocated to process the data of a data-intensive vNF-SC. Meanwhile, as a lightpath in an optical DCI usually has a relatively large capacity (*e.g.*, 100 Gbps or beyond [3]), transferring a specific volume of data on it normally takes much shorter time than the latency of processing the same data in a vNF. Hence, we assume that transferring the data of a data-oriented vNF-SC on an existing lightpath takes one TS, regardless of the volume of the data.

We use a five-tuple $R = \{s, d, b, SC, \tau\}$ to denote a data-oriented vNF-SC request, where s and d are the source and destination SNs, b is the initial volume of data from the source, $SC = \{f_1, \cdots, f_n\}$ is the set of required vNFs in sequence, and τ is the deadline on SCT. To provision such a data-oriented vNF-SC request R, we need to select existing vNFs and lightpaths in the optical DCI to satisfy its demand, and schedule the data processing and transferring on the selected vNFs and lightpaths, respectively, to ensure that its SCT satisfies the deadline τ. Specifically, for each $f_i \in SC$, we need to schedule a STW on it to process the data of R, and the scheduled STWs have to be sequential in the order of their vNFs in SC. Note that, as the data processing capacity of each vNF is fixed, the length of the STW on it for R increases with b. Meanwhile, there should be one or more lightpaths between each pair of adjacent vNFs (*e.g.*, $s \rightarrow f_1$, $f_i \rightarrow f_{i+1}$, and $f_n \rightarrow d$) to transfer the data of R, and the data transfer on each lightpaths introduces a delay of one TS.

B. Service Provisioning of Data-Oriented vNF-SCs

Note that, as a vNF might not always be free for incoming data, there can be a gap between the arrival time and STW of the data of R, *i.e.*, the data needs to be buffered for a while before it can be processed by the vNF. This introduces a "data-to-be-processed" delay. Therefore, the end-to-end latency of a data-oriented vNF-SC request R is actually the summation of the data processing and data-to-be-processed delays on all of its vNFs and the data transferring delays on all of its lightpaths.

Fig. 2 shows three provisioning schemes for a same data-oriented vNF-SC in an optical DCI, each of which is indicated by a red arrow line there. Here, we assume that the vNF-SC is *A*→*vNF* 1→*vNF* 2→*B*, where *Sites A* and *B* are locally attached to *DCs* 1 and 3, respectively, arriving at *TS* 1, and the latencies for processing the data of the vNF-SC are 1 and 2 TS' on *vNFs* 1 and 2, respectively. As illustrated in Fig. 2(a), there are a few vNFs running in the three DCs and three established lightpaths on the fiber links (*i.e.*, *LPs* 1-3), when the data-oriented vNF-SC request comes in.

- The first provisioning scheme in Fig. 2(a) selects the *vNFs* 1 and 2 on *DC* 1 and sends the processed data to *Site* B (on *DC* 3) with *LP* 3. Hence, the STW on *vNF* 1 is *TS* 1, and since the *vNF* 2 on *DC* 1 will not be available until *TS* 3, there will be a data-to-be-processed delay of 1 TS. Then, the STW on *vNF* 2 is *TS'* 3-4, and the data transferring on *LP* 3 takes 1 TS. Finally, the SCT of the first provisioning scheme in Fig. 2(a) is *TS* 6.
- The second provisioning scheme in Fig. 2(b) selects the *vNFs* 1 and 2 on *DCs* 1 and 2, respectively, and sends the processed data with *LPs* 1 and 2. Hence, the STW on *vNF* 1 is still *TS* 1, and then the data transferring on *LP* 1 uses *TS* 2. When the data arrives *DC* 2 at *TS* 3, the *vNF* 2 there is not available, and thus the STW on *vNF* 2 is *TS'* 4-5. Finally, the data transferring on *LP* 2 takes 1 *TS*, which makes the SCT of the scheme as *TS* 7.

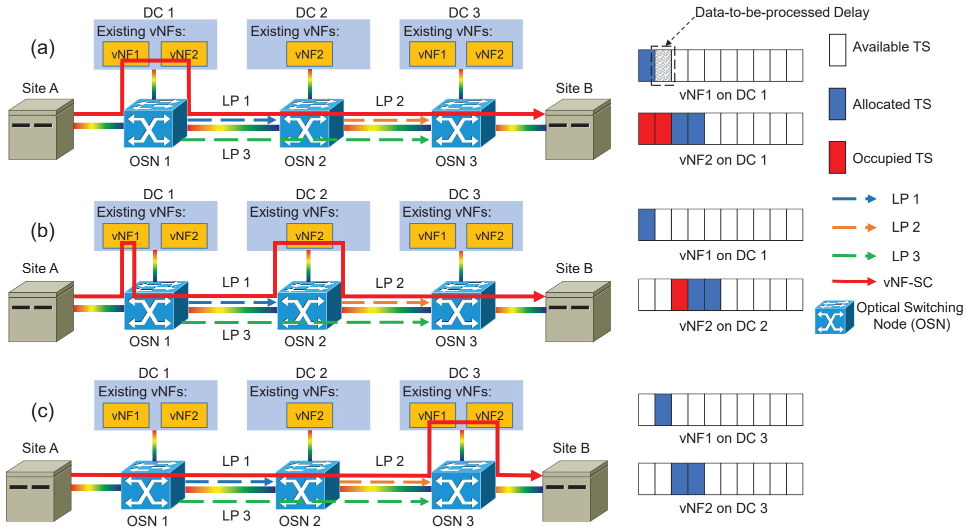


Fig. 2. Examples on the establishing and task scheduling of a data-oriented vNF-SC.

- The third provisioning scheme in Fig. 2(c) selects the *vNFs* 1 and 2 on *DC* 3 and sends the data to *DC* 3 with *LP* 3. The data transferring on *LP* 3 first takes 1 *TS*, and then the STWs on *vNFs* 1 and 2 are sequentially scheduled as *TS* 2 and *TS'* 3-4, respectively. Hence, the SCT of the provisioning scheme is *TS* 5.

Therefore, we can see that the third provisioning scheme in Fig. 2(c) provides the earliest SCT by selecting the proper vNFs to avoid unnecessary data-to-be-processed delay and using the right lightpaths to minimize the latency of data transferring. Meanwhile, the examples in Fig. 2 also illustratively explains the major differences between our problem and that of scheduling vNF-SCs [34–37], *i.e.*, the optimization of our problem has more constraints regarding the existing vNFs and lightpaths, and the routing of data transfers on lightpaths is important to the SCT of each data-oriented vNF-SC.

For a given network state, the best provisioning scheme of a data-oriented vNF-SC request R, which represents the way to get the earliest SCT for R, can be obtained by leveraging the dynamic programming (DP) in consideration of the time tables of the existing vNFs. This can be done in polynomial time. However, when it comes to serve multiple data-oriented vNF-SC requests simultaneously, the provisioning schemes of the requests might become correlated because they need to share the existing vNFs in a time-division multiplexing (TDM) manner. If we would like to optimize the provisioning schemes of the data-oriented vNF-SC requests jointly, it can be proven that the optimization can be reduced to the multi-machine scheduling problem, which is known to be $\mathcal{NP}$-hard [38].

Hence, we do not try to solve the combinational optimization exactly. Instead, for a set of pending data-oriented vNF-SCs, we provision them as follows. For each data-oriented vNF-SC, we get its provisioning scheme with DP, while the conflicts between the provisioning scheme and those of other data-oriented vNF-SCs are resolved with an algorithm. Note that, although certain conflicts can be resolved, some others cannot, which eventually lead to request blockings.

IV. Algorithm Design

In this section, we design algorithms to provision data-oriented vNF-SCs such that the blocking probability can be minimized. Specifically, we first explain how to leverage DP to find the best provisioning scheme for each pending data-oriented vNF-SC, and then design two algorithms, namely, the deadline-prioritized and conflict-aware algorithms, to resolve the conflicts among the best provisioning schemes of all the pending data-oriented vNF-SCs, as many as possible.

A. Finding the Best Provisioning Scheme with DP

For a given state of the optical DCI, the best provisioning scheme of a data-oriented vNF-SC $R = \{s, d, b, SC, \tau\}$, which provides the earliest SCT, can be obtained with DP. To achieve this, we first introduce the following variables.

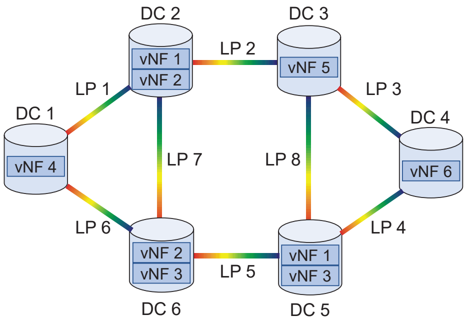
- $t_p(s, f_k, v)$: the finalized earliest time that a vNF $f_k \in SC$, which is running on DC v, can finish the data processing for R.
- $t_c(v, u)$: the completion time of data transferring on existing lightpaths from node v to node u ($u, v \in V$).
- $t_p(f_k, v)$: the earliest time that a vNF $f_k \in SC$, which is running on DC v, can finish the data processing for R.
- $\tau_{\min}$: the earliest SCT for R.

Then, we can obtain the recursive relations as

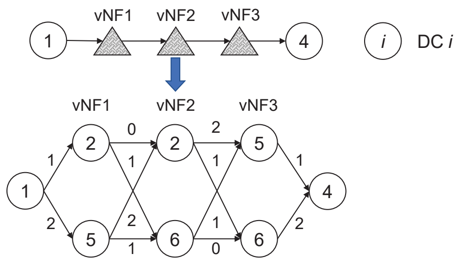
$$t_p(s, f_1, v) = [t_p(f_1, v)|t_c(s, v)], \quad \{v : f_1 \downarrow v, \ v \in V\}, \tag{1}$$

which means that for the data-oriented vNF-SC R, the earliest time that the vNF $f_1 \in SC$, which is running on DC v, can finish the data processing for R is just the earliest time that the vNF f_1 on DC v can accomplish the date processing for R, provided that the data of R arrives at DC v at time $t_c(s, v)$. Here, we define two operations. Firstly, $(t_1|t_2)$ means the earliest time of t_1 after the time t_2, and this operation can be concatenated, *e.g.*, $(t_1|t_2|t_3)$ means the earliest t_1 after t_2, where t_2 is the earliest after t_3. Secondly, $f \downarrow v$ means that a vNF f is currently running on DC v.

$$\begin{aligned} t_p(s, f_i, u) = [t_p(f_i, u)|t_c(v, u)|t_p(s, f_{i-1}, v)], \\ \forall i \in [2, n], \ \{u : f_i \downarrow u, \ u \in V\}, \end{aligned} \tag{2}$$



(a) Optical DCI



(b) Decision graph of a data-oriented vNF-SC

Fig. 3. Example on finding the best provisioning scheme with DP.

$$\tau_{\min} = \min_{\{v: f_n \downarrow v,\ v \in V\}} [t_c(v,d) | t_p(s, f_n, v)]. \tag{3}$$

With the recursive relations in Eqs. (1)-(3), we can find the best provisioning scheme for R and its earliest SCT $\tau_{\min}$ with DP. Specifically, the operation of the DP can be better understood if we build a decision graph for R based on the state of the optical DCI, as explained in the example in Fig. 3. The optical DCI in Fig. 3(a) consists of 6 DC nodes, and there are 8 established lightpaths in it. We assume that the vNF-SC is *DC* 1→*vNF* 1→*vNF* 2→*vNF* 3→*DC* 4, arriving at *TS* 1, and the latencies for processing the data of the vNF-SC are all 1 TS on *vNFs* 1-3, respectively. Based on the state of the optical DCI in Fig. 3(a), we can obtain the decision graph in Fig. 3(b) for the vNF-SC, where each column corresponds to the feasible locations of a vNF, and the number aside each link denotes the corresponding data transfer latency between its two end nodes. For instance, the second column is for *vNF* 1, and as Fig. 3(a) shows that *vNF* 1 runs on *DCs* 2 and 5, we have the two DCs in that column. The number aside the link from the source (*DC* 1) to *DC* 5 is 2, which is because the data transferring from *DC* 1 to *DC* 5 needs to uses two lightpaths (*i.e.*, *LPs* 6 and 5 in sequence).

With the decision graph in Fig. 3(b), we first calculate $t_p(\textit{DC}\ 1, \textit{vNF}\ 1, \textit{DC}\ 2)$ and $t_p(\textit{DC}\ 1, \textit{vNF}\ 1, \textit{DC}\ 5)$ with Eq. (1). Then, we proceed to the third and fourth columns in the decision graph to get $t_p(\textit{DC}\ 1, \textit{vNF}\ 3, \textit{DC}\ 5)$ and $t_p(\textit{DC}\ 1, \textit{vNF}\ 3, \textit{DC}\ 6)$ with Eq. (2). Finally, the earliest SCT $\tau_{\min}$ can be obtained with Eq. (3) as

$$\begin{aligned}\tau_{\min} = \min\{&[t_c(\textit{DC}\ 5, \textit{DC}\ 4) | t_p(\textit{DC}\ 1, \textit{vNF}\ 3, \textit{DC}\ 5)], \\ &[t_c(\textit{DC}\ 6, \textit{DC}\ 4) | t_p(\textit{DC}\ 1, \textit{vNF}\ 3, \textit{DC}\ 6)]\}.\end{aligned} \tag{4}$$

B. Deadline-Prioritized Algorithm (DLP-DP)

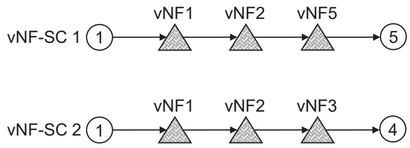
With the aforementioned DP-based approach, we can find the best provisioning scheme of each data-oriented vNF-SC under a specific state of the optical DCI. Hence, a straightforward idea of serving a set of data-oriented vNF-SCs (*i.e.*, the set is denoted as $\mathbf{R}$) is to first sort the vNF-SCs in $\mathbf{R}$ in ascending order of their deadlines on SCT and then provision the vNF-SCs in sorted order with the DP-based approach. *Algorithm* 1 shows the procedure of the deadline-prioritized approach, namely, DLP-DP. *Line* 1 is the initialization to prioritize the data-oriented vNF-SCs in $\mathbf{R}$ according to their deadlines on SCT. Then, the for-loop that covers *Lines* 2-10 tries to leverage the DP-based approach to provision all the vNF-SCs in the sorted order. Note that, if the best provisioning scheme obtained with DP still cannot satisfy the deadline of a data-oriented vNF-SC, the vNF-SC will be blocked (*Lines* 7-8). The sorting in *Line* 1 can be finished with a complexity of $O(|\mathbf{R}| \cdot \log(|\mathbf{R}|))$, where $|\cdot|$ returns the number of elements in a set. The complexity of *Lines* 2-10 is $O(M \cdot |V|)$, where M is the total number of vNFs in all the vNF-SCs in $\mathbf{R}$. Finally, we can get the complexity of *Algorithm* 1 as $O(|\mathbf{R}| \cdot \log(|\mathbf{R}|) + M \cdot |V|)$.

Algorithm 1: Deadline-Prioritized Algorithm (DLP-DP)

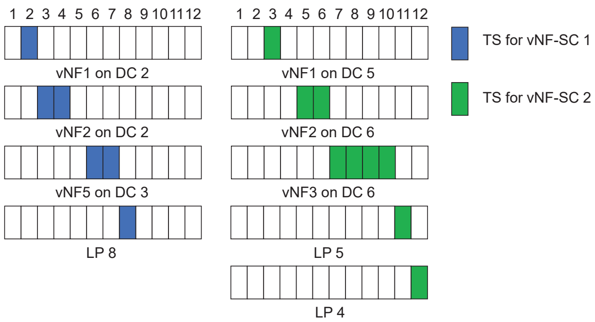
Input: Set of data-oriented vNF-SCs $\mathbf{R}$, $G(V,E)$
1 sort the data-oriented vNF-SCs in $\mathbf{R}$ in ascending order of their deadlines on SCT;
2 **for** *each vNF-SC* $R \in \mathbf{R}$ *in sorted order* **do**
3   find the best provisioning scheme for R based on the current state of optical DCI with DP;
4   **if** *the obtained SCT satisfies the preset deadline* **then**
5   provision R according to the best scheme;
6   update the state of optical DCI;
7   **else**
8   mark R as blocked;
9   **end**
10 **end**

One issue with the DLP-DP in *Algorithm* 1 is that each data-oriented vNF-SC is provisioned without any consideration on the others. For instance, although a vNF-SC whose deadline is earlier should generally be considered earlier, this does not necessarily mean that the vNF-SC should be provisioned with the best provisioning scheme. In other words, it is fine as long as the deadline of the vNF-SC can be satisfied, and thus we should be able to delay its provisioning until the deadline to avoid blocking other data-oriented vNF-SCs.

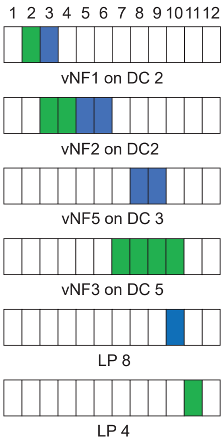
More specifically, Fig. 4 provides an example on the negative case of DLP-DP. Here, we try to provision the two pending data-oriented vNF-SCs in Fig. 4(a) still in the optical DCI in Fig. 3(a), and their deadlines on SCT are *TS* 11 and *TS* 12, respectively. To process the data of *vNF-SC* 1, *vNFs* 1, 2 and 5 use 1, 2 and 2 TS', respectively, and for *vNF-SC* 2, the data processing on *vNFs* 1, 2 and 3 takes 1, 2 and 4 TS', respectively. Hence, if we provision the two vNF-SCs with DLP-DP, the results are shown in Fig. 4(b), which indicates that the SCTs of *vNF-SCs* 1 and 2 are *TS* 9 and *TS* 13,



(a) Data-oriented vNF-SC requests



(b) Provisioning schemes obtained with DLP-DP



(c) Better provisioning schemes

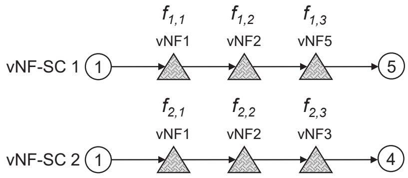
Fig. 4. Example on the negative case of DLP-DP algorithm.

respectively. This means that *vNF-SC* 2 will be blocked. On the other hand, if we use the provisioning schemes in Fig. 4(c), *i.e.*, *vNF-SC* 1 yields its provisioning on the *vNFs* 1 and 2 on *DC* 2 to that of *vNF-SC* 2, the SCTs will be *TS* 11 and *TS* 12, respectively. This means that none of the vNF-SCs will be blocked. Therefore, DLP-DP can still be improved if we consider the conflicts among pending data-oriented vNF-SCs.

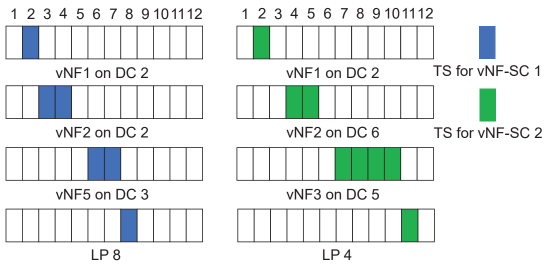
C. Conflict-Aware Algorithm (CA-DP)

The conflict-aware approach, namely, CA-DP, is designed based on the fact that certain data-oriented vNF-SCs do not need to be provisioned with the best schemes and can tolerate certain data-to-be-processed delay, and it works as follows. For each data-oriented vNF-SC, we first get its best provisioning scheme with DP, assuming that all the other vNF-SCs are not provisioned. Then, we check all the best provisioning schemes to find the conflicts among them, and resolve the conflicts by rearranging the provisioning schemes.

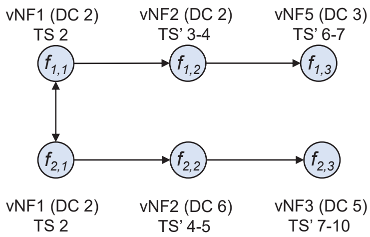
With all the best provisioning schemes, we build a dependency graph (DG) $G_d(V_d, E_d)$, where each node $v_d \in V_d$ represents a vNF in the pending data-oriented vNF-SCs, and



(a) Data-oriented vNF-SC requests



(b) Best provisioning schemes from DP



(c) Dependency graph (DG)

Fig. 5. Example on building a DG based on the best provisioning schemes.

each link $e_d \in E_d$ denotes the dependency between two vNFs. Specifically, there are two types of dependencies among the vNFs. Firstly, for the m-th vNF-SC in $\mathbf{R}$, there is a directed link from the node for $f_{m,i}$ (*i.e.*, the i-th vNF in the vNF-SC) to that for $f_{m,i+1}$, to represent the dependency between two adjacent vNFs in a vNF-SC. Secondly, for any two vNFs in different vNF-SCs, if the scheduled STWs in their best provisioning schemes have a conflict, we insert a bidirectional link between the nodes for them.

Fig. 5 gives an example on how to build the DG for data-oriented vNF-SCs. Here, we still need to provision the two data-oriented vNF-SCs in Fig. 5(a) in the optical DCI in Fig. 3(a), and their deadlines on SCT are still *TS* 11 and *TS* 12, respectively. Then, with DP, we can get the best provisioning schemes of the two vNF-SCs as in Fig. 5(b), which indicates that the STW for the *vNF* 1 in *vNF-SC* 1 conflicts with that for the *vNF* 1 in *vNF-SC* 2. Next, the DG can be built as shown in Fig. 5(c), based on the best provisioning schemes in Fig. 5(b), and the bidirectional link between the nodes for $f_{1,1}$ and $f_{2,1}$ denotes the conflict on the *vNF* 1 on *DC* 2.

With the DG, we resolve the conflicts by rearranging the provisioning schemes such that all the bidirectional links can be removed. *Algorithm* 2 shows the detailed procedure of CA-DP. *Lines* 1-5 are for the initialization, where we use F_m^p and F^p to store the pending vNFs in the m-th vNF-SC R_m and all the pending vNFs, respectively (*Line* 5). Then, the while-loop that covers *Lines* 6-32 tries to finalize the scheduling of each vNF in F^p, until F^p becomes empty. As we need to resolve the conflicts among the best provisioning schemes of the data-oriented vNF-SCs in $\mathbf{R}$, we still first sort them in ascending order of their deadlines on SCT in *Line* 7, similar to DLP-DP.

Algorithm 2: Conflict-aware Algorithm (CA-DP)

Input: Set of data-oriented vNF-SCs $\mathbf{R}$, $G(V, E)$
1 **for** *each vNF-SC* $R_m \in \mathbf{R}$ **do**
2   find the best provisioning scheme for R_m based on the current state of optical DCI with DP;
3 **end**
4 build a DG $G_d(V_d, E_d)$ based on the best schemes;
5 $F_m^p = \{f_{m,i}, \forall f_{m,i} \in SC_m\}$, $F^p = \bigcup_{R_m \in \mathbf{R}} F_m^p$, $f = \varnothing$;
6 **while** $F^p \neq \emptyset$ **do**
7   sort the data-oriented vNF-SCs in $\mathbf{R}$ in ascending order of their deadlines on SCT;
8   **for** *each* R_m *with* $F_m^p \neq \emptyset$ *in sorted order* **do**
9   select the first pending vNF $f_{m,i}$ in F_m^p;
10   **if** $f_{m,i}$ *does not conflict with others in* G_d **then**
11             $f = f_{m,i}$;
12   **else**
13             $F' = \emptyset$;
14   put $f_{m,i}$ and all the vNFs in V_d, which have conflicts with $f_{m,i}$ in set F';
15   **for** *each vNF* $f' \in F'$ **do**
16   reschedule f' to a later STW to resolve all the conflicts with other vNFs in F';
17   get penalty of rearranging f' with Eq. (5);
18   **end**
19   select $f^* = f_{n,j}$ whose penalty is the largest;
20   **if** *vNF* $f_{n,j}$ *is the first one in* F_n^p **then**
21                 $f = f_{n,j}$;
22   **end**
23   **end**
24   **if** $f \neq \varnothing$ **then**
25   finalize the scheduling of vNF f as planned; remove f from F^p and the related F_m^p or F_n^p;
26   **for** *each vNF-SC* R_m *with* $F_m^p \neq \emptyset$ **do**
27   find the best provisioning scheme for R_m based on the current state of optical DCI with DP;
28   **end**
29   update the DG G_d with the best schemes;
30   **end**
31   **end**
32 **end**
33 block the vNF-SCs whose SCTs exceed their deadlines;

Next, the for-loop covering *Lines* 8-31 checks the vNF-SCs that still have pending vNF(s) in the sorted order, and tries to finalize the scheduling of a vNF in each iteration. *Line* 9 selects the first pending vNF on a vNF-SC. If it does not have any conflict with others in GD $G_d(V_d, E_d)$, we just select it to finalize the scheduling (*Lines* 10-11). Otherwise, we put the vNF and all the vNFs that have conflicts with it in set F' (*Lines* 12-14), and try to reschedule each vNF in F' to a later STW such that all the conflicts with other vNFs in F' can be resolved (*Lines* 15-18). Meanwhile, we also calculate the penalty of the rescheduling in *Line* 17 as

$$\xi_m = \begin{cases} \dfrac{1}{\tau_m - \tau'_m + 1}, & \tau'_m \leq \tau_m, \\ +\infty, & \text{otherwise,} \end{cases} \tag{5}$$

where ξ_m is the penalty of rescheduling the vNF in vNF-SC R_m, τ'_m is the new SCT of R_m after the rescheduling, and τ_m is the deadline on SCT of R_m. Hence, the penalty of rescheduling ξ_m decreases with the gap between the new SCT and the preset deadline. Then, *Line* 19 selects the vNF that is in F' and has the largest penalty of rescheduling. We also check whether the selected vNF is the first pending one in its vNF-SC, and only choose it to finalize the scheduling if it is (*Lines* 20-22). This is because the scheduling of the vNFs in each vNF-SC has to be determined in sequence.

The aforementioned procedure can be better understood with the example in Fig. 5. Specifically, with the DG in Fig. 5(c), we first select $f_{1,1}$ to resolve its conflict with $f_{2,1}$, and thus put $f_{1,1}$ and $f_{2,1}$ in F'. Then, if we reschedule $f_{1,1}$ to resolve the conflict, its new SCT will be $\tau'_1 = 11$, which still satisfies its deadline ($\tau_1 = 11$). However, if we reschedule $f_{2,1}$, its new SCT will be $\tau'_2 = 13$, which exceeds its deadline ($\tau_2 = 12$), and this makes the penalty of rescheduling $f_{2,1}$ as $\xi_2 = +\infty$ according to Eq. (5). Therefore, *Line* 21 selects $f_{2,1}$ to finalize its scheduling, assuming that $f_{1,1}$ will be rescheduled in a subsequent iteration.

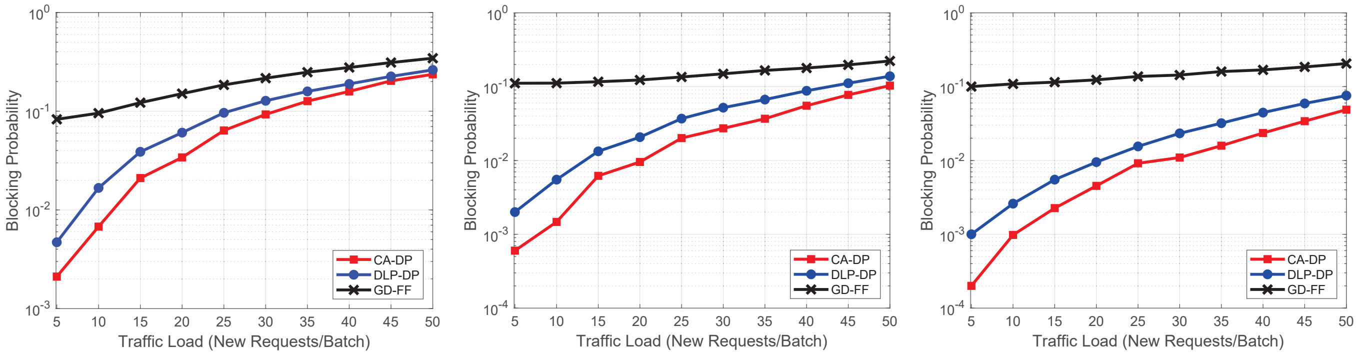
After this, *Line* 24 checks whether a vNF is found for finalizing its scheduling. If yes, *Lines* 25-29 finalize the scheduling of the vNF, update the related parameters and the state of the optical DCI accordingly, and recalculate the best provisioning schemes and the DG for pending data-oriented vNF-SCs based on the new state of the optical DCI. Finally, after the provisioning schemes of all the vNF-SCs have been determined, *Line* 33 marks the vNF-SCs whose SCTs cannot satisfy their deadlines as blocked. The time complexity of *Algorithm* 2 is $O(M^3 + M^2 \cdot |V| + M \cdot |\mathbf{R}| \cdot \log(|\mathbf{R}|))$, where M is still the total number of vNFs in all the vNF-SCs in $\mathbf{R}$. Hence, compared with DLP-DP, CA-DP sacrifices certain time complexity in exchange for better blocking performance.

V. Performance Evaluation

In this section, we conduct extensive simulations to evaluate the performance of our proposed algorithms.

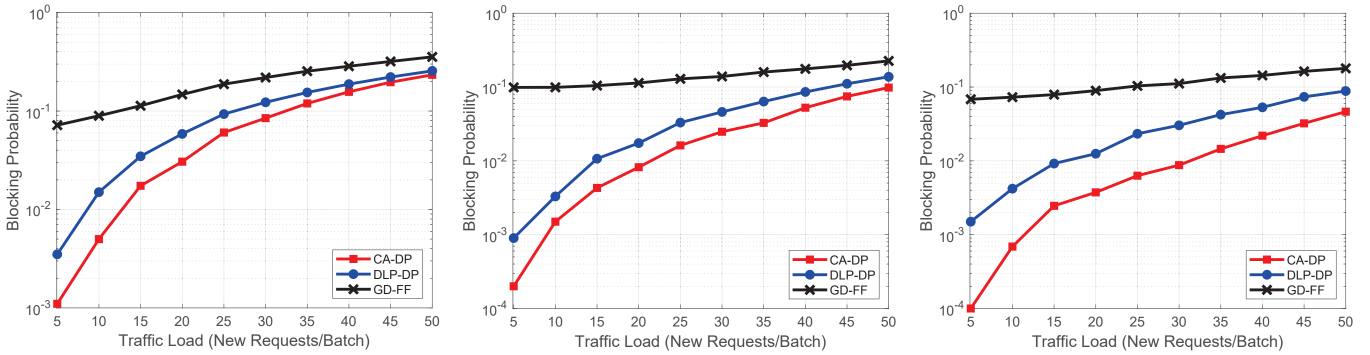
A. Simulation Setup

The simulations use the 24-node US backbone (USB) network shown in Fig. 6 as the topology of the optical DCI. We consider the cases that there are $\{8, 12, 16\}$ DC nodes in the optical DCI, and their locations are randomly selected. For each pair of nodes in the optical DCI, the probability that they are directly connected by a lightpath is set as 0.2 or 0.4. We assume that the optical DCI supports 10 types of vNFs, the types of the vNFs running on each DC are randomly selected within $[2, 4]$, and the data processing speed of each type of vNFs is randomly chosen within $[0.5, 6]$ units/TS. For each data-oriented vNF-SC $R = \{s, d, b, SC, \tau\}$, its source



(a) 8 DC nodes (b) 12 DC nodes (c) 16 DC nodes

Fig. 7. Results on blocking probability (each pair of nodes are directly connected by a lightpath with a probability of 0.2).



(a) 8 DC nodes (b) 12 DC nodes (c) 16 DC nodes

Fig. 8. Results on blocking probability (each pair of nodes are directly connected by a lightpath with a probability of 0.4).

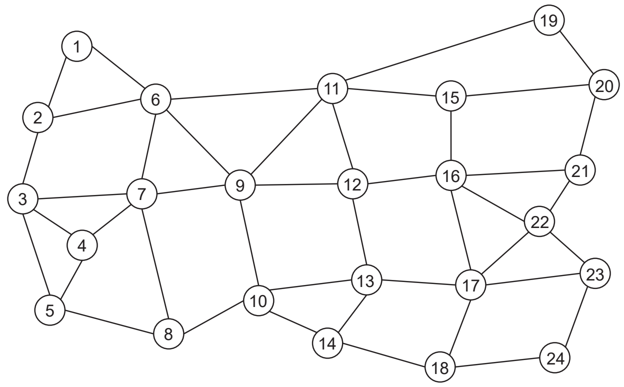


Fig. 6. Network topology of optical DCI.

and destinations are randomly chosen from the 24 nodes in the topology, the number of vNFs in SC is within $[1, 3]$, the initial data volume b is uniformly distributed within $[1, 4]$ units, and the output-to-input data volume ratio of each vNF is set within $[0.7, 1.3]$. To make sure that the deadline τ is set with a reasonable value, we first obtain the best provisioning scheme of R with DP, assuming that all the other pending vNF-SCs are not provisioned, and record the obtained SCT as the lower-bound on the deadline ($\tau_{\min}$). Then, we set the deadline as $\tau \in [\tau_{\min} + \delta_1, \tau_{\min} + \delta_2]$, where we have $\delta_2 > \delta_1 > 0$ and will test different values for them in the simulations. Table I summarizes the simulation parameters.

TABLE I
SIMULATION PARAMETERS

Parameter	Value Range
Number of DC nodes	$\{8, 12, 16\}$
Lightpath connectivity	$\{0.2, 0.4\}$
Types of vNFs on each DC	$[2, 4]$
Data processing speed of a vNF	$[0.5, 6]$
Number of vNFs in each vNF-SC	$[1, 3]$
Initial data volume of each vNF-SC	$[1, 4]$
Output-to-input data volume ratio of each vNF-SC	$[0.7, 1.3]$
δ_1	1
δ_2	$\{5, 10\}$

The simulations consider dynamic scenarios in the optical DCI, *i.e.*, data-oriented vNF-SCs arrive in batches according to the Poisson traffic model. Specifically, the number of vNF-SCs in each batch follows the Poisson distribution, while the time interval in between two adjacent batches conforms to the negative exponential distribution with a mean value of 25 TS'. Hence, the traffic load can be defined as the average number of new vNF-SC requests in each batch. In addition to DLP-DP and CA-DP, we also design a benchmark by leveraging the greedy approach in [37] (namely, GD-FF), which tries to provision each data-oriented vNF-SC greedily in the first-fit manner. To maintain sufficient statistical accuracy, we average the results from 10 independent runs to get each data point in the simulations.

B. Simulation Results and Analysis

The simulations evaluate the blocking probabilities of data-oriented vNF-SCs from the three algorithms at different traffic loads. We first set the probability that a pair of nodes in the optical DCI are directly connected by a lightpath as 0.2, which means that there are ~ 55 existing lightpaths in the network. Fig. 7 shows the results on blocking probability. It can be seen that as GD-FF cannot properly accomplish the establishing and task scheduling of data-oriented vNF-SCs, it provides the highest blocking probability. With DP, DLP-DP can serve the data-oriented vNF-SCs much better than GD-FF, but it is still not the best algorithm. Since CA-DP takes advantage of the fact that some vNF-SCs can tolerate a certain degree of "data-to-be-processed" delay and thus can give the way to other more urgent vNF-SCs, it achieves the best service provisioning and provides the lowest blocking probability.

Meanwhile, if we compare the results in Figs. 7(a)-7(c), we observe that the blocking probabilities from DLP-DP and CA-DP decrease when the number of DC nodes in the optical DCI increases. This is because when there are more DC nodes in the optical DCI, the number of existing vNFs also increases, which provides our DP-based algorithms a better chance to establish and schedule data-oriented vNF-SCs. On the other hand, because GD-FF is just a greedy heuristic, which cannot optimize the service provisioning of each data-oriented vNF-SC in a global manner (as our DP-based algorithms do), its performance on blocking probability in Figs. 7(a)-7(c) does not change much with the number of DC nodes.

Next, we increase the probability that a pair of nodes are directly connected by a lightpath to 0.4, and plot the results on blocking probability in Fig. 8. The results follow the similar trends as those in Fig. 7. Moreover, it is interesting to notice that when other conditions are the same, the performance gap between CA-DP and DLP-DP actually increases when there are more existing lightpaths in the optical DCI. Again, this is because CA-DP can resolve the conflicts among the provisioning schemes of data-oriented vNF-SCs in a global manner. Therefore, when there are more existing lightpaths, it has more flexibility to arrange the provisioning schemes of vNF-SCs and pushes the blocking probability even lower.

Table II shows the average SCT of each vNF-SC in different simulation scenarios. It can be seen that except for the scenarios with 16 DC nodes, the average SCT from CA-DP is always slightly longer than those from DLP-DP and GD-FF. This is because when the number of vNFs is limited in the optical DCI, CA-DP needs to schedule certain vNF-SCs to be finished right before their deadlines, for conflict avoidance. On the other hand, when there are 16 DC nodes, CA-DP can achieve an average SCT that is shorter than that from GD-FF. Therefore, the results in Table II confirm that CA-DP not only obtains the lowest blocking probability but also balances the tradeoff between blocking probability and average SCT well, especially when vNFs are abundant in the optical DCI.

TABLE II
RESULTS ON AVERAGE SCT (TS)

Number of DC Nodes	Lightpath Connectivity	CA-DP	DLP-DP	GD-FF
8	0.2	7.774	7.370	7.577
	0.4	7.144	6.744	6.520
12	0.2	7.177	6.996	7.143
	0.4	6.568	6.397	6.243
16	0.2	6.471	6.442	6.680
	0.4	5.910	5.875	5.980

TABLE III
AVERAGE RUNNING TIME PER VNF-SC REQUEST (MSEC)

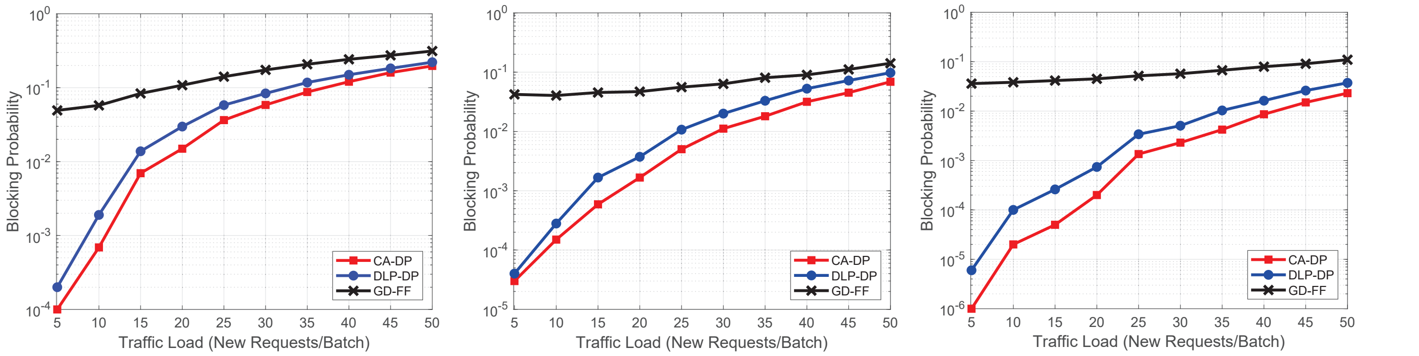
Algorithm	Number of DC Nodes		
	8	12	16
CA-DP	3.9071	4.6063	6.5300
DLP-DP	0.5036	0.5500	0.6320
GD-FF	0.4585	0.4668	0.4944

For the simulation scenarios considered in Figs. 7 and 8, we also obtain the average running time per vNF-SC request for each algorithm. Specifically, we categorize the scenarios into three cases according to the number of DC nodes in the optical DCI, average the algorithms' running time in each case, and list the results in Table III. It can be seen that all the algorithms serve each vNF-SC within a few milliseconds, which is fast enough to satisfy the requirement of dynamic provisioning. Meanwhile, we notice that the running time of CA-DP is the longest among the three algorithms. This is expected because according to our analysis in Section IV, the time complexity of CA-DP is the largest for realizing conflict avoidance. As the procedure of GD-FF is the simplest, it runs the fastest as shown in Table III. We also observe that the running time of the algorithms increases with the number of DC nodes. This is because the scale of the vNF-SC scheduling problem becomes larger when there are more DC nodes in the optical DCI.

Finally, we investigate how the deadline on the SCT of each data-oriented vNF-SC (*i.e.*, τ) affects the performance of the algorithms. Note that, for each vNF-SC, we set its deadline $\tau \in [\tau_{\min} + \delta_1, \tau_{\min} + \delta_2]$, where $\tau_{\min}$ is the lower-bound on the deadline. Hence, we fix δ_2 but increase δ_1, which means that all the vNF-SCs have a larger deadline on average. Fig. 9 illustrates the simulation results. As expected, the blocking probabilities from all the algorithms are reduced. Meanwhile, we can see that the performance gap between CA-DP and DLP-DP actually decreases. This is because if we increase δ_1, the deviation of the deadlines of vNF-SCs will decrease, which can make the advantage of CA-DP less significant.

VI. CONCLUSION

In this paper, we addressed the problem of how to jointly optimize the establishing and task scheduling of data-oriented vNF-SCs in an optical DCI, such that the probability of finishing the data-oriented vNF-SCs before their deadlines can be maximized. To make the best utilization of the resources in the optical DCI for meeting each vNF-SC's deadline, we leveraged DP to propose two time-efficient algorithms with the deadline-prioritized and conflict-aware approaches (*i.e.*, DLP-DP and CA-DP, respectively). The proposed algorithms were evaluated with extensive simulations, which confirmed that they outperform the greedy-based benchmark (GD-FF) significantly. Specifically, compared with GD-FF, DLP-DP and CA-DP can reduce the blocking probability of data-oriented



(a) 8 DC nodes

(b) 12 DC nodes

(c) 16 DC nodes

Fig. 9. Results on blocking probability (each pair of nodes are directly connected by a lightpath with a probability of 0.2, and a larger δ_1).

vNF-SCs by up to two magnitudes and maintain similar SCT. The results also showed that as CA-DP resolves the conflicts among the provisioning schemes of data-oriented vNF-SCs in a global manner, it always provides the lowest blocking probability among the algorithms.

ACKNOWLEDGMENTS

This work was supported in part by the National Key R&D Program of China (2020YFB1806400), NSFC project 61871357, SPR Program of CAS (XDC02070300), and Fundamental Funds for Central Universities (WK3500000006).

DISCLOSURES

The authors declare no conflicts of interest.

REFERENCES

[1] Cisco Visual Networking Index, 2017-2022. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html.

[2] P. Lu, L. Zhang, X. Liu, J. Yao, and Z. Zhu, "Highly efficient data migration and backup for Big Data applications in elastic optical inter-data-center networks," *IEEE Netw.*, vol. 29, pp. 36–42, Sept./Oct. 2015.

[3] V. Dukic, C. Gkantsidis, T. Karagiannis, F. Parmigiani, A. Singla, M. Filer, J. Cox, A. Ptasznik, N. Harland, W. Saunders, and C. Belady, "Beyond the mega-data center: networking multi-data center regions," in *Proc. of ACM SIGCOMM 2020*, pp. 765–781, Aug. 2020.

[4] "Network functions virtualization (NFV)," Tech. Rep., Oct. 2014. [Online]. Available: https://portal.etsi.org/Portals/0/TBpages/NFV/Docs/NFV_White_Paper3.pdf

[5] S. Mehraghdam, M. Keller, and H. Karl, "Specifying and placing chains of virtual network functions," in *Proc. of CloudNet 2014*, pp. 7–13, Oct. 2014.

[6] W. Fang, M. Zeng, X. Liu, W. Lu, and Z. Zhu, "Joint spectrum and IT resource allocation for efficient vNF service chaining in inter-datacenter elastic optical networks," *IEEE Commun. Lett.*, vol. 20, pp. 1539–1542, Aug. 2016.

[7] J. Liu, W. Lu, F. Zhou, P. Lu, and Z. Zhu, "On dynamic service function chain deployment and readjustment," *IEEE Trans. Netw. Serv. Manag.*, vol. 14, pp. 543–553, Sept. 2017.

[8] O. Gerstel, M. Jinno, A. Lord, and B. Yoo, "Elastic optical networking: a new dawn for the optical layer?" *IEEE Commun. Mag.*, vol. 50, pp. s12–s20, Feb. 2012.

[9] Z. Zhu, W. Lu, L. Zhang, and N. Ansari, "Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing," *J. Lightw. Technol.*, vol. 31, pp. 15–22, Jan. 2013.

[10] L. Gong, X. Zhou, X. Liu, W. Zhao, W. Lu, and Z. Zhu, "Efficient resource allocation for all-optical multicasting over spectrum-sliced elastic optical networks," *J. Opt. Commun. Netw.*, vol. 5, pp. 836–847, Aug. 2013.

[11] Y. Yin, H. Zhang, M. Zhang, M. Xia, Z. Zhu, S. Dahlfort, and S. Yoo, "Spectral and spatial 2D fragmentation-aware routing and spectrum assignment algorithms in elastic optical networks," *J. Opt. Commun. Netw.*, vol. 5, pp. A100–A106, Oct. 2013.

[12] M. Xia, M. Shirazipour, Y. Zhang, H. Green, and A. Takacs, "Network function placement for NFV chaining in packet/optical datacenters," *J. Lightw. Technol.*, vol. 33, pp. 1565–1570, Apr. 2015.

[13] X. Chen, Z. Zhu, R. Proietti, and B. Yoo, "On incentive-driven VNF service chaining in inter-datacenter elastic optical networks: A hierarchical game-theoretic mechanism," *IEEE Trans. Netw. Serv. Manag.*, vol. 16, pp. 1–12, Mar. 2019.

[14] B. Li, W. Lu, and Z. Zhu, "Deep-NFVOrch: Leveraging deep reinforcement learning to achieve adaptive vNF service chaining in EON-DCIs," *J. Opt. Commun. Netw.*, vol. 12, pp. A18–A27, Jan. 2020.

[15] W. Lu and Z. Zhu, "Malleable reservation based bulk-data transfer to recycle spectrum fragments in elastic optical networks," *J. Lightw. Technol.*, vol. 33, pp. 2078–2086, May 2015.

[16] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," in *Proc. of GCE 2008*, pp. 12–16, Nov. 2008.

[17] H. Kim, S. Bird, D. Brooks, S. Chintala, U. Diril, D. Dzhulgakov, M. Fawzy, B. Jia, Y. Jia, A. Kalro, J. Law, K. Lee, J. Lu, P. Noordhuis, M. Smelyanskiy, L. Xiong, and X. Wang, "Applied machine learning at Facebook: A datacenter infrastructure perspective," in *Proc. of HPCA 2018*, pp. 620–629, Feb. 2018.

[18] "IETF service function chaining (SFC)," Tech. Rep., Apr. 2014. [Online]. Available: https://datatracker.ietf.org/wg/sfc/charter

[19] Z. Yang, B. Chen, M. Dai, G. Su, and R. Lin, "VNF placement for service chaining in IP over WDM networks," in *Proc. of ACP 2018*, pp. 1–3, Oct. 2018.

[20] B. Li, W. Lu, S. Liu, and Z. Zhu, "Deep-learning-assisted network orchestration for on-demand and cost-effective vNF service chaining in inter-DC elastic optical networks," *J. Opt. Commun. Netw.*, vol. 10, pp. D29–D41, Oct. 2018.

[21] Z. Wang, J. Zhang, T. Huang, and Y. Liu, "Service function chain composition, placement, and assignment in data centers," *IEEE Trans. Netw. Serv. Manag.*, vol. 16, pp. 1638–1650, Dec. 2019.

[22] Y. Xiao, Q. Zhang, F. Liu, J. Wang, M. Zhao, Z. Zhang, and J. Zhang, "NFVdeep: Adaptive online service function chain deployment with deep reinforcement learning," in *Proc. of IWQoS 2019*, pp. 1–10, Jun. 2019.

[23] K. Han, S. Li, S. Tang, H. Huang, S. Zhao, K. Han, G. Fu, and Z. Zhu, "Application-driven end-to-end slicing: When wireless network virtualization orchestrates with NFV-based mobile edge computing," *IEEE Access*, vol. 6, pp. 26 567–26 577, 2018.

[24] F. Tashtarian, M. Zhani, B. Fatemipour, and D. Yazdani, "CoDeC: A cost-effective and delay-aware SFC deployment," *IEEE Trans. Netw. Serv. Manag.*, vol. 17, pp. 793–806, Jun. 2020.

[25] G. Sun, Z. Xu, H. Yu, X. Chen, V. Chang, and A. Vasilakos, "Low-latency and resource-efficient service function chaining orchestration in network function virtualization," *IEEE Internet Things J.*, vol. 7, pp. 5760–5772, Jul. 2020.

[26] J. Li, W. Shi, Q. Ye, N. Zhang, W. Zhuang, and X. Shen, "Multi-service function chain embedding with delay-guarantee: A game-theoretical approach," *IEEE Internet Things J., in Press*, pp. 1–14, Jan. 2021.

[27] Q. Chen, M. Zhu, T. Shen, J. Gu, C. Yan, and P. Gu, "Deep-reinforce-learning-assisted network orchestration for VNF-SC provisioning in inter-DC elastic optical networks," in *Proc. of ACP 2020*, pp. 1–3, Oct. 2020.

[28] L. Gong and Z. Zhu, "Virtual optical network embedding (VONE) over elastic optical networks," *J. Lightw. Technol.*, vol. 32, pp. 450–460, Feb. 2014.

[29] L. Gong, H. Jiang, Y. Wang, and Z. Zhu, "Novel location-constrained virtual network embedding (LC-VNE) algorithms towards integrated node and link mapping," *IEEE/ACM Trans. Netw.*, vol. 24, pp. 3648–3661, Dec. 2016.

[30] M. Zeng, W. Fang, and Z. Zhu, "Orchestrating tree-type VNF forwarding graphs in inter-DC elastic optical networks," *J. Lightw. Technol.*, vol. 34, pp. 3330–3341, Jul. 2016.

[31] K. Christodoulopoulos, I. Tomkos, and E. Varvarigos, "Elastic bandwidth allocation in flexible OFDM-based optical networks," *J. Lightw. Technol.*, vol. 29, pp. 1354–1366, May 2011.

[32] L. Gong, X. Zhou, W. Lu, and Z. Zhu, "A two-population based evolutionary approach for optimizing routing, modulation and spectrum assignments (RMSA) in O-OFDM networks," *IEEE Commun. Lett.*, vol. 16, pp. 1520–1523, Sept. 2012.

[33] Y. Wang, P. Lu, W. Lu, and Z. Zhu, "Cost-efficient virtual network function graph (vNFG) provisioning in multidomain elastic optical networks," *J. Lightw. Technol.*, vol. 35, pp. 2712–2723, Jul. 2017.

[34] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. Turck, and S. Davy, "Design and evaluation of algorithms for mapping and scheduling of virtual network functions," in *Proc. of NetSoft 2015*, pp. 1–9, Jun. 2015.

[35] L. Qu, C. Assi, and K. Shaban, "Delay-aware scheduling and resource optimization with network function virtualization," *IEEE Trans. Commun.*, vol. 64, pp. 3746–3758, Sept. 2016.

[36] H. Alameddine, S. Sebbah, and C. Assi, "On the interplay between network function mapping and scheduling in VNF-based networks: A column generation approach," *IEEE Trans. Netw. Serv. Manag.*, vol. 14, pp. 860–874, Dec. 2017.

[37] J. Li, W. Shi, P. Yang, and X. Shen, "On dynamic mapping and scheduling of service function chains in SDN/NFV-enabled networks," in *Proc. of GLOBECOM 2019*, pp. 1–6, Dec. 2019.

[38] J. Ullman, "NP-complete scheduling problems," *J. Comput. Syst. Sci.*, vol. 10, pp. 384–393, Jun. 1975.