On the Deployment and Operation of Correlated Data-Intensive vNF-SCs in Inter-DC EONs

Zuqing Zhu†, Liang Zhang‡, Bijan Jabbari‡
†School of Information Science and Technology, University of Science and Technology of China, China, {zqzhu}@ieee.org
‡Department of Electrical and Computer Engineering, George Mason University, USA, {lzhang36, bjabbari}@gmu.edu

***Abstract*—In this work, we study the problem of deploying and operating correlated data-intensive vNF-SCs in inter-datacenter elastic optical networks. Requiring for a set of correlated data-intensive vNF-SCs, the service completion time (SCT) of a network service is related to the maximum branch completion time of those vNF-SC branches, making the correlation awareness of importance for the problem optimization. Being aware of it, we propose a dynamic programming based optimization scheme for the deployment and operation of single vNF-SC branch and two correlation-aware service provisioning algorithms to minimize the average SCT of network services. Simulation results verified that the proposed algorithms can effectively reduce the average SCT of network services compared with a benchmark algorithm that ignores the importance of correlation awareness.**

I. Introduction

Network function virtualization (NFV) [1, 2] is bringing revolutionary changes to how the networks are architected. With NFV, service providers can leverage standard IT virtualization technologies to instantiate various virtual network functions (vNFs) flexibly and dynamically on commodity hardwares that are largely located in data-centers (DCs). Then, to deploy a new network service timely, service providers only need to deploy vNFs in DCs and steer the application traffic through them in sequence, *i.e.*, forming a vNF service chain (vNF-SC) [3]. Meanwhile, it is known that many emerging network services are bandwidth-intensive and/or data-intensive, *e.g.*, big data analysis [4, 5], multimedia services [6, 7] and DC backup [8]. To support them adaptively, the capacity and flexibility of the physical infrastructure of inter-DC networks become critical, since the traffic flowing through the vNF-SCs would exhibit high peak throughput and high burstiness [9]. Fortunately, recent advances on elastic optical networks (EONs) have confirmed that agile bandwidth management in the optical layer is attainable [10–14]. With this advantage, inter-DC EON becomes a promising infrastructure to support bandwidth-intensive/data-intensive vNF-SCs efficiently [3].

The problem of deploying and operating vNF-SCs involves both vNF-SC embedding and traffic steering. Previously, people have studied how to deploy vNFs or vNF-SCs and steer bandwidth-intensive traffic in inter-DC networks [3, 15–17]. Nevertheless, these studies only considered the vNFs or vNF-SC deployment and traffic steering for bandwidth-intensive network services, while after the vNFs or vNF-SCs having been deployed, how to operate them to deliver high-quality data-intensive services to the clients has not been addressed yet. Specifically, to operate a data-intensive vNF-SC, service providers need to accomplish two tasks: 1) scheduling the computing tasks in required vNFs, and 2) transferring application data between two adjacent vNFs in the vNF-SC. Note that, both of these two tasks can affect the service completion time (SCT) of a client, which is an important metric to measure the quality-of-service (QoS) of data-intensive vNF-SCs. For instance, if a client requests for a DC backup service with data encryption on intermediate vNF(s), a shorter SCT means that the client's data can be evacuated from the endangered DC, and transferred to and stored in a secure DC more quickly.

Note that, bandwidth-intensive vNF-SCs might consume a lot of bandwidth and even cause congestions on certain links. This would apparently affect the SCTs of data-intensive vNF-SCs, since it would be easily for the second task (*i.e.*, transferring application data) to be handled poorly. Moreover, we should notice that the data transfer between two adjacent vNFs is usually not continuous in time. This is because the data transfer usually would not be started until the computing task in the previous vNF has been done. Hence, to accomplish the second task, we need to not only find an available path to transfer data between each two adjacent vNFs in the vNF-SC, but also schedule the data transfer in the time domain.

In this work, we study the problem of deploying and operating correlated data-intensive vNF-SCs in inter-DC EONs. We consider a practical scenario in which a network service consists of a source, a set of parallel and correlated data-intensive vNF-SCs, and multiple destinations, to emulate the network service such as big data analysis [4] or DC backup [8]. Due to the correlation between parallel vNF-SCs, such a service's SCT would be the time when all the data from the source have been processed by the designated vNF-SCs and been delivered to the destinations. Being aware of this, we propose two correlation-aware algorithms to minimize the average SCT of such services. Note that, in a practical inter-DC EON, there should be dynamic background traffic other than that from the vNF-SCs, which would generate two-dimensional (2D) spectrum fragments on the fiber links [18]. Since these 2D fragments can be leveraged to accomplish spectrum-efficient data transfers in EONs [18, 19], our algorithms schedule the bulk-data transfers with them. Simulation results show that the proposed algorithms can reduce the average SCT effectively.

The rest of the paper is organized as follows. Section II provides the problem description. In Section III, we propose several algorithms for the deployment and operation of correlated data-intensive vNF-SCs in inter-DC EONs. Section

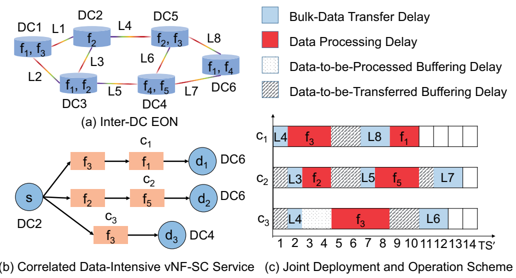


Fig. 1. Example on deploying and operating correlated data-intensive vNF-SCs in an inter-DC EON.

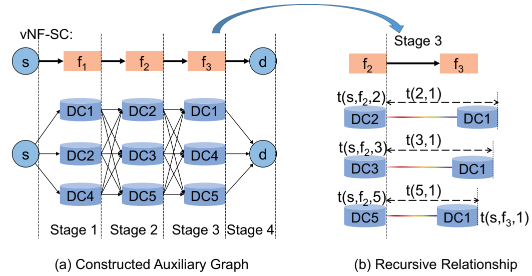


Fig. 2. Example on optimizing the deployment and operation of single data-intensive vNF-SC branch with an AG and a DP-based scheme.

IV evaluates the proposed algorithms with simulation results. Finally, Section V summarizes the paper.

II. Problem Description

We model the inter-DC EON as $G(V, L)$, where V is the DC set and L is the set of established lightpaths to interconnect the DCs. Fig. 1(a) shows an intuitive example on the inter-DC EON, which consists of 6 DCs and has 8 lightpaths[1]. Each DC $v \in V$ has certain vNF(s) deployed on it already, and there are $\mathcal{F}$ types of vNFs supported by the inter-DC EON in total. For type f vNF, the set of DCs that have such a vNF is $V_f \subseteq V$. We assume that the inter-DC EON operates as a discrete-time system [8], *i.e.*, the network operation status changes every time slot (TS). Each type f vNF can process α_f amount of data in a TS, and since the data volume can change after each processing [3], the vNF's output-to-input data volume ratio is β_f. For each DC pair $(v, u) \in V^2$, the set of established lightpaths is $L_{v,u} \subset L$. Due to the dynamics of background traffic, 2D spectrum fragments would be generated on those lightpaths, which however can be utilized for bulk-data transfers between the vNFs in data-intensive vNF-SCs.

A data-oriented vNF-SC is denoted as $c = \{s, d, b, F\}$, where s is its source, d is its destination, b is the volume of initial data generated at the source, and $F = \{f_1, \cdots, f_{|F|}\}$ is the sequence of required vNFs where operator $|\cdot|$ returns the number of elements in a set. After all the required vNFs have been deployed on selected DCs, service providers need to conduct task scheduling and bulk-data transfer for operating the data-intensive vNF-SC. For each vNF $f_j \in F$, once it being deployed in DC $v_{f_j} \in V_{f_j}$, its task processing is not allowed to be interrupted and the start time is denoted as δ_j. By dividing the amount of its input data by α_{f_j}, we can get the data processing latency as p_j. After all the input data has been processed, a 2D fragment is needed to transfer data from f_j to f_{j+1}. Then, the data transmission latency t_j can be obtained by dividing the amount of output data from f_j by the bandwidth of the selected 2D fragment, and the start time of the data transmission γ_j is that of the selected 2D fragment. As a bulk-data transfer cannot be started before all the output data of a computing task being processed by a vNF has been obtained and a vNF cannot start to process a computing task before it has received all the input data, the following constraints should be satisfied to ensure successful operation:

$$\begin{cases} \delta_j + p_j \leq \gamma_j, \\ \gamma_j + t_j \leq \delta_{j+1}, \end{cases} \quad \forall j. \tag{1}$$

Note that, when $\gamma_j > \delta_j + p_j$, the processed data cannot be transferred immediately after it becomes available and thus there would be a "data-to-be-transferred" buffering delay as $[\gamma_j - (\delta_j + p_j)]$. Similarly, when $\delta_{j+1} > (\gamma_j + t_j)$, vNF f_{j+1} is occupied when the data from vNF f_j arrives, leading to a "data-to-be-processed" buffering delay as $[\delta_{j+1} - (\gamma_j + t_j)]$. Then, we can see that the SCT of a vNF-SC includes four parts: 1) the total data processing latency in all of its vNFs, 2) the total data transmission latency on all of its lightpaths, 3) the total "data-to-be-processed" buffering delay, and 4) the total "data-to-be-transferred" buffering delay. Apparently, the first part is constant and cannot be reduced, and hence we should focus on minimizing the remaining three parts.

Further, a network service that consists of several correlated data-intensive vNF-SCs is denoted as $R = \{c_1, \cdots, c_{|R|}\}$, where c_i is the i-th data-intensive vNF-SC. Fig. 1(b) shows a network service that consists of three correlated data-intensive vNF-SCs. Since all the vNF-SCs in R have the same source s, we can regard each of them as a branch of the network service. If we define the SCT of such a branch as branch completion time (BCT), the SCT of the network service R would be the maximum BCT of all its branches. For instance, the scheme in Fig. 1(c) makes the network service in Fig. 1(b) have an SCT of 13 TS'. Note that, as the maximum BCT of its branches determines a service's SCT, the deployment and operation schemes of its other branches become flexible before the SCT. For example, in Fig. 1(c), if we schedule the data processing of vNF f_1 in c_1 to be started at $t = 12$ TS', the service's SCT is still 13 TS'. In light of this, we should utilize the correlation between vNF-SC branches to optimize the SCT of such services. Besides, as the problem is at least as complex as the task scheduling problem which has been proved to be $\mathcal{NP}$-hard in [20], it is $\mathcal{NP}$-hard too. Hence, we can directly turn to design heuristics to solve it time-efficiently.

III. Proposed Deployment and Operation Schemes for Correlated Data-Intensive vNF-SCs

A. DP-based Optimization for Single vNF-SC Branch

As explained above, to deploy and operate a data-intensive vNF-SC branch, we need to 1) select DCs to deploy requested vNFs, 2) schedule computing tasks with available TS' for being processed by the vNFs in selected DCs, and 3) schedule

[1]For simplicity, we ignore the intermediate optical switches on the lightpaths and thus each pair of DCs are directly connected in Fig. 1(a).

bulk-data transfers with 2D fragments on lightpaths to transmit data between adjacent vNFs. To solve the sub-problems, we first construct an auxiliary graph (AG) in which all the feasible solutions of vNF-SC deployment are accessible but unevaluated. As illustrated in Fig. 2, there is a data-intensive vNF-SC requiring $F = \{f_1, f_2, f_3\}$, since we have $V_{f_1} = \{1, 2, 4\}$, $V_{f_2} = \{2, 3, 5\}$, and $V_{f_3} = \{1, 4, 5\}$, we can construct its AG as that in Fig. 2(a). Then, to find the optimal vNF-SC deployment and operation scheme that has a minimum SCT, we divide the AG into several stages, each of which corresponds to the computing task scheduling in vNF f_j as well as the bulk-data transfer scheduling between vNFs f_j with f_{j+1}, and propose a dynamic programming (DP) based optimization scheme, the recursive relation of which is:

$$t\Big(s, f_{j+1}, v_{f_{j+1}}\Big) = \min_{v_{f_j} \in V_{f_j}} \Big[t(s, f_j, v_{f_j}) + t(v_{f_j}, v_{f_{j+1}})\Big], \quad (2)$$

where $t(s, f, v_f)$ is the delay from the TS when the vNF-SC starts to when vNF f on DC v_f finishes its computing task, and $t(v_{f_j}, v_{f_{j+1}})$ is the duration in between the completion time of the computing task in vNF f_j on DC v_{f_j} and that of the computing task in vNF f_{j+1} on DC $v_{f_{j+1}}$, as explained in Fig. 2(b). By calculating $t(s, f_{j+1}, v_{f_{j+1}})$ for $v_{f_{j+1}} \in V_{f_{j+1}}$ in each stage, we can get the minimum SCT of the vNF-SC (*i.e.*, $t(s, d)$) and solve the three subproblem simultaneously.

Here, to minimize $t(v_{f_j}, v_{f_{j+1}})$, we select the earliest and largest 2D spectrum fragment on established lightpaths from v_{f_j} to $v_{f_{j+1}}$, which can minimize both the "data-to-be-transferred" buffering delay and data transmission latency. Meanwhile, to minimize the "data-to-be-processed" buffering delay on v_{f_j}, we try to find the earliest start time of the computing task with task rescheduling. Specifically, for those computing tasks that have already been scheduled on v_{f_j}, their scheduling schemes could be flexible with the existence of "data-to-be-transferred" buffering delay. Because of it, we can perform task rescheduling for those tasks to squeeze a earliest service time window for the target computing task. To realize it, we leverage the scheduling algorithm that we designed in [21], which combines the binary search and the minimum late task scheduling (MLTS) algorithm [22] and has a time complexity of $O(N^2 \cdot \log([t(f_j)]_{\min}^{\max})$ where N is the number of rescheduled tasks and $[t(f_j)]_{\min}^{\max}$ is the time interval between the earliest start time with/without task rescheduling. Finally, the overall time complexity of the DP-based optimization is $O(|\mathcal{F}| \cdot |V|^2 \cdot \widehat{N}^2 \cdot \log(\widehat{[t(f_j)]_{\min}^{\max}}))$, where operation $\widehat{\cdot}$ returns the maximum value of a variable.

B. Correlation-Aware Service Provisioning Algorithm with DP-Based Optimization (CASP w/ DP)

Based on the proposed DP-based optimization scheme, we propose a correlation-aware service provisioning algorithm to minimize the average SCT of correlated data-intensive vNF-SC services. First, to emphasize the importance of correlation awareness on the optimization, we gives an example in Fig. 3, where the performance of the provisioning schemes with/without correlation-awareness in terms of average SCT are compared. There are two network services, each of which

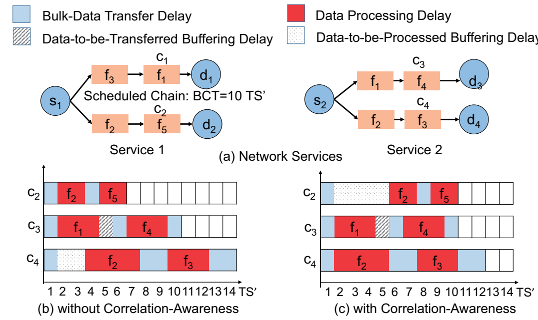


Fig. 3. Example on the importance of correlation awareness.

consists of two data-intensive vNF-SC branches. For *Service* 1, its first branch c_1 has already been deployed and scheduled with a BCT of 10 TS'. Under this premise, the provisioning scheme without correlation awareness would decide the deployment and operation schemes for the two network services one by one. As shown in Fig. 3(b), the sequence of branches to be deployed and scheduled is $\{c_2, c_3, c_4\}$, and we get the SCTs of *Services* 1 and 2 as 10 and 14 TS', respectively. Hence, the average SCT of the services is 12 TS'.

The provisioning scheme with correlation-awareness deploys and schedules the data-intensive vNF-SC branches based on their priorities, which depend on their effects on the average SCT. For example, as the branch c_1 of *Service* 1 having a BCT of 10 TS', the deployment and operation scheme of c_2 becomes flexible as long as its BCT is shorter than 10 TS'. Being aware of this, we deploy and schedule the most urgent branch in each decision round, and get the sequence of branches to be deployed and scheduled as $\{c_3, c_2, c_4\}$, which results in the SCTs of *Services* 1 and 2 becoming 10 and 12 TS', respectively, as shown in Fig. 3(c). Therefore, the average SCT is shorten to 11 TS', which verifies the importance of correlation awareness on minimizing the average SCT.

Grounded by the idea of serving the most urgent branch first while being aware of the correlation between branches, our proposed correlation-aware service provisioning algorithm with DP-based optimization (abbreviated as "CASP w/ DP") is elaborated in *Algorithm* 1. Here, we define several notations: C_p is the set of pending branches that are waiting for being deployed and scheduled, C_p^{dd} is the subset of C_p to store all the deadline-driven branches and it is initialized as $\emptyset$, C_p^{∞} is another subset of C_p to store the remaining branches, which satisfies $C_p = C_p^{dd} \cup C_p^{\infty}$ and is initialized as C_p, and t_h is a preset threshold to determine whether a deadline-driven branch has become urgent. *Line* 2 obtains the latest network status. *Lines* 3-5 calculate the BCT for all pending branches in C_p based on the proposed DP-based scheme to find the potentially-urgent branches. Then, to find the most urgent branch, we first search it in C_p^{dd} by selecting the branch that has the minimum gap between its BCT and deadline (*Line* 7). If the selected branch is not urgent enough, we continue to search it in C_p^{∞} by selecting the branch that has the smallest BCT among all the longest branches in network services (*Lines* 12 and 17). Otherwise, the selected branch in C_p^{dd} is regarded as the most urgent one. Next, we deploy and schedule the most urgent branch with the DP-based scheme and update the defined sets (*Lines* 9, 10, 13, 14, 18 and 19). The time complexity of CASP w/ DP is $O(|\widehat{C_p}|^2 \cdot O(DP))$, where $O(DP)$ is the time complexity of the DP-based scheme.

Algorithm 1: Correlation-Aware Service Provisioning Algorithm with DP-Based Optimization (CASP w/ DP)

1 **while** $C_p \neq \emptyset$ **do**
2 obtain the latest network status;
3 **for** *each vNF-SC branch in* C_p **do**
4 calculate its BCT with the DP-based scheme;
5 **end**
6 **if** $C_p^{dd} \neq \emptyset$ **then**
7 select the vNF-SC branch with minimum gap between its BCT and deadline in C_p^{dd};
8 **if** *the minimum gap is smaller than* t_h **then**
9 deploy/schedule selected branch in C_p^{dd} with DP;
10 update C_p, C_p^{dd} and deadline of branches in C_p^{dd};
11 **else**
12 select the branch with the smallest BCT among all the longest branches in C_p^{∞};
13 deploy/schedule selected branch in C_p^{∞} with DP;
14 update C_p, C_p^{dd}, C_p^{∞}, deadline of branches in C_p^{dd};
15 **end**
16 **else**
17 select the branch with the smallest BCT among all the longest branches in C_p^{∞};
18 deploy/schedule selected branch in C_p^{∞} with DP;
19 update C_p, C_p^{dd}, C_p^{∞}, deadline of branches in C_p^{dd};
20 **end**
21 **end**

C. Correlation-Aware Service Provisioning Algorithm with Feature-Based Optimization (CASP w/ BF)

Considering the non-negligible time complexity of the DP-based scheme in *Algorithm* 1, we try to design a feature-based optimization scheme to replace the calculation in *Line* 4, for finding the most urgent branch more quickly. We abbreviate the new scheme as "CASP w/ BF". For the four parts in the BCT of a vNF-SC branch, the total data processing latency is independent of network status and thus can be calculated exactly, while the remaining three parts can be estimated based on those of served/scheduled branches. To do that, for each lightpath, we record the data transmission latency and "data-to-be-transferred" buffering delay of each served/scheduled branch that goes through it, and obtain the average data transmission latency per unit data and the average "data-to-be-transferred" buffering delay. Regarding the "data-to-be-processed" buffering delay on the vNFs in selected DCs, we estimate it similarly. Finally, the BCT of a branch can be estimated by calculating $t(v_{f_j}, v_{f_{j+1}})$ in Eq. (2) as:

$$t\Big(v_{f_j}, v_{f_{j+1}}\Big) = \min_{l \in L_{v_{f_j}, v_{f_{j+1}}}} \Big[b_{f_j} \cdot (td_l^* + tb_l^* + pd_{v_{f_{j+1}}}^*) + p_{f_{j+1}}^* \Big], \tag{3}$$

where b_{f_j} is the amount of data from v_{f_j}, td_l^* is the average bulk-data transfer delay per unit data on lightpath l, tb_l^* is the average "data-to-be-transferred" buffering delay on lightpath l, $pd_{v_{f_{j+1}}}^*$ is the average "data-to-be-processed" buffering delay on $v_{f_{j+1}}$, and $p_{f_{j+1}}^*$ is the data processing latency on $v_{f_{j+1}}$. By combining Eqs. (2) and (3), we can get the estimated BCT of a pending vNF-SC branch. The time complexity of CASP w/ BF is $O(|\widehat{C_p}|^2 \cdot (|L| + |V| \cdot |\mathcal{F}|) + |\widehat{C_p}| \cdot O(DP))$.

IV. Performance Evaluation

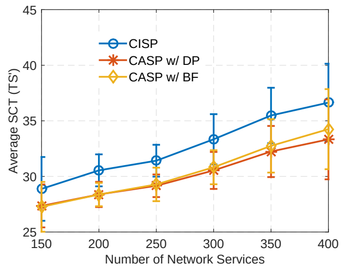
We evaluate the proposed algorithms with the 14-node NSFNET topology [10]. Here, each node in the topology is a DC node, and between each DC pair, there are $[1, 2]$ established lightpaths, each of which has 11 FS'. The background traffic occupies the lightpaths' bandwidth dynamically along the time axis and leaves 12.02% and 2.65% on average in the low and high traffic scenarios, respectively. The number of deployed vNFs on each DC is within $[2, 4]$ and there are 10 types of vNFs in total. In each simulation, the network services are generated dynamically according to the Poisson traffic model. Each of them asks for 3 correlated vNF-SC branches on average, the average number of vNFs in a vNF-SC is 5, and its initial data volume is uniformly distributed within $[1, 3]$ FS·TS. The vNFs' processing rates are within $[0.56, 1.12]$, and their output-to-input data volume ratios are within $[0.7, 1.3]$. The threshold t_h in *Algorithm* 1 is set to 5 TS'. For comparison, we adopt the service provisioning algorithm without correlation awareness (abbreviated as "CISP") as the benchmark.

TABLE I
Running Time per Network Service in Low Background Traffic Scenario (seconds)

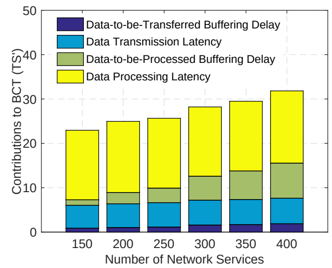
# of Network Services	Average Running Time per Network Service		
	CISP	CASP w/ DP	CASP w/ BF
200	0.68	4.64	0.92
300	0.89	7.64	1.23
400	0.98	9.77	1.39

Fig. 4 shows the results in the low background traffic scenario. In Fig. 4(a), we observe that the proposed algorithms achieve much lower average SCTs than the benchmark. More importantly, CASP w/ BF can achieve similar results as CASP w/ DP. In Fig. 4(b), the "data-to-be-transferred" buffering delay is small, while the data processing latency dominates the BCT of each branch. This is because, when the background traffic is low, there are sufficient 2D fragments to transmit the processed data immediately. Fig. 4(c) shows the average variance of correlated BCTs. Here, a lower BCT variance means that the correlation-awareness scheme functions better, and our algorithms achieve much lower BCT variances than the benchmark. Moreover, as the number of network services increases, the BCT variances from our algorithms decrease rapidly, while there is no such trend in the benchmark's results. To evaluate the time efficiency of the algorithms, Table I compares their running time per network service, in which CASP w/ DP needs much longer time than CASP w/ BF and the benchmark. To this end, in view of the similar results on average SCT in Fig. 4(a) and the shorter time in Table I, CASP w/ BF is proved to be more effective than CASP w/ DP.

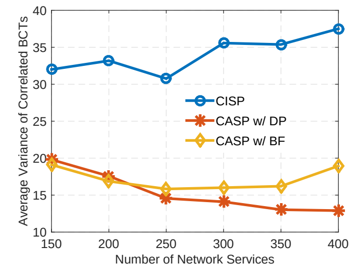
Fig. 5 shows the results in the high background traffic scenario, which still confirms the advantage of our proposed algorithms. However, the performance gaps on the average SCT becomes slightly smaller. This is because in the high



(a) Average SCT of network services

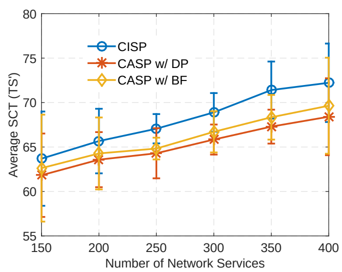


(b) Average BCT of vNF-SC branches

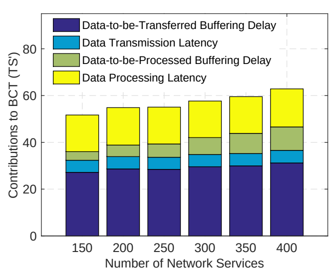


(c) Average variance of correlated BCTs

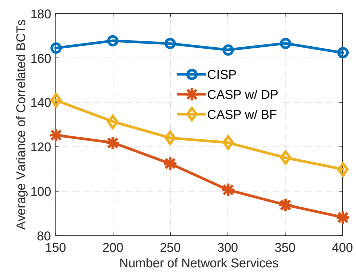
Fig. 4. Results in low background traffic scenario.



(a) Average SCT of network services



(b) Average BCT of vNF-SC branches



(c) Average variance of correlated BCTs

Fig. 5. Results in high background traffic scenario.

background traffic scenario, the processed data has less transmission opportunities, which would affect the estimation accuracy of the proposed feature-based optimization scheme.

V. Conclusion

In this paper, we studied how to deploy and operate correlated data-intensive vNF-SCs to minimize their average SCT. We proposed a DP-based scheme for the deployment and operation of single vNF-SC branch and two correlation-aware algorithms to optimize the average SCT of network services. Simulation results verified the effectiveness of our proposals.

References

[1] J. Liu *et al.*, "On dynamic service function chain deployment and readjustment," *IEEE Trans. Netw. Serv. Manag.*, vol. 14, pp. 543–553, Sept. 2017.

[2] L. Gong, Y. Wen, Z. Zhu, and T. Lee, "Toward profit-seeking virtual network embedding algorithm via global resource capacity," in *Proc. of INFOCOM 2014*, pp. 1–9, Apr. 2014.

[3] W. Fang *et al.*, "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.

[4] X. Chen *et al.*, "Deep-RMSA: A deep-reinforcement-learning routing, modulation and spectrum assignment agent for elastic optical networks," in *Proc. of OFC 2018*, pp. 1–3, Mar. 2018.

[5] R. Proietti *et al.*, "Experimental demonstration of machine-learning-aided QoT estimation in multi-domain elastic optical networks with alien wavelengths," *J. Opt. Commun. Netw.*, vol. 11, pp. A1–A10, Jan. 2019.

[6] Z. Pan *et al.*, "Advanced optical-label routing system supporting multicast, optical TTL, and multimedia applications," *J. Lightw. Technol.*, vol. 23, pp. 3270–3281, Oct. 2005.

[7] P. Lu, Q. Sun, K. Wu, and Z. Zhu, "Distributed online hybrid cloud management for profit-driven multimedia cloud computing," *IEEE Trans. Multimedia*, vol. 17, pp. 1297–1308, Aug. 2015.

[8] J. Yao, P. Lu, L. Gong, and Z. Zhu, "On fast and coordinated data backup in geo-distributed optical inter-datacenter networks," *J. Lightw. Technol.*, vol. 33, pp. 3005–3015, Jul. 2015.

[9] P. Lu *et al.*, "Highly-efficient data migration and backup for Big Data applications in elastic optical inter-datacenter networks," *IEEE Netw.*, vol. 29, pp. 36–42, Sept./Oct. 2015.

[10] 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.

[11] W. Shi, Z. Zhu, M. Zhang, and N. Ansari, "On the effect of bandwidth fragmentation on blocking probability in elastic optical networks," *IEEE Trans. Commun.*, vol. 61, pp. 2970–2978, Jul. 2013.

[12] L. Gong *et al.*, "Efficient resource allocation for all-optical multicasting over spectrum-sliced elastic optical networks," *J. Opt. Commun. Netw.*, vol. 5, pp. 836–847, Aug. 2013.

[13] Y. Yin *et al.*, "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.

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

[15] M. Xia *et al.*, "Network function placement for NFV chaining in packet/optical datacenters," *J. Lightw. Technol.*, vol. 33, pp. 1565–1570, Apr. 2015.

[16] 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.

[17] 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.

[18] 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.

[19] Z. Xu and Z. Zhu, "On establishing and task scheduling of data-oriented vNF-SCs in an optical DCI," *J. Opt. Commun. Netw.*, vol. 14, pp. 89–99, Mar. 2022.

[20] J. Riera *et al.*, "Virtual network function scheduling: Concept and challenges," in *Proc. of SaCoNeT 2014*, pp. 1–5, Jun. 2014.

[21] W. Lu, L. Liang, and Z. Zhu, "Orchestrating data-intensive vNF service chains in inter-DC elastic optical networks," in *Proc. of ONDM 2017*, pp. 1–6, May 2017.

[22] S. Dauzère-Pérès, "Minimizing late jobs in the general one machine scheduling problem," *Eur. J. Oper. Res.*, vol. 81, pp. 134–142, Feb. 1995.