Optimized Service Chain Mapping and reduced flow processing with Application-Awareness

Deepak Nadig, Byrav Ramamurthy, Brian Bockelman, and David Swanson
Dept. of Computer Science & Engg, University of Nebraska-Lincoln, Lincoln, NE 68588-0115, USA.
Email: {deepaknadig,byrav,bbockelm,dswanson}@cse.unl.edu

***Abstract*—Network Function Virtualization (NFV) brings a new set of challenges when deploying virtualized services on commercial-off-the-shelf (COTS) hardware. Network functions can be dynamically managed to provide the necessary services on-demand and further, services can be chained together to form a larger composite. In this paper, we address an important technical problem of mapping service function chains (SFCs) across different data centers with the objective of reducing the flow processing costs. We develop an integer linear programming (ILP) formulation to optimally map service function chains to multiple data centers while adhering to the data center's capacity constraints. We propose a novel application-aware flow reduction (AAFR) algorithm to simplify the SFC-ILP to significantly reduce the number of flows processed by the SFCs. We perform a thorough study of the SFC mapping problem for multiple data centers and evaluate the performance of our proposed approach with respect to three parameters: i) impact of number of SFCs and SFC length on flow processing cost, ii) capacitated/uncapacitated flow processing cost gains, and iii) balancing flow-to-SFC mappings across data centers. Our evaluations show that our proposed AAFR algorithm reduces flow-processing costs by 70% for the capacitated-SFC mapping case over the SFC-ILP. In addition, our uncapacitated AAFR (AAFR-U) algorithm provides a further 4.1% cost-gain over its capacitated counterpart (AAFR-C).**

***Index Terms*—Software Defined Networks; Network Functions Virtualization; Service Chaining; Application-awareness.**

I. Introduction

Recently, several evolutionary trends in networking such as software defined networking (SDN) and network functions virtualization (NFV) have had a significant impact on the next-generation of network architectures. SDN introduces the network control and data plane separation and allows dynamic and programmatic control of the network via open interfaces. NFV, on the other hand, utilizes traditional server virtualization techniques to provide an architecture where network functions run over commercial-off-the-shelf (COTS) hardware rather than dedicated boxes. Despite NFV presenting many new opportunities, challenges exist in NFV placement and SFC mapping across data centers. Placing network/service functions far from the user results in additional delays. Security services provided via NFV infrastructure for large networks are expected to support context-aware and low-latency applications in a highly efficient manner. The introduction of security services at the network edge, while reducing the response time, may impact core-network utilization. To address the above challenges, we study the SFC mapping and placement problem across multiple data centers in this work. We focus on solving the SFC mapping problem to provide security services to both interactive and large-volume data transfers. We model our solution to suit an existing 100G production network topology, which is an important difference compared to previous works.

The main contributions of this paper are as follows: i) We formulate an integer linear programming (ILP) problem (SFC-ILP) for optimized SFC mapping in a multi-data center topology and propose an application-aware flow reduction (AAFR) algorithm to reduce the NFV flow processing workloads. Traffic and resource characteristics of a testbed network consisting of four data centers are employed in the proposed ILP model, ii) We compare the results of AAFR with the SFC-ILP formulations for a production U.S. CMS Tier-2 network, iii) We conduct extensive performance evaluations of the proposed AAFR algorithm and show flow processing workload savings of $47\% - 70\%$ can be achieved for the capacitated-SFC mapping problem, and iv) we demonstrate the AAFR algorithm's effectiveness in balancing flow-to-VNF (virtual network function) mappings to avoid SFC-loading problems.

II. Virtualized Services Model

The physical network substrate is composed of two data centers, connected to a common border router. We model the physical network substrate as a directed graph $G^P(N, E)$ composed of a set of physical nodes and links. While the nodes host the VNFs that are then networked to form a service function chain, the virtual network traffic is carried over the links in the physical network substrate. The physical nodes and links are hosted on commercial off-the-shelf (COTS) hardware and have processing and transfer capacity constraints. The physical network substrate forms the NFV infrastructure (NFVI). The virtual network substrate is also modeled as a directed graph $G^V(V, E_v)$ and consists of VNFs and their associated links.

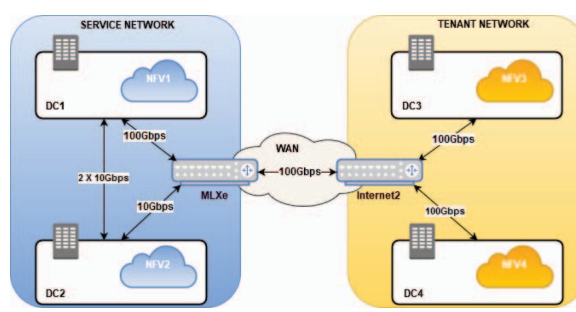


Fig. 1: Virtualized Services Network Scenario.

TABLE I: NOTATIONS USED IN THE ILP FORMULATION

Parameters	Description
$B_{i,j}$	The bandwidth of the physical link $(i,j) \in E$.
B_v	The bandwidth capacity of a virtual link $e_v \in E_v$.
B_s	The bandwidth capacity of the SFC $s \in S$.
K_u	The maximum allowed VNF instances u of type $v \in V$.
K_v	The maximum allowed VNFs on the network.
K_s	The maximum allowed SFCs on the network.
b_f	Bandwidth demand of the flow $f \in F$.

The network scenario used by our optimization model is as shown in Figure 1. Our work models the service function chain mapping to data centers across multiple campus networks. The SFCs are set up only in the data centers 1 and 2. Each data center generates two types of traffic: i) Traffic from experimental science projects such as CMS [1] and LIGO [2], and ii) Commodity Internet traffic from campus network users. Data centers DC1 and DC2 are managed by the same network and connect to the wide area network (WAN) through the Brocade MLXe border router as shown in Figure 1. This network hosts the NFV infrastructure (NFVI) and provides the service functions for processing both traffic types. Two other data centers connect to this network over the Internet2 backbone.

III. SFC MAPPING PROBLEM

A network operator's strategy for service provisioning across data centers should provide a convenient mechanism for i) the placement and deployment of VNFs, ii) decisions regarding the mapping of SFCs to VNF instances and iii) how the ingress traffic flows are routed to the appropriate SFCs. In this section, we present a model for mapping SFCs for optimizing their placement cost across multiple data centers. The model can be used to make *flow-to-SFC* mapping decisions, while minimizing the flow processing workloads on the NFVI.

A. Service Function Chaining Model

In this section, we present a formulation of the SFC mapping problem. It is defined as follows:

Problem Definition 1: Given a physical network substrate graph $G^P(N,E)$, find the optimal placement of VNFs on a service function chain to minimize the flow processing costs.

We denote by S the set of all SFCs in the network. Each SFC s comprises of a set ($V' \subseteq V$) of VNFs "chained together" in some predefined order through virtual links. The SFC is also connected to a set of ingress/egress endpoints (mapped to physical nodes) that are responsible for forwarding traffic through the chain. Service classifiers are defined at the ingress node and are tasked with mapping incoming flows to appropriate SFCs. Each service chain request is characterized by a flow request and a bandwidth specification. The SFC is modeled as a graph $G^S(V^s, E^s_v)$, where $V^s = \{v(in), v(s) \subset V, v(out)\}$ and $E^s_v = \big((v(in), v_1), (v_1, v_2), \dots, (v_n, v(out))\big)$, with $E^s_v \subseteq E_v$. To differentiate between endpoint nodes and function nodes on the service chain, we denote the set of endpoint nodes as $V^s_{end} = \{v(in), v(out)\}$, and the set of function nodes as $V^s_{fn} = \{v'\} = V^s - V^s_{end}$.

B. Problem Formulation: SFC-LP

1) Decision Variables: We define the following binary decision variables for the SFC mapping problem.

$$\beta_{v,n} = \begin{cases} 1, & \text{if VNF } v \text{ is placed on node } n \in N. \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

$$\beta^{v'}_{s,n} = \begin{cases} 1, & \text{if VNF } v' \text{ mapped to SFC } s \text{ is on } n \in N. \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

$$\lambda^{e_v}_{s,e} = \begin{cases} 1, & \text{if virtual link } e_v \text{ is placed on SFC } s. \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

$$\delta^{f}_{s,n} = \begin{cases} 1, & \text{if flow } f \text{ is mapped to } s \text{ on node } n \in N. \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

2) Objective: To minimize the SFC flow processing costs.

$$\text{Minimize} \sum_{n\in N}\sum_{v\in V}\beta_{v,n} + \sum_{e\in E}\sum_{s\in S}\sum_{e_v\in E^s_v}\lambda^{e_v}_{s,e} \quad (5)$$

The objective in (5) is subject to the constraints below.

3) Constraints for SFC placement: Constraint (6) ensures that the VNF must be available before we can place it on the SFC. Constraint (7) makes sure that only one VNF per SFC is mapped to one physical node. (8) ensures that all VNF instances in the network do not exceed a defined maximum.

$$\beta^{v'}_{s,n} \le \beta_{v,n}, \quad \forall n \in N, s \in S, v \in V, v' \in V^s \quad (6)$$

$$\sum_{n\in N}\beta^{v'}_{s,n} \le 1, \quad \forall s \in S, v' \in V^s \quad (7)$$

$$\sum_{n\in N}\beta_{v,n} \le K_v, \quad \forall v \in V \quad (8)$$

$$\sum_{n\in N}\beta^{v(in)}_{s,n} \le 1, \text{ and } \sum_{n\in N}\beta^{v(out)}_{s,n} \le 1, \quad (9)$$
$$\forall s \in S, v(in), v(out) \in V^s_{end}$$

Further, constraint (9) ensures that we only have one ingress and one egress endpoint per service chain.

4) Constraints for Resource Capacity:

$$\sum_{v\in V}\beta_{v,n}\cdot C_v \le C_n, \quad \forall n \in N \quad (10)$$

$$\sum_{v\in V}\beta^{v'}_{s,n}\cdot B_s \le \beta_{v,n}\cdot B_v, \quad \forall n \in N. \forall v' \in V^s \quad (11)$$

$$\sum_{s\in S}\sum_{e^s_v\in E^s_v}\lambda^{e_v}_{s,e}\cdot B_s \le B_{i,j}, \quad \forall e \in E \quad (12)$$

The resources requested by the VNFs cannot exceed the physical node capacity that they are mapped to, as presented in constraint (10). Further, in constraint (11), we ensure that the VNFs have sufficient traffic processing capabilities to handle the traffic from all SFC that they are mapped to, and finally, constraint (12) ensures that the bandwidth capacity of the physical link is sufficient to handle the traffic from all SFCs.

5) Constraints for Flow-to-SFC mapping: In (13), the number of flows that are mapped to SFCs are constrained to not exceed the number of service chains. The bandwidth capacity of service chain is lower-bounded by B_v as in constraint (14).

$$\beta^{v'}_{s,n} \le \sum_{f\in F}\delta^{f}_{s,n}, \quad \forall n \in N, v' \in V^s, 1 \le s \le K_s \quad (13)$$

$$\sum_{n\in N}\sum_{f\in F}\delta^{f}_{s,n}\cdot b_f \le B_v, \qquad \forall 1 \le u \le K_u, v \in V \quad (14)$$

6) Constraints for Flow Conservation: Constraints (15,16) represent the flow conservation constraints for the SFCs. Also, α_f is a binary decision variable that is set to 1 for admitted flows and 0 otherwise.

$$\sum_{e \in E} \lambda_{s,e_1}^{e_v} - \sum_{e \in E} \lambda_{s,e_2}^{e_v} = \sum_{f \in F} \delta_{s,n}^{v(in),f} - \sum_{f \in F} \delta_{s,n}^{v(out),f}, \quad (15)$$

where $e_1 = (i,j)$ and $e_2 = (j,i)$

$$\sum_{e \in E} \lambda_{s,e_1}^{e_v} - \sum_{e \in E} \lambda_{s,e_2}^{e_v} = \begin{cases} \alpha_f, & \text{if } e_1 = v(in) \\ -\alpha_f, & \text{if } e_2 = v(out) \\ 0, & \text{otherwise} \end{cases} \quad (16)$$

The cost function in (17) is used to estimate the total processing cost of a single flow by a service function chain mapped to a data center as allocated by the SFC-ILP algorithm.

$$\mathcal{C}_{\mathcal{SFC}} = \mathcal{C}_f + \mathcal{C}_v \quad (17)$$

$$\text{where, } \mathcal{C}_v = \mathcal{C}_B \cdot \omega_B + \mathcal{C}_l \cdot \omega_l + \mathcal{C}_h \cdot \omega_h \quad (18)$$

The total flow processing cost $\mathcal{C}_{\mathcal{SFC}}$ is a function of fixed and variable costs. The fixed cost $\mathcal{C}_f$ depends on the number of VNFs in the SFC, and the variable cost $\mathcal{C}_v$ depends on the flows' impact on bandwith, latency and associated host costs (i.e. $\mathcal{C}_B, \mathcal{C}_l, \mathcal{C}_h$) respectively. Each component of the variable cost $\mathcal{C}_v$ has an associated weight function (i.e. ω_B, ω_l, and ω_h) to account for variations in link bandwidth, link latency and the host resource capacity of each data center. The weights were assigned by normalizing the bandwidth, delay and CPU usage based on the corresponding measurements across each of the four data centers. Figure 2a shows the number of experimental science transfers at data center 1 averaged over a period of one month categorized by user type. We assume that a total of 15000 flows (both experimental science and commodity Internet) are processed by a data center every day. The available bandwidth between data centers and the corresponding round-trip time (RTT) measurements are as shown in Figures 2b and 2c.

IV. Application-Aware Flow Reduction

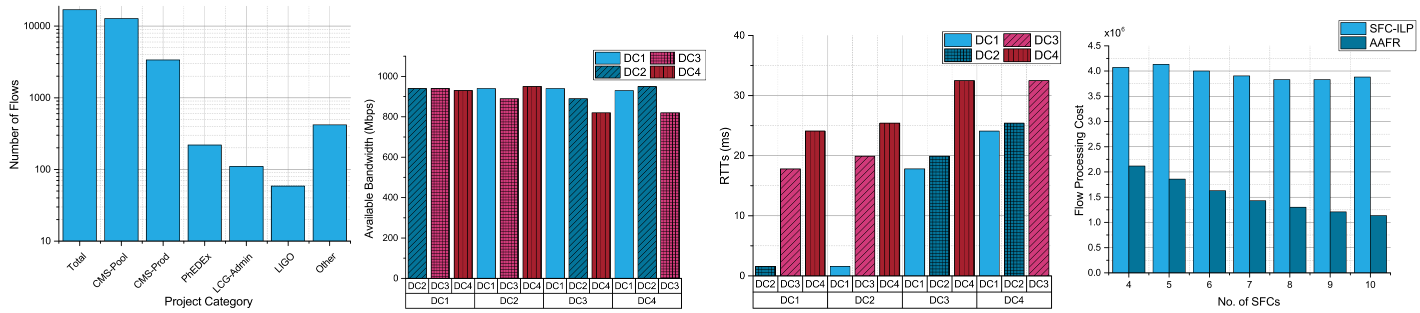
Application-awareness is achieved using the SDN-managed Network Architecture for GridFTP transfers (SNAG) proposed in [3]. We define application-awareness as the exchange of application-layer metadata with the network-layer, thereby facilitating collaboration between the two layers. SNAG exposes an application program interface (API) and communicates the application-layer metadata associated with the underlying connections over a representational state transfer (REST) interface. The traffic classification information in Figure 2a cannot be obtained *without* application-awareness since GridFTP [4] protocol uses encrypted TCP sessions between end-points for data movement. In our work, we use SNAG to accurately identify and to differentiate experimental science transfers from the commodity Internet traffic. Thus, application-awareness results in reduced flow processing workload for the SFCs as pre-classified experimental science traffic are not subject to flow processing by the SFCs in the NFVI. Subjecting only commodity Internet traffic to SFC processing is justified since experimental science workflows incorporate multiple security components to establish user/service identity. These components (such as X.509 PKI and proxy certificates) are used to protect communication between end-points and determine user credentials and authorization for specific actions. Thus, application-awareness reduces the flow-processing workload by subjecting only commodity Internet traffic to SFC processing. Therefore, the flow-processing cost of the AAFR algorithm accounts only for flow-switching and associated host costs as no resources are allocated for processing end-to-end experimental science traffic. However, commodity Internet traffic flows incur the same cost as before. Thus, AAFR reduces the flow processing costs by mapping SFCs only to commodity Internet traffic.

Algorithm 1 The AAFR Mapping Algorithm

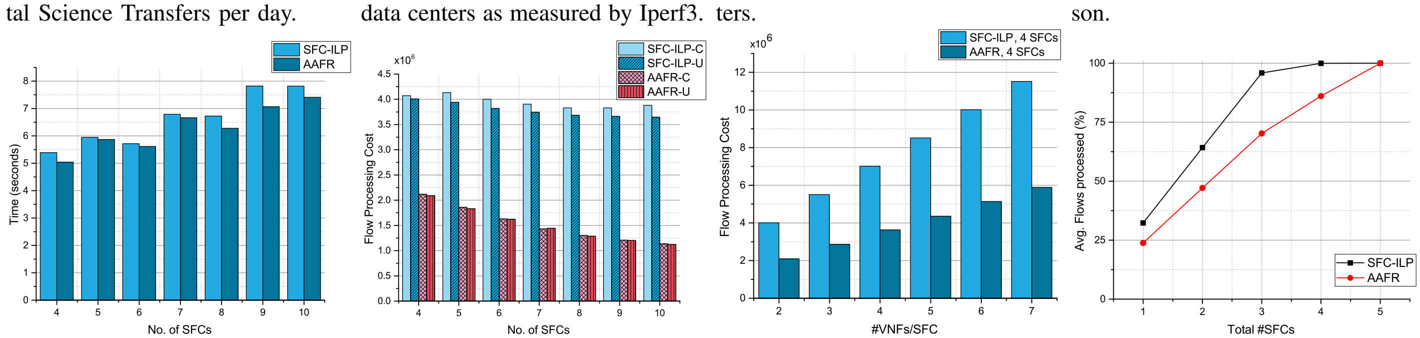
for all $\{f_i\} \subset F$ **do**
  Construct a flow set F_e
  if *SNAG*($f_i ==$ TRUE) **then**
    $F_e \cup \{f_i\}$ and $F' \leftarrow F - F_e$
  end if
end for
for all $\{f_i\} \subseteq F'$ **do**
  Solve the (SFC-ILP) on f_i and find the mapping of $G^S(V^s, E_v^s)$ to $G^p(N, E)$.
  Update C_n' and C_v' to reflect the new capacities.
end for
Compute the mappings of SFC $s \in S, \forall n \in N$ to satisfy the updated resource/capacity constraints C_n' and C_v'.
Obtain the *flow-to-SFC* allocations $\forall f_i \in F'$ and assign a subset of the flows $\{f_i\} \rightarrow s_i, \forall s_i \in S$.
Setup service function chains $\sum_{j=1}^{|S_i|} s_i, \forall s_i \in S, \forall i \in$ $\{\text{DCs}\}$ with a total of $S_{i,j}$ SFCs in data center i.

V. Experimental Study

We solve the optimization problems described in Section III using IBM CPLEX 12.7.1. We create a testbed data center network and use its traffic and resource characteristics to provide inputs to the optimization model. We then use the SFC mapping solutions provided by the models for SFC placement and associated flow processing. We set up four data centers as shown in Figure 1. Each data center is set up on a high-performance server hosting an OpenStack Ocata cloud and SFC extensions for NFV management. Data centers DC1, DC3 and DC4 run one compute and two controller nodes, with 8 cores and 128GB RAM on each node. DC2 is hosted on a shared private cloud with 80 VCPUs and 448GB of RAM. DC1 and DC2 communicate with each other over the WAN through the same border gateway router, and connect to the other data centers over the Internet2 backbone. Of the 15000 flows processed per day on average, about $70\% - 80\%$ flows constitute experimental science transfers with the remaining $20\% - 30\%$ forming the commodity Internet traffic. Each node in the physical network substrate is assumed to have a fixed resource capacity aggregate that is shared equally by all of the SFCs set up on that data center. The total number of VNFs per



(a) Average number of Experimental Science Transfers per day.
(b) Available Bandwidth between data centers as measured by Iperf3.
(c) Average RTT between data centers.
(d) Flow processing cost comparison.



(e) Execution time performance comparison.
(f) Capacitated vs. Uncapacitated SFC flow processing cost.
(g) Impact of #VNFs per SFC on flow processing cost.
(h) Cumulative average #flows processed by SFCs.

Fig. 2: Experimental setup parameters for different data centers and performance evaluation.

SFC is assumed to be constant for all SFCs in the network across all data centers. Unless otherwise specified, the flow processing capacity is limited to the total number of flows (F) and shared equally between each SFC (i.e. F/n flows/SFC for a total of n SFCs) in the network. A fixed per-flow processing cost associated with the SFC setup on each data center is added to the cost computation on each run. The NFVI for hosting SFCs are created only on data centers DC1 and DC2 (*service networks*), with data centers DC3 and DC4 forming the *tenant networks*. Thus, flows that are both internal and external to the *service networks* are processed by the DC1 and DC2.

Figure 2d shows the flow processing cost performance of the two algorithms for increasing number of SFCs in the *service network*. The number of SFCs in the *service network* is varied between 4 and 10 per data center and the flow processing cost is computed for both algorithms. We show that costs reduce with increasing number of SFCs in both cases. Increasing the number of SFCs in a data center reduces flow queuing and facilitates faster parallel processing leading to lowered costs. However, since SFC-ILP processes a larger number of flows it shows lower gains (about 4.9%) from increasing the number of SFCs compared to AAFR which shows gains of about 47%. In comparison to SFC-ILP, AAFR shows reduced cost gains between $47\% - 70\%$ for the same number of SFCs for both algorithms. We present the execution time performance for both algorithms in Figure 2e. Both algorithms perform comparably, with AAFR performing about 5% faster than SFC-ILP.

We present the flow processing cost for the capacitated and uncapacitated versions of both algorithms in Figure 2f. For the capacitated case, we limit the flow processing capacity of each SFC equally to F/n for F flows and n SFCs in the network. This results in a SFC mapping problem that is similar to the capacitated facility location problem wherein each facility has a fixed processing capacity. We refer to this as the *capacitated-SFC* mapping problem (SFC-ILP-C and AAFR-C). We compare the above to an *uncapacitated-SFC* mapping problem (SFC-ILP-U and AAFR-U). Our evaluations show limited gains when each algorithm is compared to its uncapacitated counterpart i.e. SFC-ILP-C vs. SFC-ILP-U and AAFR-C vs. AAFR-U. The cost-gains are about 0.6% and 4.1% for the SFC-ILP and the AAFR algorithms respectively. For the AAFR algorithm, the 4.1% cost-gain is in addition to the $47\% - 70\%$ gains described before when per-SFC processing capacity limits are increased to large values. Thus, increasing the SFC flow processing capacity to an arbitrarily large value does not lower flow processing costs.

The impact of the number of VNFs per SFC on flow processing cost are as shown in Figure 2g. The SFC flow processing cost increases monotonically with a corresponding increase in the number of VNFs per SFC. The number of VNFs for each service chain is varied between 2 and 7 and we see that the cost increases by about 18.7% on average per additional VNF per SFC for both algorithms.

In Figure 2h, we present the flow distribution across SFCs in each data center for both algorithms. This evaluation is an extension of the *uncapacitated-SFC* mapping problem described above. The evaluation is presented for the $\#SFC = 4$ case for each data center in the *service network*. The result in Figure 2h shows the cumulative average number of flows placed in each

SFC. We see that about 95.9% of the flows were mapped to the first three SFCs in the case of SFC-ILP, whereas only 70.25% of flows were mapped in the case of AAFR. Therefore, the majority of the flow processing load falls on roughly two-thirds of the SFCs in the network when SFC-ILP is used. Thus, AAFR is better at load balancing and mapping flows to SFCs in each data center due to reduced flow-processing workloads compared to SFC-ILP. Thus, our evaluations show that the AAFR algorithm effectively reduces flow processing workloads across data centers by using application-aware flow classification.

VI. Related Work

VNF placement problems have been discussed widely in the literature. Examples include [5]–[7]. However, such works focus on optimizing VNF placement rather than reducing flow processing cost per SFC. SFC placement/embedding problems are discussed in [8]–[12]. The work in [8] looks at how to optimize the deployment of SFCs for new users while balancing and readjusting the existing users' SFC to minimize deployment costs. An ILP formulation and a heuristic algorithm for QoS guaranteed service function chain placement across multiple clouds is proposed in [9]. A service function selection algorithm is proposed in [10] to balance a service function's path distance and its load. The proposed algorithm considers load, latency and QoS class constraints to select service function paths during the initial deployment of the SFC across multiple data centers. Joint topology design and SFC mapping for Telco clouds is explored in [11] with the objective of minimizing bandwidth consumption. The proposed method uses feedback from critical sub-topology mappings to optimize SFC mapping. Although deep packet inspection (DPI) can be employed to achieve application awareness, it is not applicable to encrypted transports. Therefore, we limit our application-awareness discussion to works that implement seamless application metadata exchange (without resorting to DPI) between application- and network-layers. Such works are limited to [13]–[15]. SNAG [3] introduced application-awareness for data-intensive science and has been used to create novel NFV-based approaches to securing scientific data transfers [16]. In this work, we look at optimizing service chain mapping across multiple data centers. Our work focuses on optimizing SFC mappings and using application-awareness to reduce SFC flow processing workloads.

VII. Conclusions

In this paper, we formulate the SFC mapping problem for multi-data center network topology and present an integer linear programming formulation. We propose a novel application-aware flow reduction (AAFR) algorithm which significantly reduces the flow processing costs across multiple data centers. While SFC-ILP provides the optimal SFC mapping, the proposed AAFR algorithm simplifies the SFC-ILP by using a modified cost function to reduce the flow-processing workloads for the mapping in the *service network*. We evaluate the performance of our AAFR algorithm and quantify the impacts of the number of SFCs and the SFC length on mapping cost, compare capacitated/uncapacitated cost gains, and finally investigate balancing flow-to-SFC mappings across data centers. Extensive performance evaluations show that our proposed AAFR algorithm reduces flow-processing costs by 70% for the capacitated SFC mapping case over the SFC-ILP. Further, for AAFR-U, our algorithm provides an additional 4.1% cost-gain over the AAFR-C case. We also demonstrate that our proposed AAFR algorithm is better at balancing flow-to-SFC mappings due to reduced flow-processing workloads.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant Numbers OAC-1541442. This work was completed using the Holland Computing Center of the University of Nebraska, which receives support from the Nebraska Research Initiative.

References

[1] S. Chatrchyan, G. Hmayakyan, V. Khachatryan, A. M. Sirunyan *et al.*, "The CMS experiment at the CERN LHC," *Journal of Instrumentation*, vol. 3, no. 08, p. S08004, 2008.

[2] B. P. Abbott, R. Abbott, R. Adhikari, P. Ajith *et al.*, "LIGO: the Laser Interferometer Gravitational-Wave Observatory," *Reports on Progress in Physics*, vol. 72, no. 7, p. 076901, 2009.

[3] D. N. Anantha, Z. Zhang, B. Ramamurthy, B. Bockelman *et al.*, "SNAG: SDN-managed Network Architecture for GridFTP Transfers," in *INDIS '16*, SLC, Utah, November 2016.

[4] W. Allcock, J. Bresnahan, R. Kettimuthu, and M. Link, "The Globus Striped GridFTP Framework and Server," in *Supercomputing, 2005. Proc. ACM/IEEE SC 2005 Conference*, Nov 2005, pp. 54–54.

[5] H. Moens and F. D. Turck, "VNF-P: A model for efficient placement of virtualized network functions," in *10th International Conference on Network and Service Management (CNSM) and Workshop*, Nov 2014, pp. 418–423.

[6] B. Martini, F. Paganelli, P. Cappanera, S. Turchi *et al.*, "Latency-aware composition of Virtual Functions in 5G," in *2015 IEEE Conference on Network Softwarization (NetSoft)*, April 2015, pp. 1–6.

[7] A. Hirwe and K. Kataoka, "LightChain: A lightweight optimisation of VNF placement for service chaining in NFV," in *2016 IEEE NetSoft Conference and Workshops (NetSoft)*, June 2016, pp. 33–37.

[8] J. Liu, W. Lu, F. Zhou, P. Lu *et al.*, "On Dynamic Service Function Chain Deployment and Readjustment," *IEEE Transactions on Network and Service Management*, vol. PP, no. 99, pp. 1–1, 2017.

[9] H. Chen, S. Xu, X. Wang, Y. Zhao *et al.*, "Towards optimal outsourcing of service function chain across multiple clouds," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–7.

[10] A. M. Medhat, G. Carella, C. Lck, M. I. Corici *et al.*, "Near optimal service function path instantiation in a multi-datacenter environment," in *Conference on Network and Service Management*, 2015, pp. 336–341.

[11] Z. Ye, X. Cao, J. Wang, H. Yu *et al.*, "Joint topology design and mapping of service function chains for efficient, scalable, and reliable network functions virtualization," *IEEE Network*, vol. 30, no. 3, pp. 81–87, 2016.

[12] L. Wang, Z. Lu, X. Wen, R. Knopp *et al.*, "Joint Optimization of Service Function Chaining and Resource Allocation in Network Function Virtualization," *IEEE Access*, vol. 4, pp. 8084–8094, 2016.

[13] Z. A. Qazi, J. Lee, T. Jin, G. Bellala *et al.*, "Application-awareness in SDN," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 487–488, Aug. 2013.

[14] S. Zhao, A. Sydney, and D. Medhi, "Building Application-Aware Network Environments Using SDN for Optimizing Hadoop Applications," in *2016 ACM SIGCOMM*. NY, USA: ACM, 2016, pp. 583–584.

[15] B. Wang, J. Su, L. Chen, J. Deng *et al.*, "EffiEye: Application-aware Large Flow Detection in Data Center," in *17th IEEE/ACM CCGrid '17*. NJ, USA: IEEE Press, 2017, pp. 794–796.

[16] D. N. Anantha and B. Ramamurthy, "ScienceSDS: A Novel Software Defined Security Framework for Large-scale Data-intensive Science (Invited Paper)," in *Proc. ACM SDN-NFVSec '17*. NY, USA: ACM, 2017, pp. 13–18.