Optimization Model and Algorithm for Dynamic Service-Aware Traffic Steering in Network Functions Virtualization

Thi-Thuy-Lien Nguyen
VNU-University of Engineering and Technology
Hanoi National University of Education
Hanoi, Vietnam
lienntt@hnue.edu.vn

Tuan-Minh Pham
Thuyloi University
Hanoi, Vietnam
minhpt@tlu.edu.vn

***Abstract*—Network Functions Virtualization (NFV) has emerged as a paradigm for efficient, flexible and agile network function provisioning. In such NFV-based networks, ensuring network performance and cost efficiency is an important challenge to tackle when network traffic is steered through a chain of virtual network functions (VNF). In this work, we consider the dynamics of traffic demand in different time periods, and multipath routing for minimizing the routing cost in NFV. We focus primarily on optimization models and algorithms for finding a traffic steering solution that effectively splits a demand volume into several flows and selects appropriate links and nodes for these flows. We formulate the problem as a mixed linear integer programming model for obtaining an optimal solution taking into account the dynamics of service demand, multipath routing and service function chaining. For the large scale problem, we propose a heuristic algorithm to find an approximation solution. Particularly, our proposed model and algorithm allows a controller to update a link weight system for effectively steering traffic demand to appropriate nodes in a NFV infrastructure. The evaluation results show that our approach to traffic steering significantly improves a number of major performance metrics including the routing cost, the maximum link utilization, and the accepted demands. In addition, the approximation solution is very close to the optimal solution.**

***Index Terms*—Network Functions Virtualization, traffic steering, integer programming, dynamic service demand, service function chaining**

I. Introduction

Today's network infrastructures and services are built almost on specialized appliances. To reduce capital and operational expenditures, network operators have been exploring network softwarization and cloudification to build efficient, flexible and agile networks, and to have them offered as a cloud service. Network Functions Virtualization (NFV) has emerged as a new network architecture concept where the network functions can be deployed on the commodity server as software, and chained together to create a communication service. NFV enables network operators to rapidly create, destruct on-demand network services in a flexible way. In NFV, data traffic is processed by a service function chain (SFC) composed of several network functions (e.g., WAN accelerators, Intrusion detection system (IDS), firewall) in a given sequence [1]. As a virtual network function (VNF) is located dispersedly through a NFV infrastructure (NFVI), it is challenging to steer network traffic through a SFC while maintaining high network performance and cost efficiency, especially in the dynamics of service demand in different time periods.

In this paper, we aim at finding a traffic steering solution that minimizes the routing cost by effectively splitting a demand volume into multiple flows and routing these flows through appropriate paths. Several practical factors are taken into account. Particularly, the traffic volume of a service demand is different in successive time periods. We focus on optimizing the link weight system over multiple time periods in a traffic steering solution when considering the dynamics of service demand. We also consider the Equal-cost Multipath (ECMP) routing protocol and NFV characteristics including SFC and resource constraints at NFV infrastructure. ECMP is a common multipath routing scheme which splits equally the total traffic passing through one network node for the minimum-cost paths of a source-destination pair. We investigate both an optimization model and approximation algorithm for the traffic steering problem in NFV that takes into account the dynamic service demand, ECMP, and SFC.

The major contributions of this paper are as follows:

- We formulate the traffic steering problem as a mixed linear integer programming model (MILP) considering several practical aspects including the dynamics of traffic demands in different time periods, the ECMP routing strategy, and NFV characteristics under resource constraints at NFVI nodes and links.
- We propose an efficient heuristic algorithm for the large scale problem. The algorithm produces a link metric vector and a traffic splitting solution regarding the dynamics of traffic demand and available system resources for minimizing the total network routing cost.
- We evaluate our proposed model and algorithm in two real network datasets. We compare our solution considering multiple time periods with the non-period scheme in three important performance metrics including the routing cost, the number of accepted demands, and the maximum

link utilization. The evaluation results show that our approach to traffic steering performs better than a traffic steering scheme without taking into account the dynamics of service demand. Moreover, the results obtained by the heuristic algorithm are very close to the optimal solution.

The rest of this paper is organized as follows. We review previous researches in Section II. We formally state the traffic steering problem under our consideration in Section III. The heuristic algorithm is described in Section IV for solving the large scale problem. Section V presents evaluations of our solution. Conclusions are stated in Section VI.

II. Related Work

NFV performance has been being investigated in several related aspects such as VNF placement, SFC routing, resource management and orchestration. In [2], authors present a scalable SFC orchestrator that can deploy VNF chains and orchestrate the runtime phase by rerouting the flow to an alternative path when a certain instance of service function is overload. Eramo et al. propose three algorithms capable of addressing NFVI placement, SFC routing and VNF Instance migration in response to workload changes [3]. Some other platforms are also developed to improve packet processing on standardized servers [4], [5]. The NFVnice framework is proposed to schedule resource fairly, efficiently and dynamically as well as to improve NF performance (throughput and loss) on NFV platforms [6]. NFV performance improvement is also investigated by enabling network function parallelism [7]. In [8], the authors present an adaptive multipath routing scheme for improving the NFV performance, in which the different traffic volumes of demands in multiple time periods have not been taken into account, and an optimization model has not been investigated. More research results about the NFV performance are presented in [9]–[12].

However, none of these studies has focused on traffic steering taking into account the dynamics of service demand, multipath routing and fundamental characteristics of NFV, while these practical factors have a major impact on the NFV performance. It is our motivation to study a traffic steering solution for NFV considering the dynamics of traffic demand in different time periods, ECMP multipath routing, and SFC. Specifically, our work focuses on the difference of workload in different time periods to minimize the total network routing cost. Furthermore, we provide an evaluation of important performance metrics of an NFV system in real network datasets for a traffic steering solution in scenarios of dynamic and non-dynamic service demands.

III. System Description and MILP Model

We study a NFV system that provides network functions as a service. The system is composed of three primary components: NFVI, a set of VNFs and NFV MANO [13]. In NFVI, the software implementation of a network function is referred to as a VNF. A service demand in NFV is a request from users for a network service that may include several VNFs. To serve a service demand, NFV MANO decides the location required to deploy a VNF and how to route its traffic through a given sequence of VNFs. As the data traffic of demands may change in different time periods because of user requirements, we consider the variation of traffic demand volume and cost structure in multiple time periods for minimizing the total routing cost. We aim to find the optimal link metric system over multiple time periods under constraints on NFVI resources and the requirements of the service demand. We shall refer to the problem under our consideration as the multi-period case or the dynamic case interchangeably.

TABLE I: Summary of notations

Sets	
D	Service demands
E	Links on NFVI
V	Nodes on NFVI
F	VNFs available on NFVI
P	Possible paths on NFVI
P_d	Possible paths for demand d
T	Time periods
Demand parameters	
h_{dt}	The traffic volume of $d \in D$ at time period $t \in T$
s_d	The source of demand d
t_d	The destination of demand d
F_d	The SFC of demand d, $F_d \subset F$
Network parameters	
$C_{1,e}$	Bandwidth capacity of link $e \in E$
C_{et}	Network routing cost of link $e \in E$ at time period $t \in T$
$C_{2,v}$	Processing capacity of node $v \in V$
M_z	The maximum link capacity
k_{vdit}	A parameter indicates that node v supports the ith VNF of demand d at time period t if $K_{vdit} = 1$, otherwise $K_{vdit} = 0$
Binary variables	
b_{epdt}	A binary variable indicates that flow p of demand d uses link e at time period t if $b_{epdt} = 1$, otherwise $b_{epdt} = 0$
u_{ev}	A binary variable indicates that link e is on a minimum-cost path to node v if $u_{ev} = 1$, otherwise $u_{ev} = 0$
Continuous variables	
x_{epdt}	The traffic rate on link e of a data flow p of demand d at time period t
l_{uv}	A non-negative integer variable indicates the smallest length of the paths from node u to node v
y_{et}	Total of traffic data flows going through link e at time period $t \in T$
g_{uvt}	A non-negative continuous variable whose value is traffic volume assigned to outgoing links of node v that belongs to the minimum-cost paths from node u to node v at time period t
Metric vector variables	
w	A link weight system on NFVI, $\mathrm{w} = (w_e : e \in E)$
x	A traffic splitting vector for all demands at all time periods, $\mathrm{x} = \left(x_{epdt} : e \in E, p \in P, d \in D, t \in T\right)$

We model NFVI as a directed graph $G = (V, E)$ composed of a set of n virtual nodes V, and a set of k directed links E. Let $T = \{t_i | i = 1, 2, ...r\}$ be a set of r time periods. The traffic volume of a service demand can change dynamically in different time periods. A node v represents a commodity hardware device where one or more VNF can be instantiated flexibly to serve demands. A node has a limited capacity for instantiating VNFs. We use $C_{2,v}$ to denote the processing capacity of node $v \in V$. Each link $e \in E$ is a physical connection between the starting node i_e and the terminating

node j_e. Let $C_{1,e}$ and C_{et} denote the bandwidth capacity and routing cost for each data traffic unit at time period t of link $e \in E$, respectively. Let $F = \{f_i | i = 1, 2, ...m\}$ be all VNFs available in the system. We use D to represent a set of l service demands requested by tenants. Each demand $d \in D$ is defined as a customer request that requires a SFC from a source s_d to a destination t_d with different traffic volumes h_{dt} at different time period t. Note that we do not put any constraint on a number of VNF instances, e.g., a multiple use of a single VNF in a SFC.

We define w $= (w_e : e \in E)$ to be the link weight system on NFVI. The system applies the ECMP routing scheme to decide a traffic splitting vector x(w) $= (x_{epdt} : e \in E, d \in D, p \in P_d, t \in T)$ where P_d is all available paths on NFVI of demand d, and x_{epdt} is the amount of data on link e at time period t of flow p of demand d according to the link metric vector w.

Let k_{vd_it} be a parameter indicates that node v can provide the ith VNF of demand d if $k_{vd_it} = 1$, otherwise $k_{vd_it} = 0$. Next, we assume that r_{vf} is the computing resources required to process function f with one unit of traffic rate at node v.

The objective aims at obtaining a link metric vector w and a solution of traffic splitting x(w) for minimizing the maximum total network routing cost of data flows over all time periods while satisfying all service demands as well as NFVI resource constraints. The system determines x(w) based on the ECMP routing strategy for each link metric vector w.

We use some following variables in our model:

- b_{epdt} is a binary variable indicates that flow p of demand d uses link e at time period t if $b_{epdt} = 1$, otherwise $b_{epdt} = 0$.
- g_{uvt} is a non-negative continuous variable whose value is traffic assigned to outgoing links of node v that belongs to the minimum-cost paths of a source-destination pair (u, v) at time period t. The cost of a routing path is the total weight of links on the path.
- l_{uv} is a non-negative integer variable that is the smallest length of all paths from node u to node v $(u \neq t)$.
- u_{ev} is a binary variable indicates that link e is on a minimum-cost path to node v if $u_{ev} = 1$, otherwise $u_{ev} = 0$.
- x_{epdt} is a non-negative continuous variable that denotes the traffic on link e of flow p of service demand d at time period t.
- y_{et} is the total traffic through link e at time period t.

Table I provides a summary of notations that we use to formulate the traffic steering problem in a NFV system.

Our objective is to find a solution having the minimal total routing cost. The objective function is given by

$$F = \max_{t \in T} \sum_e C_{et} y_{et}$$

where $y_{et} = \sum_{d,p} x_{epdt}$.

We now present the constraints of our model. First, to ensure that total traffic incoming a link equals to total traffic outgoing that link, we use constraint (1):

$$\sum_{p,e:i_e=v} x_{epdt} - \sum_{p,e:j_e=v} x_{epdt} = 0, \quad \forall d, \forall v, \forall t, v \neq s_d, v \neq t_d \tag{1}$$

Constraint (2) and (3) make sure that the total traffic outgoing from the source node of demand d and the total traffic incoming the destination node of demand d are the same as the traffic volume of demand d for each time period, respectively. It means that service demand d is served fully by the system.

$$\sum_{p,e:i_e=s_d} x_{epdt} = h_{dt}, \quad \forall d, \forall t \tag{2}$$

$$\sum_{p,e:i_e=t_d} x_{epdt} = h_{dt}, \quad \forall d, \forall t \tag{3}$$

Next, we need to ensure that the total traffic passing through a link is not over the link capacity. This constraint is expressed as follows:

$$\sum_{d,p} x_{epdt} \leq C_{1,e}, \quad \forall e, \forall t \tag{4}$$

In addition, it is essential to make sure that the VNF deployment does not violate the computing capacity of nodes. We use constraint (6) to present it where equation (5) computes resources required to process functions f with one unit of traffic at node v.

$$R_v(x, f) = x r_{vf} \tag{5}$$

$$\sum_{d,i} R_v(k_{vdit}. \sum_{p,e:j_e=v} x_{epdt}, F_{di}) \leq C_{2,v} \tag{6}$$

Each VNF in the service function chain of demand d must be deployed to one node, and any flow of demand d must go through a sequence of VNFs specified in the SFC of demand d. We represent these constraints as follows:

$$\sum_e x_{epdt}(k_{i_e d_i t} + k_{j_e d_i t}) > 0, \quad \forall d, i, t, h_{dt} \tag{7}$$

$$\sum_e x_{epdt} > 0, \quad \forall d, p, h_{dt} > 0 \tag{8}$$

$$0 \leq x_{epdt} \leq M_z b_{epdt}, \quad \forall d, p, e, t \tag{9}$$

Next, we formulate the flow constraint to ensure that the in-flow and out-flow of each node is equal where M_z is the maximum link capacity over all links.

$$x_{epdt} \geq \sum_{\{e':j_{e'}=i_e\}} x_{e'pdt} - M_z(1 - b_{epdt}), \quad \forall d, e, p, t \tag{10}$$

$$x_{epdt} \leq \sum_{\{e':j_{e'}=i_e\}} x_{e'pdt}, \quad \forall d, e, p, t \tag{11}$$

Finally, we use constraint (12), (13), (14) to represent the constraint on traffic splitting according to the ECMP routing scheme.

$$0 \leq g_{i_e vt} - \sum_{p,d:t_d=v} x_{epdt} \leq (1 - u_{ev}) \sum_{d:t_d=v} h_{dt}, \quad \forall t, v, e \tag{12}$$

$$\sum_{p} x_{epdt} \leq u_{et_d} h_{dt}, \quad \forall d, e, t \tag{13}$$

$$1 - u_{et_d} \leq l_{j_e t_d} + w_e - l_{i_e t_d} \leq (1 - u_{et_d}) M_z, \forall d, e \tag{14}$$

Our MILP formulation presented above can be effectively solved by a MILP solver such as CPLEX for a moderate number of service demands and network size. We propose a heuristic algorithm to find an approximation solution for the large scale problem in the next section.

IV. Heuristic Algorithm

We propose a resource allocation algorithm for traffic steering, called RAP, which finds a weight system vector and a routing solution for minimizing the total network routing cost in NFV. The heuristic includes two primary components: (i) optimizing link metric vector while considering the change of traffic volume of demands at different time periods as well as available NFVI resources, (ii) deciding a traffic splitting vector by applying the ECMP routing scheme based on service demands, the weight system vector and available system resources.

A. Optimizing the link metric vector

To find the optimal link metric vector, we use an approximation algorithm based on Simulated Annealing heuristic [14] with our proposal for neighborhood selection (Algorithm 1). Firstly, the algorithm initializes a metric vector by setting all link weights as 1. Then, the algorithm optimizes the link weight system in two loops. In the neighborhood selection (i.e., line 9), the algorithm finds links whose routing cost is largest, and increases these link weights by 1 for each time period. The algorithm decides a traffic splitting vector, and calculates the objective function $U(z)$ by using Algorithm 2 with the new obtained link metric vector z. The details of Algorithm 2 are described in the next section. The algorithm chooses the new link metric vector z if the value of $U(z)$ is not worse and it does not lead to a decrease of the minimum accepted demands. Otherwise, z will be chosen with probability $e^{-\Delta U/Q}$ to overcome a local optimization. Finally, we obtain the best link metric vector for updating the system.

B. Deciding the traffic splitting vector

With each link weight system on NFVI, Algorithm 2 computes the maximum total network routing cost and the minimum number of accepted demands over time periods after determining the traffic splitting vector. The first step in the algorithm is to find all minimum-cost paths for the source-destination pair of each demand. Then, the algorithm uses the ECMP routing scheme to decide how to split the traffic volume of each demand on these paths. The procedure loops over all demands for each time period.

Algorithm 1 Optimizing the link metric vector

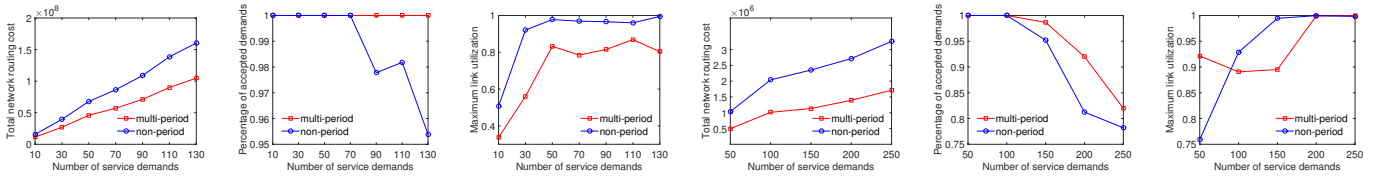
Input: *G, D, F, T*
Output: Traffic splitting solution x, link metric vector w

1: Initialize w
2: $\mathrm{w}_{best} \leftarrow$ w, $U_{best} \leftarrow \mathrm{U}_{\mathrm{w}_{best}}$, $A_{best} \leftarrow \mathrm{A}_{\mathrm{w}_{best}}$
3: $Q \leftarrow Q_o$
4: **while** $Q > 1$ **do**
5: $l \leftarrow 0$
6: **while** $l < L$ **do**
7: z $\leftarrow$ generate a neighbor vector
8: $\Delta U \leftarrow U_{\mathrm{z}}$ - U_{w}, $\Delta A \leftarrow A_{\mathrm{z}}$ - A_{w}
9: **if** $\Delta U \leq 0$ **and** $\Delta A \geq 0$ **then**
10: w $\leftarrow$ z
11: **if** $U_{\mathrm{z}} < U_{best}$ **then**
12: $\mathrm{w}_{best} \leftarrow$ z, $U_{best} \leftarrow U_{\mathrm{z}}$, $A_{best} \leftarrow A_{\mathrm{z}}$
13: **end if**
14: **else**
15: **if** random(0,1) $< e^{-\Delta U/Q}$ **then**
16: w $\leftarrow$ z
17: **end if**
18: **end if**
19: $l \leftarrow l + 1$
20: **end while**
21: reduce temperature Q
22: **end while**

Algorithm 2 Deciding the traffic splitting vector

Input: *G, D, F, T*, w
Output: Traffic splitting solution x,
minimum accepted demand *A*

1: **for all** time period $t \in T$ **do**
2: Sort decreasingly demands by these demand volumes
3: **for all** $d \in D$ **do**
4: $P_d \leftarrow$ all minimum-cost paths from s_d to t_d
5: x $\leftarrow$ split flow traffic for P_d according to ECMP
6: **if** has no overload links **then**
7: map VNFs of demand d with nodes of P_d
8: **end if**
9: **if** satisfy all constraints **then**
10: update resource constraints
11: **end if**
12: **end for**
13: $U \leftarrow$ calculate the maximum total network routing cost
14: $A \leftarrow$ calculate the minimum accepted demand
15: **end for**



(a) Routing cost with Internet2 (b) Accepted demands with Internet2 (c) Link utilization with Internet2 (d) Routing cost with Geant (e) Accepted demands with Geant (f) Link utilization with Geant

Fig. 1: Comparison between the multi-period solution and the non-period solution

V. Evaluation

A. Parameter setting

We use two datasets of real network traffic and network topologies in our evaluation. The first dataset is the Internet2 research network including a topology of 12 nodes, 15 links and traffic matrices with 130 demands [15]. In the Internet2 network, we consider three time periods. The demand volumes in the first time period are tracked from the real traffic. The others are generated randomly between the minimum and maximum traffic volume of all demands. The second dataset is the Geant dataset that contains a topology of 22 nodes, 36 links, 250 service demands with four time periods [15]. The traffic volumes of demands are recorded from the Geant network in the first time period and randomly generated in the others. We consider four VNFs available on NFVI. The computing capacity, the resource requirements of a VNF, and the SFC of each demand are randomly generated.

B. Experimental results

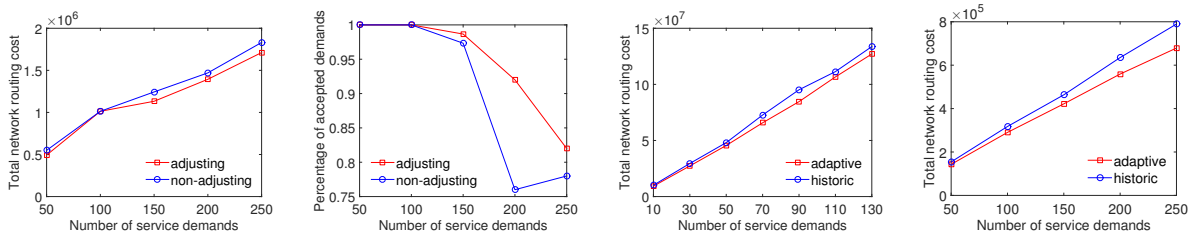
First, we evaluate the efficiency of our approach by comparing results achieved by the case of multiple time periods and the non-period case. We perform RAP in the same network scenarios and demands for two cases. In the non-period case, there is only one demand volume for each service demand and this value is the maximum demand volume over all time periods in the multi-period case. We compute three performance metrics including the total network routing cost, the maximum link utilization, and the minimum acceptance ratio of service demand.

We can observe that the multi-period solution outperforms the non-period solution in term of the three metrics. For the evaluation using the Internet2 dataset, Fig. 1a and Fig. 1c show that the multi-period solution can save at least 27% the total network routing cost and 10% the maximum link utilization as compared to the non-period solution. For the ability of satisfying demands, while the multi-period solution always serves 100% number of requested demands, the non-period solution serves less than 95.5% for 130 requested demands (Fig. 1b). For the evaluation using the Geant dataset, Fig. 1d shows that the total network routing cost of the multi-period solution is only approximately 50% that of the non-period solution. Similarly, the multi-period solution achieves a higher acceptance ratio comparing with the non-period solution (Fig. 1e). Especially, the benefits of considering the multiple time periods in a service demand over the non-period case are higher when a number of demands and network size are large. Fig. 1f shows that the multi-period solution obtains the maximum link utilization that is not as good as the non-period solution with 50 service demands because of the aggregation of traffic on a low-cost link of a minimum-cost path.

Second, we evaluate the performance of our approach when optimizing the link weight system over multiple time periods. The ECMP routing is independently repeated in each time period, but it uses different link weight systems in two cases. In the first case, i.e., the non-adjusting case, the algorithm ignores the step of optimizing the link metric vector and the weight of links is fixed as 1. In the second case, i.e., the adjusting case, the algorithm finds the optimal link weight system over multiple time periods. Fig. 2a and Fig. 2b show the results with the Geant dataset. We observe that the solution in the adjusting case is much better than that in the non-adjusting case in both the total network routing cost and the percentage of accepted demands.

Third, we compare the efficiency of our approach when using the historic link weight and the adaptive link weight for finding a NFV routing solution in each new time period. In the former, the algorithm computes the optimal link weight system based on the information of historic demands, and then it uses this link weight system for routing next demands. In the latter, the algorithm uses the information of demands in the current time period to compute again the optimal link weight vector and the routing solution. Fig. 2c and Fig. 2d illustrate that the total network routing cost achieved by using the historic link weight system is very close to the result obtained by re-running the algorithm for each new time period. In spite of a slightly better result, the computation of the optimal link weight for each time period is not practical because of the high disruption to the network routing [16].

Finally, we analyze the performance of our approach in comparing RAP and the optimal solution. We use the IBM ILOG CPLEX Optimizer to achieve the optimal solution of the MILP model for the small-scale problem (i.e., the number of service demands varies between 4 and 32). Fig. 3a and Fig. 3b provide a comparison between the total network routing cost obtained by RAP and the optimal solution with the Internet2 and Geant network. The total network routing costs achieved by two algorithms are almost equivalent in the Internet2 topology and very close in the Geant topology. Fig. 3c and Fig.



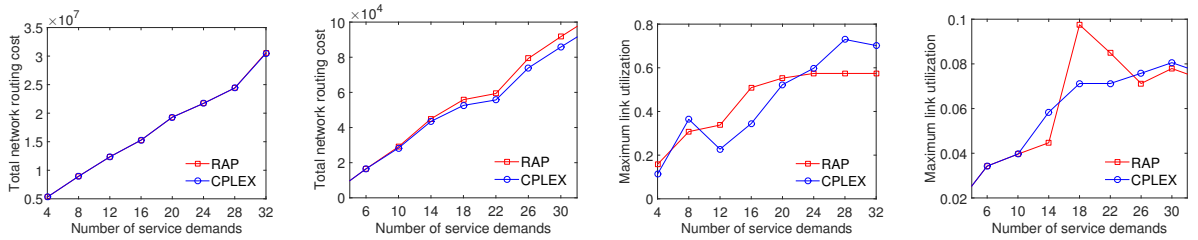
(a) Routing cost in the use of the adjusting and non-adjusting link weight systems with Geant

(b) Accepted demands in the use of the adjusting and non-adjusting link weight systems with Geant

(c) Routing cost in the use of the historic and adaptive link weight systems with Internet2

(d) Routing cost in the use of the historic and adaptive link weight systems with Geant

Fig. 2: Comparison among the use of the adjusting, non-adjusting, historic, and adaptive link weight systems



(a) Routing cost with Internet2

(b) Routing cost with Geant

(c) Link utilization with Internet2

(d) Link utilization with Geant

Fig. 3: Comparison between RAP and CPLEX

3d depict a comparison of the maximum link utilization when using the Internet2 and Geant networks. The results show that the maximum link utilization provided by RAP is a little bit higher as compared to the optimal solution.

VI. Conclusion

In this paper, we investigated the traffic steering problem considering the dynamics of traffic demands in different time periods and taking into account the ECMP routing strategy and SFC in NFV. We introduced a MILP model for obtaining the optimal solution of the problem. For the large scale problem, the RAP algorithm can find an approximate solution that is very close to the optimal solution. The evaluation results show that a traffic steering solution considering the dynamics of traffic demands can reduce approximately twice time the total network routing cost compared to the non-period case. Possible extensions of our work include the consideration of the forecast traffic and failure scenarios in the traffic steering problem.

References

[1] J. Halpern and C. Pignataro, *RFC 7665: Service Function Chaining (SFC) Architecture*, 2015.

[2] A. M. Medhat, G. A. Carella, M. Pauls, and T. Magedanz, "Orchestrating scalable service function chains in a nfv environment," in *Proc. IEEE NetSoft*, 2017, pp. 1–5.

[3] V. Eramo, E. Miucci, M. Ammar, and F. G. Lavacca, "An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures," *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2008–2025, 2017.

[4] J. Hwang, K. K. Ramakrishnan, and T. Wood, "Netvm: High performance and flexible networking using virtualization on commodity platforms," *IEEE Trans. Netw. Service Manag.*, vol. 12, no. 1, pp. 34–47, 2015.

[5] J. Martins, M. Ahmed, C. Raiciu, V. Olteanu, M. Honda, R. Bifulco, and F. Huici, "Clickos and the art of network function virtualization," in *Proc. USENIX Conference on Networked Systems Design and Implementation*, 2014, pp. 459–473.

[6] S. G. Kulkarni, W. Zhang, J. Hwang, S. Rajagopalan, K. K. Ramakrishnan, T. Wood, M. Arumaithurai, and X. Fu, "Nfvnice: Dynamic backpressure and scheduling for nfv service chains," in *Proc. ACM SIGCOMM*, 2017, pp. 71–84.

[7] C. Sun, J. Bi, Z. Zheng, H. Yu, and H. Hu, "Nfp: Enabling network function parallelism in nfv," in *Proc. ACM SIGCOMM*, 2017, pp. 43–56.

[8] T.-T.-L. Nguyen, T.-M. Pham, and H. T. T. Binh, "Adaptive multipath routing for network functions virtualization," in *Proc. SoICT*, 2016, pp. 222–228.

[9] T. M. Pham and L. M. Pham, "Load balancing using multipath routing in network functions virtualization," in *Proc. IEEE RIVF*, 2016, pp. 85–90.

[10] J. Elias, F. Martignon, S. Paris, and J. Wang, "Optimization models for congestion mitigation in virtual networks," in *Proc. IEEE ICNP*, 2014, pp. 471–476.

[11] R. Cohen, L. Lewin-Eytan, J. S. Naor, and D. Raz, "Near optimal placement of virtual network functions," in *Proc. IEEE INFOCOM*, 2015, pp. 1346–1354.

[12] T. M. Pham, T. T. L. Nguyen, S. Fdida, and H. T. T. Binh, "Online load balancing for network functions virtualization," in *Proc. IEEE ICC*, 2017, pp. 1–6.

[13] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. D. Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 2016.

[14] Hwang and Chii-Ruey, "Simulated annealing: Theory and applications," *Acta Applicandae Mathematica*, vol. 12, no. 1, pp. 108–111, 1988.

[15] Sndlib. http://sndlib.zib.de.

[16] J. Rexford, *Route Optimization in IP Networks*. Springer, 2006, pp. 679–700.