Optimal delay-aware service function chaining in NFV

Fatemeh Yaghoubpour
Department of Computer Engineering and Information Technology
Amirkabir University of Technology
Tehran, Iran
Email: f.yaghoubpour@aut.ac.ir

Bahador Bakhshi
Department of Computer Engineering and Information Technology
Amirkabir University of Technology
Tehran, Iran
Email: bbakhshi@aut.ac.ir

Abstract— **Network Function Virtualization (NFV) enables the networks to provide dynamic and agile services by decoupling the network functions from hardware. Resource allocation is one of the most important challenges in NFV-based networks to maximize the providers' profile while satisfying customer requirement. End-to-end delay is one of the requirements that have got little attention in the literature. In this paper, we formulate the VNF embedding problem subject to delay constraint as a MINLP problem. In this formulation, the objective is to maximize the provider's profit such that the constraints of the nodes' capacity (i.e., memory, CPU, and storage), the links' capacity (i.e., bandwidth), the end-to-end delay threshold, and required resources (i.e., memory, CPU, and storage) for each instance are satisfied. This formulation can be used to analyze the effect of system parameters on the objective. The problem is solved optimally by the SCIP optimization tool. The simulation results verify our proposed model and the solvers optimal solution.**

Keywords- Network Function Virtualization, Service chain, VNF embedding, Instance, End-to-end delay, Optimality

I. INTRODUCTION

In recent years, by an explosion of mobile devices, the requirements for more diverse and new services with high data rate have been increased. To provide the services, operators require dense deployments of network equipment and rapidly changing skills for managing this equipment. These requirements increase the Operating Expenses (OPEX) and Capital Expenses (CAPEX). To build more dynamic and service-aware networks with reducing OPEX and CAPEX, Network Function Virtualization (NFV) is proposed. The main idea of NFV is migrating network functions from dedicated hardware to software instances running on general purpose virtualized networking and computing infrastructures. Reducing CAPEX and OPEX, especially the power consumption, and increasing the speed of service provisioning are some of NFV's advantages [1].

The deployment of NFV faces several challenges, some of these challenges are, management and orchestration, energy efficiency, resource allocation, and security. Resource allocation in NFV-based networks as one of the main challenges of NFV consists of three stages, 1) VNFs - Chain composition: to compose the chains of Virtual Network Functions (VNFs) dynamically and deploy them on a set of physical network nodes to meet a service-level-agreement, 2) VNF - Forwarding Graph Embedding: to find where to allocate the VNFs in the network infrastructure in a suitable way considering a set of requested network services, and 3) VNFs – Scheduling: to perform scheduling of VNFs' execution in order to minimize the total execution time of the network services.

Some existing works considered the resource allocation in NFV-based networks [3-8]. Authors in [3] addressed the problem of minimizing the cost of the occupied nodes and links. In [4], the problem of minimizing the links' delay and processing time is addressed considering the end-to-end delay as a summation of processing and propagation delay. In that paper, the queuing delay is ignored. Authors in [5] proposed a model to minimize the number of embedded VNFs into the infrastructure considering the processing and propagation delay. In [6], the minimization problem of occupied substrate nodes is subject to the Quality of Service (QoS) satisfaction of service chains which is formulated as a MILP problem. In [7], a heuristic algorithm based on the Genetic algorithm is proposed to solve the problem of minimizing the number of occupied nodes and links. In that paper, the end-to-end delay is also ignored. Authors in [8], proposed a heuristic algorithm to address a multi-objective problem. Although in the paper the end-to-end delay is concerned as a QoS factor, queuing delay is not calculated correctly and is based on a node parameter.

To the best of our knowledge, none of the existing works considered the VNF embedding to infrastructure in a hierarchical model and the end-to-end delay as a summation of queuing and processing delay. Our main contributions are as follows: 1) VNF embedding to infrastructure in a hierarchical model, 2) defining QoS as end-to-end delay by taking into account the queuing delay. In this paper, we define the problem of maximization of difference of the revenue of service chains' admissions and the cost of the instances' activation and bandwidth usage as a MINLP problem, which is called DA-SFC problem. The constraints of this problem are nodes' capacity (i.e., memory, CPU, and storage), links' capacity (i.e., bandwidth), end-to-end delay threshold, traffic passing through each instance, and resource (i.e., memory, CPU, and storage) requirement of each instance.

The remainder of this paper is organized as follows. In Section II, the system model is proposed. The problem formulation is proposed in Section III. Finally, the simulation results and conclusion are presented in Section IV and V, respectively.

II. SYSTEM MODEL

The substrate network is modeled as a directed graph $G(N_p, E_p)$ wherein N_p indicates the set of nodes, and the set of links is indicated by E_p. Each node $n \in N_p$ is a physical server in the substrate network with a limited amount of CPU, memory, and storage capacity denoted by θ^n_{cpu}, θ^n_{mem}, and θ^n_{strg} , respectively. Also, each physical link $(n, m) \in E_p$ has a limited bandwidth capacity indicated by $\theta^{(n,m)}$.

Set T contains all available VNF types that can be requested by users. The service provider has multiple instances of each type. I_t is the set of instances of type $t \in T$. Since several virtual machines are available on each physical node, multiple instances can be embedded on a physical node. Note that an instance can be mapped just on one individual server. It is assumed that the vendor VNFs determined the amount of required CPU, memory, and storage as well as the admittable traffic volume for each type $t \in T$, which are respectively denoted by CPU_t, mem_t, $strg_t$, and U_t. In order to use an instance of VNF type t its license should be activated, its fee is C_t.

The user requests for a SFC. Set G_v contains all SFCs requested by users. Each SFC $r \in G_v$ is a sequence of VNFs denoted by N^r_v, which are connected by virtual links denoted by E^r_v. For link$(k, l) \in E^r_v$, $w^{(k,l)}_r$ is the required bandwidth between consecutive VNFs $k, l \in N^r_v$ in SFC $r \in G_v$. Each VNF $n' \in N^r_v$ has its specific type indicated by $t^r_{n'}$ and is supposed to be mapped to an instance of the same type. As mentioned before, one of the most important factors of QoS is the end-to-end delay between the source and the destination of each service. The threshold of the maximum tolerable end-to-end delay of service $r \in G_v$ is also denoted byd^r_{th}. If the provider accepts the service $r \in G_v$, it receives R_r units of profit for it.

Since the most effective factor imposes on the end-to-end delay of network services is the various queues that the flow passes within them, we focused on the queuing and processing delays of hypervisors and activated instances in end-to-end delay calculation and others are ignored. The mentioned queues are modeled as M/M/1 queues. In an M/M/1 queue, the arrival process is Poisson, the service time is distributed exponentially, and there is just a single server processing the incoming traffic. According the relations governing the queuing theory, in an M/M/1 queue, if the arrival rate is denoted by λ, the processing rate is denoted by μ, the latency of a packet from its entrance time to the queue till its departure time from the server, denoted by d, is [9]:

$$d = \frac{1}{\mu - \lambda} \qquad (1)$$

The end-to-end delay of a service chain is composed of the delays imposed by the physical nodes that the VNFs of the chain is mapped to them. In each node there is such a queuing network as Fig. 1 declares.

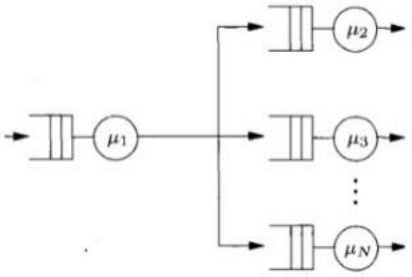

Fig. 1. Queuing network at physical.

The first queue is the hypervisor of the physical node. Its arrival rate equals to the summation of all incoming flows and its processing rate is a parameter determined beforehand. The second queue that a flow passes within at each physical node is the instance queue. This instance is the instance that the corresponding VNF of the mentioned chain is mapped to it. Its arrival rate equals to the summation of all incoming flows that this instances services to them and its processing rate is a determined parameter. Table I discribes all parameters of the system model.

Table I. Parameters definitions

Parameter name	***Parameter definition***
G_p	Infrastructure network graph
N_p	The set of substrate graph nodes
E_p	The set of substrate graph links
$\theta^{(n,m)}$	The available capacity of link $(n,m) \epsilon E_p$
θ^n_{cpu}	The available CPU cores of node $n \epsilon N_p$
θ^n_{mem}	The available memory of node $n \epsilon N_p$
θ^n_{strg}	The available storage of node $n \epsilon N_p$
μ^n_{node}	Processing rate of node $n \epsilon N_p$
BW_{fee}	Bandwidth fee of physical link
G_v	Set of service function chains that all users have requested
N_v	Set of all VNFs of all SFCs
E_v	Set of all virtual links of all SFCs
N^r_v	Set of all VNFs of SFC $r \epsilon G_v$
E^r_v	Set of all virtual links of SFC $r \epsilon G_v$
d^r_{th}	The maximum end-to-end delay threshold of SFC $r \epsilon G_v$
R_r	The revenue of SFC $r \epsilon G_v$
$w^{(k,l)}_r$	The required bandwidth of link $(k,l) \epsilon E^r_v$
T	Set of all available VNF types
I_t	Set of available instances of types $t \epsilon T$
U_t	Traffic capacity of type $t \epsilon T$
C_t	The cost that the provider pays for the activation of each instance of type $t \epsilon T$
CPU_t	The required CPU of each instance of type $t \epsilon T$
mem_t	The required memory of each instance of $t \epsilon T$
$strg_t$	The required storage of each instance of $t \epsilon T$
$t^r_{n'}$	The type of VNF $n' \in N^r_v$ of chain $r \epsilon G_v$

III. PROBLEM FORMULATION

In this section, we formulate the DA-SFC problem. Ar is a binary variable denoting whether SFC $r \in G_v$ is accepted or not. $Ar = 1$ if and only if SFC $r \in G_v$ is accepted and $Ar = 0$ otherwise. In this formulation, in fact, there is a kind of hierarchical mapping with two steps; first, mapping VNFs to appropriate instances, second, mapping instances to physical nodes. $B_{i,t}^{n',r}$ and $B_n^{i,t}$ are the binary variables denoting whether VNF $n' \in N_v^r$ from SFC $r \in G_v$ is mapped to the instance $i \in I_t$ of type $t \in T$ and whether instance $i \in I_t$ of type $t \in T$ is embedded on the physical node $n \in N_p$ respectively. The binary indicator variable $B_{(n,m)}^{r,(k,l)}$ equals 1 if and only if the virtual link $(k,l) \in E_v^r$ of SFC $r \in G_v$ is mapped to the physical link $(n,m) \in E_p$ of the substrate network. $maping\_temp_{n,i}^{n',r}$ is another binary variable, calculated by the multiplication of $B_{i,t_{n'}}^{n',r}$ and $B_n^{i,t_{n'}}$ as equation (2), indicates whether VNF $n' \in N_v^r$ from SFC $r \in G_v$ is mapped to the ith instance of type $t_{n'}$.

$$maping\_temp_{n,i}^{n',r} = B_{i,t_{n'}}^{n',r} * B_n^{i,t_{n'}} \quad \forall r \in G_v, \forall n' \in N_v^r, \forall i \in I_{t_{n'}}, \forall n \in N_p \tag{2}$$

$x_n^{n',r}$ is another binary variable indicating whether VNF $n' \epsilon N_v^r$ of SFC $r \epsilon G_v$ is embedded to the physical node $n \epsilon N_p$. $allocated\_BW_{(m,n)}$ is a real variable indicating the amount of bandwidth allocated from the physical link $(m,n) \epsilon E_p$ to the accepted chains crossing from the link. $y_n^{i,t}$ is the traffic entering the hypervisor queue at node $n \epsilon N_p$ taking service from the instance $i \in I_t$ of type $t \in T$, which is obtained by the multiplication of $B_n^{i,t}$ and $\lambda_{ins}^{i,t}$ as follows:

$$y_n^{i,t} = B_n^{i,t} * \lambda_{ins}^{i,t} \; t \in T, i \in I_t, \forall n \in N_p \tag{3}$$

d_{node}^n and $d_{ins}^{i,t}$ denote queuing and processing delays of the hypervisor in node $n \in N_p$ and the instance $i \in I_t$ of type $t \in T$, respectively. α_n^r is a binary variable that equals 1 if and only if at least one of the VNFs of the chain $r \epsilon G_v$ is mapped to the physical node $n \in N_p$.Variable $delay\_temp_n^r$, which is obtained by equation (4), indicates the delay that SFC $r \epsilon G_v$ tolerates for passing through physical node $n \in N_p$.

$$delay\_temp_n^r = \alpha_n^r * d_{node}^n \quad \forall r \in G_v, \forall n \in N_p \tag{4}$$

$Q_{i,t}^{n',r}$ is another variable denoting the delay that SFC $r \epsilon G_v$ tolerates since one of its VNFs is mapped to instance $i \in I_t$ of type $t \in T$. Equation (5) clarifies its calculation.

$$Q_{i,t}^{n',r} = B_{i,t}^{n',r} * d_{ins}^{i,t} \quad \forall r \in G_v, \forall n' \in N_v^r, \forall i \in I_{t_{n'}}, \forall t \in T \tag{5}$$

$cost$ is the total money payed by the service provider for bandwidth allocation and instance activation $total\_revenue$ is the amount of money that the provider takes for the accepted SFCs as well as $gain$ is the difference between $total\_revrenue$ and $cost$ and indicates the provider' s profit.

The formulation of the DA-SFC problem as MINLP model is as follows.

$$max \; gain \tag{6}$$

s.t.

$$\sum_{i \in I_{n'}} B_{i,t}^{n',r} == Ar \;\; \forall r \in G_v, \forall n' \in N_v^r \tag{7}$$

$$B_{i,t}^{n',r} \leq \sum_{n \in N_p} B_n^{i,t} \quad \forall r \in G_v, \forall n' \in N_v^r, \forall i \in I_{t_{n'}} \tag{8}$$

$$maping\_temp_{n,i}^{n',r} \leq B_{i,t_{n'}}^{n',r} \; \forall r \in G_v, \forall n' \in N_v^r, \forall i \in I_{t_{n'}}, \forall n \in N_p \tag{9}$$

$$maping\_temp_{n,i}^{n',r} \leq B_n^{i,t_{n'}} \; \forall r \in G_v, \forall n' \in N_v^r, \forall i \in I_{t_{n'}}, \forall n \in N_p \tag{10}$$

$$B_{i,t_{n'}}^{n',r} + B_n^{i,t_{n'}} \leq maping\_temp_{n,i}^{n',r} \; \forall r \in G_v, \forall n' \in N_v^r, \forall i \in I_{t_{n'}}, \forall n \in N_p \tag{11}$$

$$\sum_{i \in I_{t_{n'}}} maping_{temp}{}_{n,i}^{n',r} = x_n^{n',r} \; \forall r \in G_v, \forall n' \in N_v^r, \forall n \in N_p \tag{12}$$

$$\sum_{(n,m) \in E_p} B_{(n,m)}^{r,(k,l)} - \sum_{(m,n)} B_{(m,n)}^{r,(k,l)} = x_n^{k,r} - x_n^{l,r} \; \forall r \in G_v, \forall (k,l) \in E_v^r, \forall n \in N_p \tag{13}$$

$$\sum_{r \in G_v} \sum_{(k,l) \in E_v^r} w_r^{(k,l)} * B_{(n,m)}^{r,(k,l)} = allocated\_BW_{(n,m)} \quad \forall (n,m) \in E_p \tag{14}$$

$$allocated\_BW_{(n,m)} \leq \theta^{(n,m)} \; \forall (n,m) \in E_p \tag{15}$$

$$\sum_{t \in f} \sum_{i \in I_t} CPU_t * B_n^{i,t} \leq \theta_{CPU}^n \; \forall n \in N_p \tag{16}$$

$$\sum_{t \in f} \sum_{i \in I_t} strg_t * B_n^{i,t} \leq \theta_{strg}^n \; \forall n \in N_p \tag{17}$$

$$\sum_{t \in f} \sum_{i \in I_t} mem_t * B_n^{i,t} \leq \theta_{mem}^n \; \forall n \in N_p \tag{18}$$

$$\sum_{r \in G_v} \sum_{(l,n') \in E_v^r} w_{i,t}^{(l,n')} * B_{i,t}^{n',r} \leq U_t \; t \in T, i \in I_t \tag{19}$$

$$\sum_{r \in G_v} \sum_{(l,n') \in E_v^r} w_{i,t}^{(l,n')} * B_{i,t}^{n',r} = \lambda_{ins}^{i,t} \; t \in T, i \in I_t \tag{20}$$

$$y_n^{i,t} \leq B_n^{i,t} * M \; t \in T, i \in I_t, \forall n \in N_p \tag{21}$$

$$\lambda_{ins}^{i,t} - \left(1 - B_n^{i,t}\right) * M \leq y_n^{i,t} \; t \in T, i \in I_t, \forall n \in N_p \tag{22}$$

$$y_n^{i,t} \leq \lambda_{ins}^{i,t} * M \; t \in T, i \in I_t, \forall n \in N_p \tag{23}$$

$$\sum_{t \epsilon T} \sum_{i \epsilon I_t} y_n^{i,t} = \lambda_{node}^n \; \forall n \in N_p \tag{24}$$

$$\log(d_{node}^n) + \log(\mu_{node}^n - \lambda_{node}^n) = 0 \; \forall n \in N_p \tag{25}$$

$$\log\left(d_{ins}^{i,t}\right) + \log\left(U_t - \lambda_{ins}^{i,t}\right) = 0 \;\; t \in T, i \in I_t \tag{26}$$

$$x_n^{n',r} \leq \alpha_n^r \; \forall r \in G_v, \forall n' \in N_v^r, \forall n \in N_p \tag{27}$$

$$\alpha_n^r \leq \sum_{n' \epsilon N_v^r} x_n^{n',r} \quad \forall r \in G_v, \forall n \in N_p \tag{28}$$

$$delay\_temp_n^r \leq \alpha_n^r * M \; \forall r \in G_v, \forall n \in N_p \tag{29}$$

$$d_{node}^n - (1 - \alpha_n^r) * M \leq delay\_temp_n^r \; \forall r \in G_v, \forall n \in N_p \tag{30}$$

$$delay\_temp_n^r \leq d_{node}^n \; \forall r \in G_v, \forall n \in N_p \tag{31}$$

$$\sum_{n \in N_p} delay\_temp_n^r = d_{hyp,r} \; \forall r \in G_v \tag{32}$$

$$\sum_{n' \epsilon N_v^r} \sum_{i \epsilon I_t} Q_{i,t}^{n',r} = d_{ins,r} \; \forall r \in G_v \tag{33}$$

$$Q_{i,t}^{n',r} \le B_{i,t}^{n',r} * M \ \ \forall r \in G_v, \forall n' \in N_v^r, \forall i \in I_{t_{n'}}, \forall t \in T \quad (34)$$

$$d_{ins}^{i,t} - \left(1 - B_{i,t}^{n',r}\right) * M \le Q_{i,t}^{n',r} \ \forall r \in G_v, \forall n' \in N_v^r, \forall i \in I_t, \forall t \in T \quad (35)$$

$$Q_{i,t}^{n',r} \le B_{i,t}^{n',r} \ \forall r \in G_v, \forall n' \in N_v^r, \forall i \in I_t, \forall t \in T \quad (36)$$

$$d_{hyp,r} + d_{ins,r} \le d_{th}^r \ \forall r \in G_v \quad (37)$$

$$B_{i,t}^{n',r} \le x_t^i \ \forall r \in G_v, \forall n' \in N_v^r, \forall i \in I_t, \forall t \in T \quad (38)$$

$$\sum_{t \in T} \sum_{i \in I_t} C_t * x_t^i + BW_{fee} * \sum_{(m,n) \in E_p} allocated\_BW_{(m,n)} = cost \quad (39)$$

$$\sum_{r \in G_r} Rvn_r * A_r = total\_revrenue \quad (40)$$

$$total\_revenue - cost = gain \quad (41)$$

$$Ar, B_{i,t}^{n',r}, B_n^{i,t}, B_{(n,m)}^{r,(k,l)}, \in \{0,1\} \quad (42)$$

$$maping\_temp_{n,i}^{n',r}, x_n^{n',r}, \alpha_n^r \in \{0,1\} \quad (43)$$

$$Q_{i,t}^{n',r}, y_n^{i,t}, \lambda_{ins}^{i,t}, d_{node}^n, d_{ins}^{i,t} \in R^+ \quad (44)$$

$$delay\_temp_n^r, allocated\_BW_{(m,n)} \in R^+ \quad (45)$$

$$cost, total_{revrenue} \in R^+ \quad (46)$$

$$gain \in R \quad (47)$$

IV. Simulation Results

In this section, we evaluate the effect of various system parameters on the objective by simulation results. To do so, we consider a directed graph and a set of SFCs as shown in Fig. (2). The bandwidth capacity of all substrate links equal 100 units. Additionally, in each physical node the CPU, memory, and storage available resources equal to each other and from node 0 to node 3 it equals 130, 130, 120, and 120 respectively. It is worth to mention that by setting a coefficient to these physical nodes' capacity, we evaluate the effect of resources on the network profit. The amount of money that the provider pays for each bandwidth unit equals 10. This model is solved twice; firstly, end-to-end delay of all SFCs is 0.1 and then, it equals 0.05. In the first experiment 3 SFCs while in the second one, 2 SFCs are accepted. This is a simple example to verify the correctness of SFC mapping. The properties of SFCs and the VNF types are represented in Table II as well as Table III illustrates the properties of instance types.

Fig. 3 shows the accepted SFCs, and their mappings into the substrate network. The green arrows show the VNF mapping. It is worth mentioning that the instances' mappings are not shown in this Figure.

In the next step, we increase and change the number of substrate nodes and requested SFCs as illustrated in Fig. 4 and Table IV. We observe the influence of some parameters on our proposed model by the simulation results.

One of the most *concerned* QoS factors is end-to-end delay. It is predictable to increase the gain as the end-to-end delay thresholds of requested SFCs increase. To prepare different values for end-to-end delays, we define a coefficient and multiply it to the requested VNF numbers of each SFC and assumed the result as the end-to-end delay threshold of the corresponding SFC. Fig. 5 represents the effect of end-to-end delay threshold on gain. It can be seen that by the increase of delay threshold, the gain increases obviously.

In addition, Fig. 6 illustrates end-to-end delay threshold influence on the number of accepted SFCs. By observing this Fig., we conclude that the number of accepted SFCs usually, but not always, increases when the delay threshold increases.

By comparing Fig. 5 and Fig. 6, it is observed that in some cases when the delay threshold increases, the number of accepted SFCs may stay the same or reduce, since some SFCs worth more. Although in some cases the number of accepted SFCs reduces, the gain always increases.

In addition, we observe the influence of instance capacity on gain and number of accepted instances by setting the data as the previous example but changing the traffic capacity of all instances from 50 to 200. Fig. 7 and Fig. 8 represent its effect on gain and number of accepted SFCs respectively. As shown in Fig. 7 and Fig. 8, by increasing the capacity of the allowed traffic, passing through each instance, the gain always increases but the number of accepted SFCs may increase or not.

For the next step we evaluate the influence of the capacity of the physical nodes on the gain and the number of accepted SFCs. For this aim, we cross a coefficient to the CPU, storage, and memory capacities of all physical nodes. As shown in Fig. 9 and 10, although by the increase of the physical nodes' capacities the gain increases, the number of accepted SFCs might or might not increase.

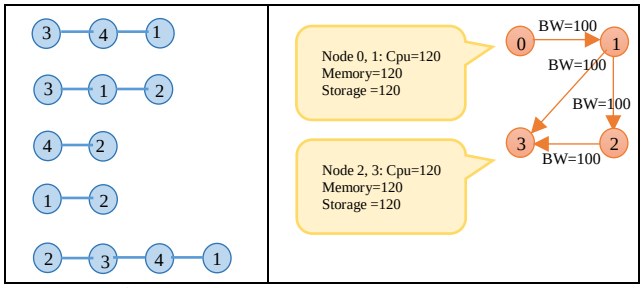

Fig. 2. The left side represents the input SFCs. In each VNF the required instance type is illustrated. The right side represents the substrate network.

Table II. The properties of SFCs

SFC	*Number of required VNFs*	*VNF types*	*Required bandwidth between consequent VNFs*	*Revenue*
0	3	4, 1, 3	20, 20	1050
1	3	1, 2, 3	20, 30	1050
2	2	2, 4	30	700
3	2	2, 1	40	700
4	4	3, 4, 1, 2	40, 30, 60	1400

Table III. The properties of VNF types

Type	*Cost of each instance*	*Required CPU*	*Required Memory*	*Required Storage*
1	70	100	100	100
2	60	100	100	100
3	80	100	100	100
4	90	100	100	100

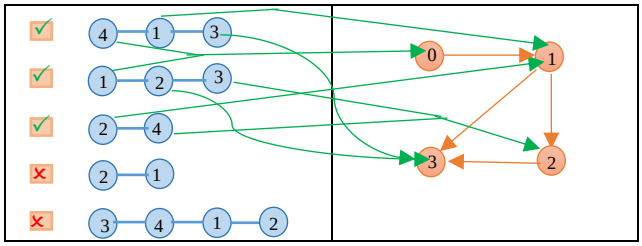

Fig. 3. The accepted SFCs and their mappings where end-to-end delay=0.1.

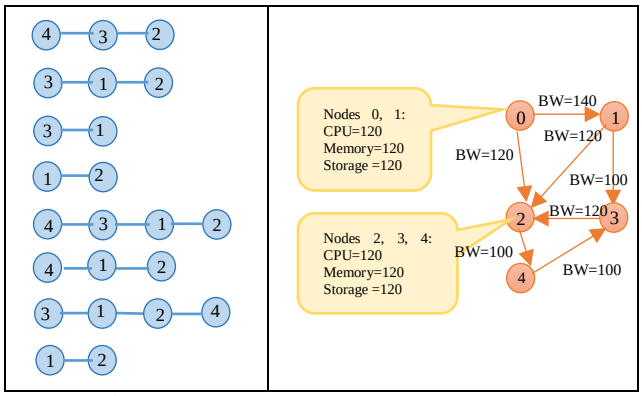

Fig. 4. The left side represents the input SFCs and the right side represents the substrate network where we increase and change the number of substrate nodes and SFCs.

Table IV. The features of the input SFCs

SFCs	***Number of VNFs***	***VNF types***	***Bandwidth of links between consequent VNFs***	***End-to-end delay***	***revenue***
0	3	4, 3,2	40, 40	3*coefficient	3000
1	3	3, 1, 2	40, 50	3*coefficient	3000
2	2	3, 1	30	3*coefficient	2000
3	2	1, 2	40	2*coefficient	2000
4	4	4, 3, 1, 2	30, 40, 50	4*coefficient	4000
5	3	4, 1, 2	20, 30	3*coefficient	3000
6	4	3, 1, 2, 4	10, 20, 20	4*coefficient	4000
7	2	1, 2	30	2*coefficient	2000

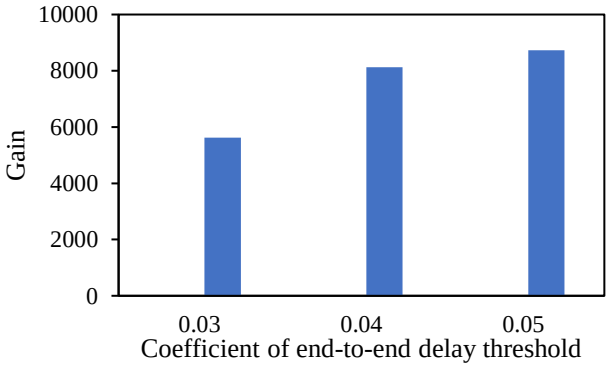

Fig. 5. Gain versus end-to-end delay threshold.

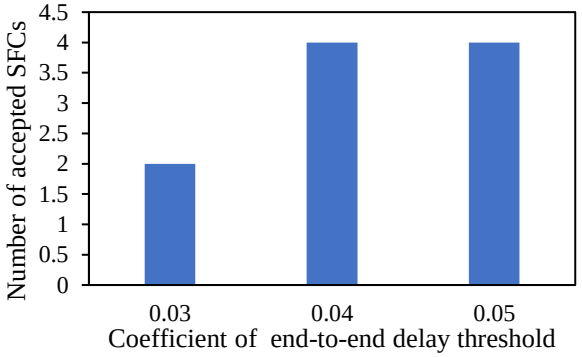

Fig. 6. Number of accepted SFCs versus end-to-end delay threshold.

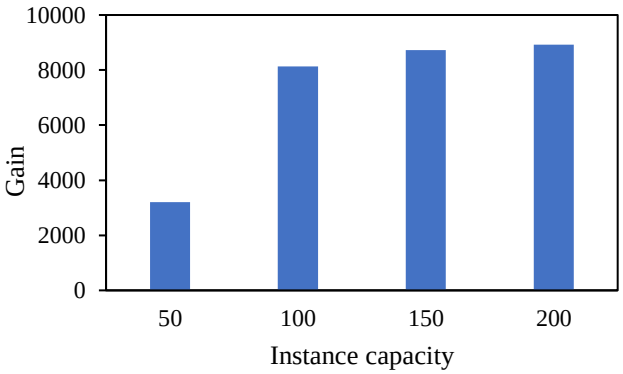

Fig. 7. Gain versus instance capacity.

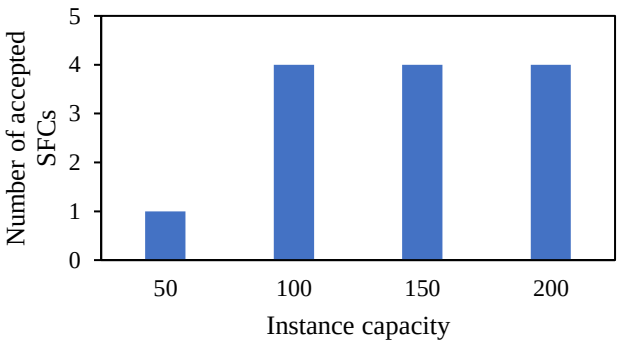

Fig. 8. Number of accepted SFCs versus instance capacity.

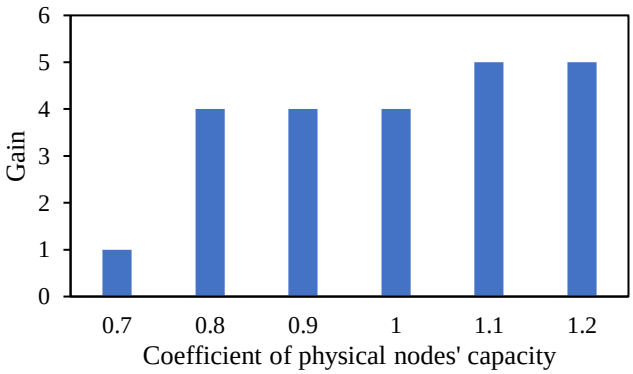

Fig. 9. Gain versus coefficient of the physical nodes' capacity.

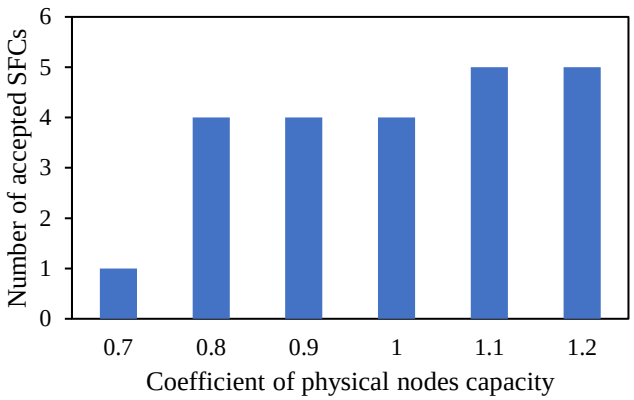

Fig. 10. Number of accepted SFCs versus coefficient of the physical nodes' capacity.

V. CONCLUSION

In this paper, we formulated the VNF embedding problem subject to delay constraint as a MINLP problem. In this formulation, the objective is to maximize the provider's profit such that the constraints of the nodes' capacity (i.e., memory, CPU, and storage), the links' capacity (i.e., bandwidth), the end-to-end delay threshold, and required resources (i.e., memory, CPU, and storage) for each instance are satisfied. The problem was solved optimally by the SCIP optimization tool. Finally, the simulation results verified that by the increase of the end-to-end delay threshold, the capacity of instances, and the capacity of the physical nodes, the gain always increases while the number of accepted SFCs might or might not. It is worth mentioning that when the gain increases while the number of accepted SFCs does not, the solver has selected other SFCs, compared with the previous selection, much more valuable than the previous ones. Therefore, the gain increases but the number of selected SFCs does not.

REFRENCES

[1]. B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Communications Magazine*, vol. 53, no. 2, pp. 90-97, Feb. 2015.

[2]. R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network Function Virtualization: State-of-the-Art and Research Challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 236-262, Firstquarter 2016.

[3]. A. Baumgartner, V. S. Reddy and T. Bauschert, "Combined Virtual Mobile Core Network Function Placement and Topology Optimization with Latency Bounds," *2015 Fourth European Workshop on Software Defined Networks*, Bilbao, 2015, pp. 97-102.

[4]. B. Martini, F. Paganelli, P. Cappanera, S. Turchi, and P. Castoldi, "Latency-aware composition of Virtual Functions in 5G," *Proc. 2015 1st IEEE Conf. Netw. Softwarization*, pp. 1–6, Apr. 2015.

[5]. M. C. Luizelli, L. R. Bays, L. S. Buriol, M. P. Barcellos and L. P. Gaspary, "Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions," *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, Ottawa, ON, 2015, pp. 98-106.

[6]. A. Hmaity, M. Savi, F. Musumeci, M. Tornatore and A. Pattavina, "Virtual Network Function placement for resilient Service Chain provisioning," *2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM)*, Halmstad, 2016, pp. 245-252.

[7]. W. Rankothge, F. Le, A. Russo and J. Lobo, "Optimizing Resource Allocation for Virtualized Network Functions in a Cloud Center Using Genetic Algorithms," in *IEEE Transactions on Network and Service Management*, vol. 14, no. 2, pp. 343-356, June 2017.

[8]. Y. T. Woldeyohannes, A. Mohammadkhan, K. K. Ramakrishnan and Y. Jiang, "ClusPR: Balancing Multiple Objectives at Scale for NFV Resource Allocation," in *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1307-1321, Dec. 2018.

[9]. G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi, "Queuing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications," *John Wiley & Sons*, 2006.