Towards Load-Balanced VNF Assignment in Geo-distributed NFV Infrastructure

Xincai Fei[1] Fangming Liu*[1] Hong Xu[2] Hai Jin[1]

[1]Key Laboratory of Services Computing Technology and System, Ministry of Education,
School of Computer Science and Technology, Huazhong University of Science and Technology
[2]NetX Lab @ City University of Hong Kong

***Abstract*—Network functions virtualization (NFV) is increasingly adopted by telecommunications (telecos) service providers for cost savings and flexible management. However, deploying virtual network functions (VNFs) in geo-distributed central offices (COs) is not straightforward. Unlike most existing centralized schemes in clouds, VNFs of a service chain usually need to be deployed in multiple COs due to limited resource capacity and uneven setup cost at various locations. To ensure the Quality of Service of service chains, a key problem for service providers is to determine where a VNF should go, in order to achieve cost-efficiency and load balancing of both computing and bandwidth resources, across all selected COs. To this end, we present a framework of CO Selection (CS) and VNF Assignment (VA) for distributed deployment of NFV. Specifically, we first select a set of COs that minimizes the communication cost among the selected COs. Then, we employ a shadow-routing based approach, which minimizes the maximum of appropriately defined CO utilizations, to jointly solve the VNF-CO and VNF-server assignment problem. Simulations demonstrate the effectiveness of CS algorithm, and asymptotic optimality, scalability and high adaptivity of the VNF assignment approach.**

I. Introduction

Over the past few years, Network Functions Virtualization (NFV) has attracted extensive attention of telecommunications (telecos) service providers as a means to accelerate service delivery while reducing costs. By moving network function from dedicated hardware to software which is known as *virtual network function* (VNF), NFV is transforming the way that legacy networks are built and operated [1]. As telecos service providers need to serve millions of customers, central offices (COs) are widely dispersed across a large geographical area [2] [3]. These COs serve as a solid base when the service providers set about deploying NFV [4] [5].

Current works mainly focus on the deployment of NFV in a centralized scheme, such as NFV-RT [6] and JVP [7]. Since all customer requests need to be processed in a location, it always leads to increased networking costs and more strict reliability requirements [8] [9]. In the scenario of telecos networks, service providers are usually faced with large number of requests waiting for service. Once these services are implemented by NFV, an urgent task for service providers is to achieve flexible VNF assignment in their operating COs. Due to the relatively limited resource capacity of a CO compared with a data center, VNFs of service chains need to be deployed across multiple COs. For another, the setup cost of VNFs in a CO, which is related to operational and deployment cost (e.g., electricity price [10], bandwidth charges [11], etc.), usually vary from locations. Fig. 1 shows an example of deploying VNFs in geo-distributed COs. A typical application that benefits from such a distributed scheme is the virtualized Customer Premises Equipment (vCPE) devices for Voice over IP (VoIP) service [12]. In this case, the distributed COs operated by service providers become ideal locations to host these vCPE devices (i.e., VNFs). When customers from various locations request their services that need to go through these VNFs, the service providers achieve great flexibility in assigning VNFs and allocating NFV Infrastructure (NFVI) resources in the most economical way, leading to potential reductions of capital expenses. Meanwhile, the end-to-end Quality of Service (QoS) of the VoIP service can fulfill customer expectations.

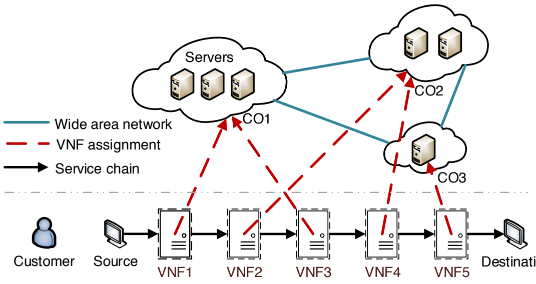


Fig. 1. An example of VNF assignment in geo-distributed COs.

However, deploying VNF chains across multiple COs is a challenging problem. First, traffic demands of these service chains need to travel through the networks among COs. Hence, the communication cost among COs should be minimized. Second, as different VNFs have different resource requirements [13] and the COs have limited resource capacity, it is vital to avoid resource bottleneck in a particular CO, otherwise even one of VNFs in the chain gets overloaded, the whole QoS of the chain also gets affected [14]. Therefore, the load balancing of resources across multiple COs should be carefully achieved. Third, the VNF-server assignment has to respect packing constraints, i.e., the total resources consumed by VNFs assigned to a server should not exceed the resource capacity of the server, which further complicates the problem.

To summarize, *how can we dynamically split the incoming VNF load across a number of geo-distributed COs and pack*

*This work was supported in part by the National 973 Basic Research Program under Grant 2014CB347800, and in part by NSFC under Grant 61520106005. (Corresponding author: Fangming Liu)

VNFs onto servers in the selected CO, and to achieve load balancing of both computing and bandwidth resources, over the long term? In this paper, we consider the problem as two phases: (i) CO Selection (CS) for deploying NFV; and (ii) VNF Assignment (VA) for service chains, including the VNF-CO and VNF-server assignment. The first phase provides optional COs with low communication cost for the second phase so that the performance (e.g., low latency) and connectivity of service chains are likely to be satisfied. Concretely, we aim at minimizing the communication cost among the selected COs. We use the distance between two COs to represent their corresponding communication cost, since longer path usually means larger communication cost (e.g., more bandwidth consumption or larger latency) [7]. We assume that the service provider is subject to a budget, which can be interpreted as her maximum investment on deploying NFV. This corresponds to finding a subgraph with a minimum diameter. We prove its hardness and propose a 2-approximation algorithm to solve it.

For the second phase, we turn to a shadow-routing based approach, which belongs to a virtual queueing system and brings many advantages to solving problems with various objectives and constraints [15] as long as they are within a common framework [16]. Moreover, each of its routing decisions is easily made by choosing an index based on the values and updates of virtual queues. It runs continuously to adapt to changes in network demands. Note that VNF assignment is similar to VM placement [17] due to the fact that both of them have packing constraints. However, VNF assignment brings new challenges in designing the algorithm. In general, the VMs, or jobs, required by tenants only need to be packed onto physical machines in VM placement problem, while there are no relations between any two jobs. In contrast, each service chain is an ordered sequence of VNFs. The traffic demand between any two VNFs should be carefully considered when assigning VNFs, since link bandwidth is also limited resources [18] in a CO and needs to be better utilized as well.

However, in the context that VNFs of a service chain can be assigned to multiple COs, it is more difficult to exactly compute the bandwidth consumption of a service chain in a particular CO. To overcome this, we introduce a binary auxiliary variable, which can be approximately determined by the compositive VNFs of a service chain, to help calculate the bandwidth consumption of a service chain in the CO. Based on this, we design the VNF-CO and VNF-server assignment algorithms, respectively, and achieve load balancing of both computing and bandwidth resource across all selected COs. The proposed algorithm is proven to be asymptotically optimal on the basis of results derived from [16], and this is also verified by the simulations.

The remainder of this paper is organized as follows. In Sec. II, we formally describe the distributed NFV model and the overview of CS-VA framework. We define the CO selection problem, derive its hardness and then present an approximation algorithm in Sec. III. In Sec. IV, we give the linear program form of the VNF assignment problem, and propose a shadow-routing based algorithm in Sec. V. Our scheme is evaluated via simulations in Sec. VI. Sec.VII summarizes related work and Sec. VIII concludes this work, respectively.

II. Model and Overview

In this paper, we consider a Distributed NFV (D-NFV) scenario with a service provider that supports many customers. The service provider deploys a set of NFV service chains in her operating COs, for purpose of serving traffic requests of her customers. The goal of the service provider is to develop a VNF assignment strategy for all VNF types in all service chains, so that load balancing from a network-wide view can be achieved. Note that we consider one service provider, but it can be easily extended to allow more service providers.

A. Geo-distributed NFV model

We model a network as an undirected graph $G = (V, E)$, where the vertices V represents the set of COs and the edges E represents the set of paths among the COs. Each vertex and edge is attached with a weight, namely, c_e is the communication cost of edge (link) $e \in E$ and w_v is the setup cost of CO $v \in V$. The setup cost can be viewed as a combination of multiple factors such as networking resource (i.e., bandwidth) cost and electricity price in the location where CO v is located. The service provider's maximum investment on NFV is subject to a budget, which is denoted as $\mathit{\Delta}$. Each CO contains the needed NFVI resources for hosting service chains. Let the selected COs be indexed by j, and each CO j contains β_j physical servers. Without loss of generality, we assume that the servers in CO j are homogeneous and each has the amount C_{jr} of computing resource $r \in R$, where R is the set of different types of computing resources, as exemplified by CPU and memory. Apart from the computing resources, each CO j is also associated with B_j bandwidth resource. Here, B_j can be computed from the sum of bandwidth capacity of all available links in CO j.

The service provider deploys a set of service chains, indexed by $s \in \mathcal{S} = \{1, 2, \cdots, S\}$. Each service chain s contains a set of service functions (SFs) that the service provider intends to route her customers' traffic through. It should be noted that each SF is instantiated by a type of VNF, at which the required NFVI resources are assigned. The traffic demand of a service chain is called a *request*. The request for service chain s has an input traffic rate of α_s. The set of interconnecting SFs in s is denoted by $\mathbb{L}^s$, where $(v, t) \in \mathbb{L}^s$ if SF v is the predecessor of SF t in service chain s. SFs in service chains s arrive at the rate λ_s. Notably, we focus on single-path flows through network forwarding graphs of service chains.

Say a set of $\mathcal{I} = \{1, 2, \cdots, I\}$ types of different VNFs are offered in the system. Each VNF is instantiated in a VM instance in order to support an SF in the service chain. We assume that each VM can run at most one VNF and different VMs can support the same VNF. These VMs running VNFs all have specific configurations and require fixed resources (see the simulation setup). A VNF $i \in \mathcal{I}$ consumes c_{ir} of resource $r \in R$. When a service chain s is placed for service, each SF in s is allocated the required amounts of resources. We

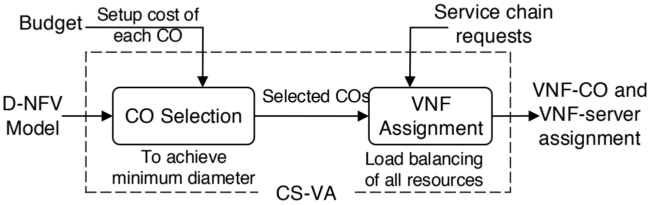


Fig. 2. Overview of CS-VA.

assume that the average service time of each VNF $i \in \mathcal{I}$ is $1/\mu_i$. After all SFs in s are finished, service chain s releases all resources allocated to them and leaves the system.

B. An overview of CS-VA

For implementing NFV in telecos networks, the service provider needs first to determine (i) which COs should be selected for deploying VNFs, and (ii) for a specific VNF in a service chain, which CO should the VNF be routed to from the selected COs in (i). After the VNF-CO assignment, the service provider then needs to decide which physical server the VNF should be packed in the CO. The joint assignment of VNF-CO and VNF-server makes it possible for the service provider to achieve load balancing from a network-wide view, otherwise the VNF load cannot be dynamically split. To sum up, an assignment of service chains involves assigning every VNF in service chains to a certain number of COs that have enough available resources, and assigning VNFs to the corresponding physical servers in a particular CO. CS-VA is such a framework, which is shown in Fig. 2, that a service provider can use to solve the above problems.

In the next three sections, we present our solutions to CS-VA in detail.

III. CO SELECTION FOR NFV DEPLOYMENT

In this section, we present our solution that selects a subset of COs for NFV deployment, such that it minimizes the maximum communication cost between any two COs.

A. Hardness of CO selection problem

CO selection (CS) problem can be viewed as a subgraph selection problem. Many problems on graphs seek a subgraph $G'(V', E')$ of the original graph $G(V, E)$ that satisfies a set of constraints and has minimum total weight, either vertex weight or edge weight. In our case, CS problem corresponds to finding a largest subgraph G' of G, whose sum of weights on vertices is at most budget $\varDelta$ and with a minimum diameter (i.e., maximum communication cost between any two COs). It is NP-hard, since it contains other well-known NP-hard problems (e.g., maxclique [19]) as subproblems. Because the original graph G is a complete graph, the subgraph obtained from G is also a complete graph. Furthermore, all graphs represent undirected graph unless otherwise specified. For ease of presentation, we replace the communication cost between two vertices as distance. The CS problem is given as follows:

Problem 1. *Find a largest subset $V' \subseteq V$, such that $\sum_{v \in V'} w(v) \leq \varDelta$, and the diameter of V', $d(V') = \max\{d(v,t) : v, t \in V'\}$, is minimized.*

Lemma 1. *Unless P=NP, no polynomial approximation to Problem 1 with an approximative ratio of less than 2 exists, even when the triangle inequality holds.*

Proof: Consider a circle with two sides. We put all vertices belonging to V' on the right side and all vertices belonging to $V \setminus V'$ on the opposite side. The distances between two vertices that locate on the opposite side are defined to be 2, while other distances on the right side are defined to be 1. If the triangle inequality holds and there exists a feasible solution if and only if this solution to Problem 1 is 1. Otherwise, the solution is 2. Therefore, it ensures that an approximation algorithm which produces a solution at most δ times the optimal, for $\delta < 2$ must obtain a solution of diameter 1. Hence, if $P \neq NP$, the algorithm cannot be polynomial, otherwise the algorithm is at most 2-competitive. ■

Remark. Lemma 1 corresponds to the results in [20], which is derived from the creation of Maxclique problem [19] to the original problem. Judged by Lemma 1, Problem 1 does not exist an competitive ratio of less than 2 in polynomial time. Hence in the later, we assume that triangle inequality holds for the edge weights in the graph. If a triangle inequality violates, then the triangle inequality can be established by replacing appropriate distance for the edge weights.

B. A 2-approximation algorithm

Now we present the algorithm of CO selection for deploying VNFs of service chains. Specifically, in the algorithm, each vertex v in V takes turns to be set as a staring vertex. With a starting vertex, other vertices are sorted in an increasing order of distance to v, denote by $\{v_1, v_2, \cdots, v_{n-1}\}$ (Assume $|V| = n$). The algorithm first adds vertex v_1 and corresponding edges into the subgraph, while computing the maximum diameter of the subgraph. Then, it takes turns to add vertex v_2, v_3, ..., to the subgraph. Once a new vertex is added, the edges induced by the new vertex will be added to the subgraph, accordingly. Meanwhile, the maximum diameter of the subgraph maintains updated with the new vertex added. This process will last until all vertices are added to the subgraph or the total sum of weights on all added vertices exceeds the budget, i.e., each iteration of the loop should satisfy $\sum_{v_i \in V'} c(v_i) \leq \varDelta$, where V' is the set of vertices of the current subgraph. From each iteration t, we can get a maximum diameter of a subgraph that satisfies the budget constraint. At last, the algorithm chooses the subgraph with a smallest maximum diameter among all subgraphs. Assume this subgraph is obtained in iteration t, then vertex $\{v_0^t, v_1^t, \cdots\}$ will be selected as target COs for deploying VNFs in service chains. Algorithm 1 summarizes such process of CO selection.

Theorem 1. *The diameter obtained from Alg. 1 is at most twice the optimum, while the sum of weights on vertices is at most $\varDelta$.*

Here Alg. 1 being at most twice the optimum means it is a 2-approximation algorithm. The basic idea of the algorithm above is derived from the results of [20]. Therefore, details of the optimality proof of Theorem 1 can be found in [20].

Algorithm 1 CO selection algorithm

```
Input: Graph $G(V,E)$ with vertex and edge weights, budget $\varDelta$ of the service provider
Output: a set of vertices $\Lambda_t$.
Initialize $\Lambda=\emptyset, D=\emptyset, t=1$;
for each $v \in V$ do
    Let $V'=\{v\}$, $E'=\emptyset$, $cost=c(v)$;
    Let $v_1, v_2, ..., v_{n-1}$ be the vertices of $G$ sorted in increasing order of weight on edges to $v$;
    $i=1, \Lambda_t=\{v\}, d_t=0$;
    while $cost < \varDelta$ and $i < n$ do
        $E'=E' \cup \{(v', v_i) : v' \in V'\}$;
        $V'=V' \cup \{v_i\}, \Lambda_t=\Lambda_t \cup \{v_i\}$;
        $cost=cost+c(v_i)$;
        $d_t=\max\{d_t, \{c(v', v_i) : v' \in V'\}\}$;
        $i=i+1$;
    end while
    $D=D \cup \{d_t\}$;
    $t=t+1$;
end for
$t=\arg\min_{\{t: d_t \in D\}} d_t$;
return $\Lambda_t$.
```

The worst case running time complexity of the algorithm is $O(n^3)$, where n is the total number of optional COs. Note that Theorem 1 also conforms to our proposed Lemma 1.

IV. VNF assignment in CO and server level

With COs selected in the previous section, we consider the VNF assignment in CO and server level in this section.

A. Feasible configurations for VNFs

Recall that the VNF-server assignment should respect packing constraints. To establish the connection between the assignment of VNF-CO and VNF-server, we first obtain the feasible configurations for all types of VNFs. Specifically, as multiple VM instances can be provisioned in a physical server, multiple VNFs can thus be instantiated in a physical server as long as the amount of required resources is allocated. This means, in particular, that a physical server in CO j can simultaneously serve a number of f_i VNF instances of the same type i. We denote a vector by $f=(f_i, i \in \mathcal{I})$ if

$$\sum_{i \in \mathcal{I}} f_i c_{ir} \leq C_{jr}, \forall r \in R,\ \forall j. \tag{1}$$

Vector f means the number of simultaneously supported VNF instances of different VNF types in a physical server. As stated in [17], we call such vectors f that satisfy the condition (1) as *feasible configurations* of a physical server in CO j. Configuration $f=(f_1, f_2, \cdots, f_I)$ means the number of VNF i placed in a physical server will never be larger than f_i. The set of configurations which are maximal and not dominated by other configurations is denoted by F_j. For all VNF types, there exists VNF $i \in \mathcal{I}$ such that $f_i \geq 1$, otherwise VNF i can not be served in any physical machines at all.

In the initial deployment phase of NFV, the most urgent task for the service provider is how to deploy the needed VNFs of different types in her operating COs, according to the customer demands for going though a set of service chains. Namely, given the set of service chains required by customers, the number of SFs implemented by the same type of VNF can be obtained. Being aware of the arrival rate of SFs in all service chains, the arrival rate of a specific VNF type can then be computed as well.

B. The problem of minimizing maximum system utilization

Our dynamic VNF provisioning problem in multiple distributed COs is to decide which CO a VNF should go and how to pack the VNF onto a physical server in the selected CO. For each CO j, we define its *physical server utilization* (*PS-utilization*) as the fraction of non-idle physical servers inside it, and define *B-utilization* as the fraction of bandwidth that service chains are in use. The objective of the problem is dynamically provisioning VNFs, in a way that minimizes the maximum of all average PS-utilizations and all average B-utilizations across all COs. This objective can be naturally viewed as the load balancing of all kind of resources across all COs. From another perspective, it can also be treated as an objective of system capacity maximization. Because if it can be achieved, the system will have more spared bandwidth resource and more residual physical servers to accept more service chain requests, across all COs.

More specifically, we define a binary function $x_i^s=1$ if VNF $i \in \mathcal{I}$ is in service chain $s \in \mathcal{S}$, otherwise 0. We also define routing variable $A_{vt}^{sj}(p)=1$ if SF $v \to t$ traffic of service chain s is routed on path $p \in \mathbb{L}^s(v,t)$ in CO j, $\forall v, t \in \mathcal{I}, s \in \mathcal{S}$, otherwise 0. Then the average arrival rate λ_i of type i VNF is as follows:

$$\lambda_i=\sum_{s \in \mathcal{S}} x_i^s \lambda_s, \forall i \in \mathcal{I}, \tag{2}$$

and the bandwidth utilization φ_j in CO j is

$$\varphi_j=\sum_{s \in \mathcal{S}} \sum_{p \in \mathbb{L}^s(v,t)} A_{vt}^{sj}(p) \alpha_s / B_j, \forall v, t \in \mathcal{I}, \forall j. \tag{3}$$

Here, it worth noting that the flow rate of traffic may change after being processed by a SF [21], e.g., firewall, might drop some packets or reshape the traffic. Our formulation that computes the bandwidth utilization in (3) can be easily extended to deal with this traffic change, by simply adding a gain/drop factor for each VNF i in each service chain s.

Denote by λ_{ij} the average arrival rate of VNF i routed to CO j; by $\phi_{fj} \geq 0$ the average fractions of used physical servers in CO j under the configuration $f \in F_j$. The objective function is denoted by ρ, which can be interpreted to the average system utilization across all COs. Then, the VNF provisioning problem for minimizing the maximum load of all COs can consider the following *static planning problem* (SPP), first introduced in [22], which is a linear program (LP):

$$\min_{\{\lambda_{ij}\},\{\phi_{fj}\},\{\varphi_j\},\rho} \rho \tag{4}$$

subject to

$$\lambda_{ij} \geq 0,\ \forall (i,j), \quad \phi_{fj} \geq 0,\ \forall (f,j), \tag{5}$$

$$\varphi_j \leq \rho,\ \forall j, \tag{6}$$

$$\sum_{j} \lambda_{ij} = \lambda_i,\ \forall i, \tag{7}$$

$$\lambda_{ij}/(\beta_j \mu_i) \leq \sum_{f \in F_j} f_i \phi_{fj},\ \forall (i,j), \tag{8}$$

$$\sum_{f \in F_j} \phi_{fj} = \rho,\ \forall j. \tag{9}$$

The constraint (6) means that the B-utilization of any CO cannot larger than ρ, where φ_j is obtained from equation (3). Constraint (7) makes sure that the total arrival rate of a VNF i equals to the arrival rate of VNF i to all COs, where λ_i is determined by (2). Constraint (8) holds because $\lambda_{ij}/\mu_i \leq \sum_{f \in F_j} f_i \phi_{fj} \beta_j$, where λ_{ij}/μ_i is the average number of VNF i in service at CO j, and $\phi_{fj}\beta_j$ is the average number of physical servers in CO j that are in use under configuration f, hence, $f_i \phi_{fj} \beta_j$ is the maximum average number of VNF i served by physical servers in configuration f.

Even though the meaning of ρ is the average fraction of utilized resources, the LP (4)-(9) still make sense if the optimal ρ is greater than 1. In this case, the system cannot process all service chain load, and will be overloaded. On the contrary, if the optimal $\rho < 1$, all service chain demands requested by customers can be dealt with. Further, this means the underlying stochastic process of the system can keep stable, as long as the virtual queues waiting for service is allowed to be queued.

An important aspect we need to discuss is about the variable φ_j in constraint (6) of the LP. It is clear that φ_j cannot be intuitively obtained, as the binary variable $A^{sj}_{vt}(p)$ in equation (3) needs to be determined. Consider the following situation: the system supports a number of customers, each of which requests for going through a chain of services. Since a service chain is a predefined order of SFs, these SFs running on some certain types of VNFs cannot be totally separated from one to another, especially when they are allocated to the same CO. If the arrival rate of any VNF i routed to a CO j (i.e., λ_{ij}) is given, then according to interconnectivity of SFs in the service chain s, whether the traffic of s will pass through a path $\{p(v,t), \forall v, t \in \mathcal{I}\}$ or not in CO j can be determined. That is to say, the 0-1 value of variable $A^{sj}_{vt}(p)$ depends on λ_{ij} in the LP above. According to Sec. III, we can assume that there is sufficient capacity for transmitting traffic of interconnecting SFs located at different COs.

Imagine when the system is on a large scale, in the sense of a large number of service chains waiting to be served. In this extreme case, all λ_i, β_j and total amounts of available bandwidth will be large simultaneously. Therefore, when the optimal $\rho < 1$, not only all offered service chains can be handled, but also the system can be controlled in a way so that there exist no virtual queues (i.e., all SFs in all service chains can be assigned into the physical servers in the chosen COs for service on time, while the bandwidth requirements of all service chains can also be satisfied). As a result, PS-utilization in configuration f in CO j keeps close to an optimal ϕ_{fj}, and B-utilization in CO j becomes non-random meanwhile $\varphi_j, \forall j$ is not exceeding ρ.

In a scenario of telecos network, a service provider in general is capable of supporting a large number of customers, each of whom will request one or more service chains. Hence, we focus on dealing with large scale system. To ensure the system can process the offered load, our target is to design a scheme that optimally completes the VNF-CO and VNF-server assignment. Next, we give such a scheme in detail.

V. Shadow routing based scheme

A. Construction of the virtual queueing system

Now we start to construct virtual queueing system. First of all, a concept of virtual (shadow) queues need to be spelt out. That is, for a CO j, the resource requirements induced by customer service requests can be supposed to be virtual queues that the system needs to maintain and handle. The algorithm maintains and updates the virtual queues and makes routing and service decisions based on the current state of the system.

In general, we assume for simplicity that the arrival rate of each VNF i is Poisson. Then the sequence of VNF arrivals is determined randomly, regardless of their types. Namely, the arrival of VNF i is not dependent on other arrivals. In regard to each CO j, there are two associated virtual queues. For each VNF i, there is an associated virtual queue $q(i,j)$, whose length is denoted by Q_{ij}. Another virtual queue $q(j)$ is associated with bandwidth requirement in CO j, whose length is similarly denoted by Q^b_j. It is worth noting that the virtual queues are just variables maintained by the algorithm, they are not actual queues where VNFs, or anything else, waiting for service. And there is no predefined limit on the queue length.

When a VNF arrives, and its type is i, the algorithm rapidly makes a decision on routing it to one of the COs. If the chosen CO is k, then the algorithm places amount of "workload" $1/(\beta_k \mu_i)$ into virtual queue $q(i,k)$, namely $Q_{ik} = Q_{ik} + 1/(\beta_k \mu_i)$ and place amount of "workload" $\alpha^s_i/(\varphi_k B_k)$ into virtual queue $q(k)$, namely $Q^b_k = Q^b_k + \alpha^s_i/(\varphi_k B_k),\ \forall s \in \mathcal{S}$. Here, α^s_i is the traffic going through VNF i in service chain s, and $\varphi_k B_k$ means the induced bandwidth consumption for serving VNF i in CO k.

After the routing decision to a certain CO is made and corresponding updates on the involved variables are done, the algorithm then needs to decide whether or not to activate a controller (or call superserver in [17]) for the virtual queues. If the controller is activated, then a "control mode" $\pi^j \in F_j$ is chosen for each CO j. In order to keep the virtual queues updated, the amount of "workload" $c\pi^j_i$ is removed from each virtual queue $q(i,j)$, namely $Q_{ij} = \max\{Q_{ij} - c\pi^j_i, 0\}$ and the amount of "workload" c is removed from virtual queue $q(j)$, namely $Q^b_j = \max\{Q^b_j - c, 0\}$. Here, $c > 0$ is another parameter. To ensure the virtual queue system to be sufficient to maintain the incoming load, the choice of parameter c should satisfy

$$c > \max_{i,j} 1/(\beta_j \mu_i) \text{ and } c > \max_{i,j} \alpha^s_i/(\varphi_j B_j),\ \forall s \in \mathcal{S}. \tag{10}$$

It can be easily seen that such c satisfying (10) will be sufficient to keep all virtual queues in the system stable, under the control of the controller of the algorithm. In other words, the virtual queues will never run away to infinity.

In the following, we aim at designing a routing algorithm, such that the average times of controller activation is minimized, and all virtual queues remain stable at the same time. The virtual queueing system described above and the problem of assigning VNFs to COs and allocating the required bandwidth resource belongs to the framework of general model in [16], which introduces a Greedy Primal-Dual (GPD) algorithm and establishes asymptotic optimality of the GPD algorithm. Inspired from the work [17] for VM placement and the original work [16], we employ GPD for VNF assignment and bandwidth allocation in our distributed scenario. The proposed algorithm is asymptotically optimal, which means the average rates of different VNFs routing different types of VNFs to the COs are close to the optimal solution of LP (4)-(9).

Remarks on model assumptions. The assumption that λ_{ij} and μ_i for the CO j and VNF i are fixed numbers is not essential. They can be non-negative random variables with finite means. This can be thought of the dynamic nature of incoming service chains, as well as their component VNFs in the chains. For another, we need recall that virtual queues are not buffered in real physical buffers, or VNFs and bandwidth requirements are placed for waiting; but they are just variables maintained by the algorithm. Hence, the length of virtual queues is not concerned with the waiting times of actual service chain request. Moreover, in the case that the result of ρ obtained from the algorithm is close to the optimal value ρ^* meanwhile $\rho < 1$, the incoming service chains will be served immediately without any waiting time.

B. Algorithm of VNF assignment to COs

The assignment of VNFs to different COs is achieved by Alg. 2. Denote by D_j the subset of all chosen COs, on the condition that one type of VNF i can at least fit into a physical servers in these COs $j \in D_j$, i.e., $c_{ir} \le C_{jr}, \forall r \in R$.

To better understand the algorithm, we give explanations on some key steps. The choice of a CO in (11) is based on the corresponding virtual queue length of all feasible COs and the algorithm chooses the the CO with minimum queue length. The setting on the small parameter η in (13) is for minimizing the frequency of controller activations. The parameter c should satisfy condition (10). Newly introduced $E(f, \pi^j_{i'})$ in (14) is a binary variable, and its 0-1 value is given as follows:

$$E(f,\pi^j_{i'}) = \begin{cases} 1 & \text{if } f = \pi^j_{i'} \text{ and condition (13) holds,} \\ 0 & \text{otherwise.} \end{cases}$$

It should be noted that the update of variable ϕ_{fj} (i.e., the connection between VNF-CO and VNF-server) needs to be used for the assignment of specific VNF to physical servers in the chosen CO in Sec. VI-D.

Next, we claim that when parameter η is small, the output of VNF-CO assignment rates from Alg. 2, together with configuration usage fractions, is close to optimal solutions of LP (4)-(9). The following theorem gives the asymptotic optimality of Alg. 2.

Algorithm 2 VNF assignment to center offices

```
Input: A small parameter $\eta > 0$, parameter $c$, small parameter $\theta > 0$, and
       other given parameters
Output: Assignment of VNFs to corresponding COs; $\phi_{fj},\ \forall f, j$
1: Upon each new arrival of VNF $i$, do the following:
   1) Compute CO index
        $k \in \arg\min_{j \in D_j} [Q_{ij}/(\beta_j \mu_i) + Q^b_j \alpha^s_i/(\varphi_j B_j)].$   (11)
   2) Route this VNF $i$ to CO $k$ and for this $k$ do the following updates:
        $Q_{ik} = Q_{ik} + 1/(\beta_k \mu_i),$
        $Q^b_k = Q^b_k + \alpha^s_i/(\varphi_k B_k),\ \forall s \in \mathcal{S}.$
   3) For each CO $j$, compute an eligible configuration
        $\pi^j \in \arg\max_{f \in F_j} \sum_{i' \in \mathcal{I}} f_i Q_{i'j}.$   (12)
      If condition
        $\eta \sum_j [\sum_{i' \in \mathcal{I}} \pi^j_{i'} Q_{i'j} + Q^b_j] \ge 1$   (13)
      holds, then do the following updates:
        $Q_{i'j} = \max\{Q_{i'j} - c\pi^j_{i'}, 0\},$ for all $j$ and $i' \in \mathcal{I}$,
        $Q^b_j = \max\{Q^b_j - c, 0\},$ for all $j$.
   4) For each CO $j$, update the following usage fractions under different
      configurations:
        $\phi_{fj} = \theta E(f, \pi^j_{i'}) + (1-\theta)\phi_{fj}, \forall f \in F_j, i' \in \mathcal{I}.$   (14)
2: Return $\phi_{fj}, \forall f, j$.
```

Theorem 2. *Assume that condition (10) holds for all system parameters and all parameters in Alg. 2 (except η) being set to be fixed reasonable parameters. When the parameter η is close to 0, there exists a control strategy so that the virtual queueing system can be kept stable and reach a sequence of stable states. Denote by $\hat{\phi}_{fj}$ the probability of the steady state, which is updated from (14) when configuration $f = \pi^j_i$ and condition (13) holds, for a fixed η; denote by $\hat{\psi}_{ij}$ the steady state probability of VNF i is assigned to CO j. Then, the distribution of these states to a limiting point when $\eta \to 0$ satisfy:*

$$\lambda c \hat{\phi}_{fj} = \phi^*_{fj},\ \forall (f,j), f \in F_j,$$
$$\lambda_i \hat{\psi}_{ij} = \lambda^*_{ij},\ \forall (i,j),$$

where $\lambda = \sum_{i \in \mathcal{I}} \lambda_i$, and $\{\phi^*_{fj}\}, \{\lambda^*_{ij}\}$ is an optimal solution of LP (4)-(9).

The shadow algorithm is obviously within the general framework of [16]. The steady state can be reached by the virtual queueing system for any value η, which means the controller has sufficient capacity to hold the arriving "workload" in the virtual queues. Then the conditions (more details can refer to section 4 in [16]) for proving the asymptotic optimality of shadow algorithm will hold or can be established. The proof of Theorem 2 is identical to Proposition 1 in [17], so see more details in [17].

C. Algorithm parameter setting

In practical implementation stage, the parameters involved in the algorithm need to be properly set. To see the form of condition (10) , parameter c can be set as $c = 1.01\max\{C_1, C_2\}$, where C_1 and C_2 are the right-hand sides of condition (10), and it is sufficient to maintain the virtual queues stable. The parameter $\theta \in [0,1]$ is not very crucial, so it can be set to $\theta = 0.01$ for example in our implementation.

Next we discuss the setting of parameter η. According to the general results in [16], when $\eta \to 0$, the adjusted virtual queue lengths ηQ_{ij} and ηQ_j^b converge to a set of optimal non-negative finite values, respectively. Hence, it means that the variable ηQ_{ij} and ηQ_j^b become nearly constant (i.e., do not fluctuate much) in condition (13) as $\eta \to 0$. Although the smaller η the more stationary that the system will stay, the system transforming from the current steady state to another new steady state will spend time in of the order $O(1/\eta)$. As a result, the desirable η setting should not be too small as well. Recall the structure of condition (13) in Alg. 2, we can obtain $\eta[J(B+I)c] \approx 1$ when the parameters are set to be "crude" values, where J is the number of COs and B is the average bandwidth of all COs. In our simulation, we use an adjustment parameter ϵ for η, such that $\eta = \dfrac{1}{J(B+I)c\epsilon}$ with $\epsilon = 3, 5, 7$.

It should be noted that the initial state of the virtual queues does not matter, in that the algorithm runs continuously.

D. Algorithm of VNF assignment to physical servers

In this subsection, we focus on the assignment of VNFs to physical servers in any CO j, after determining which CO should VNF i be routed to. Then, we give the corresponding algorithm.

In fact, there exist empty and non-empty physical servers in a CO, simultaneously. For each non-empty physical server, it is associated with a configuration $f \in F_j$. Recall that configuration $f = (f_1, f_2, \cdots, f_I)$ means the number of VNF i placed in a physical server will never be larger than f_i. We aim at minimizing the average fraction of used physical servers in all COs. If a physical server is empty, there is no configuration associated with it. For each configuration f in CO j, denote by $v_i^j(f)$ the total number of type i VNFs under configuration f. Parameter ϕ_{fj} obtained from Alg. 2 will be used in this algorithm once VNFs associated with configuration f is waiting to be assigned. The algorithm for assigning VNFs to physical servers is shown in Alg. 3.

Remark. In actual process of VNF assignment, in order to save more bandwidth in a CO, a VNF prefers to steel the traffic to its successor VNF inside the same physical server as it will not consume bandwidth resource when traffic of a service chain going through the same physical server. So when we design the algorithm for assigning VNFs to physical servers, whether VNFs of different types waiting to be served belong to a same service chain needs to be taken into account. Specifically, when a VNF i arrives at a CO, we will not always choose the physical server with maximal residual space. Instead, if there exist VNFs belonging to the successor VNFs of VNF i in a same chain in another physical with feasible configuration (i.e., it is sufficient to hold VNF i in this physical server), then this physical server will be chosen.

Algorithm 3 VNF assignment to physical servers

```
Input: Configuration parameter $\phi_{fj}$
Output: Assignment of VNFs to physical servers in CO $j$
Upon each new arrival of VNF $i$ in CO $j$, compute configuration index
    $$f' \in \underset{f \in F_j : f_i > 0}{\arg\min}\ v_i^j/(f_i \phi_{fj}).$$
Choose the physical server with maximal residual space for holding VNF $i$ under the configuration $f'$.
if the existing number of type $i$ VNFs is less than $f_i$ then
    Assign the VNF to this physical server.
else
    Place the VNF to an empty physical server and add configuration $f'$ associated with this newly allocated physical server.
end if
```

VI. Evaluation

In this section, we conduct simulations to evaluate the performance of our designed CS-VA scheme, including CO selection (CS) and VNF assignment (VS).

A. Simulation setup for CS-VA

CS: To reflect the real location of COs, we obtain the CO information from a central office lookup tool [3], which provides detailed information about specific COs including location, customer, etc. We choose the location information of 20 COs in New York State as our trace. And the position coordinates of each CO are computed from its $\langle$latitude, longitude$\rangle$ pair. As assumed in Sec. IV-A, we use the shortest distance between two COs to roughly estimate the communication cost between them. The setup cost of each CO is chosen uniformly from 10 (\$1,000×) to 30 following random distribution, and the total budget $\varDelta$ ranges from 100 to 300 in the simulation.

VA: A set of four COs is considered. The total amounts of available bandwidth in the four COs are the same and set to be 320 Gbps. The number of physical servers in the four COs are set to be $\{10, 10, 15, 20\}$, respectively. Each VNF instance requires two types of resource: CPU and memory. The configurations of physical servers in the four COs is shown as in Table I. The service provider offers eight types of VNFs, whose resource requirements is within a reasonable range compared with real VNF products, as shown in Table II (denote by F1 $\to$ F8). We assume that the incoming VNFs is a Poisson process with average inter-arrival time of 2 seconds. Then the arrival of these VNFs is independent of each other and the probability of each arriving VNF type follows random distribution. The service time of VNFs follows normal distribution with average of 20 minutes. Other parameters in the algorithm are set according to Sec. VI-C. The running time is set to be 16 hours in the simulation.

B. Effectiveness of CO selection

To evaluate the effectiveness of proposed CS algorithm, we first implement two baselines for comparison: (i) SCF, which means that the baseline algorithm selects COs according to the

TABLE I

CONFIGURATIONS OF PHYSICAL SERVERS IN THE FOUR COS

Options	CO1	CO2	CO3	CO4
CPU (GHz)	42	40	26	20
Memory (GB)	96	96	72	8

TABLE II

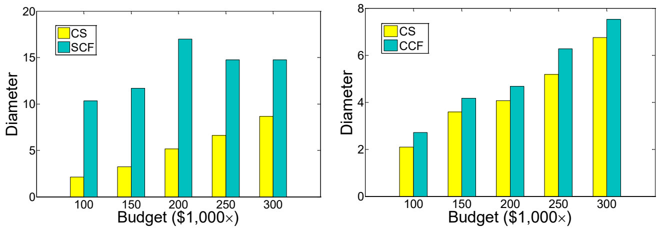
RESOURCE REQUIREMENTS FOR EACH TYPE OF VNF

Options	F1	F2	F3	F4	F5	F6	F7	F8
CPU (GHz)	6.7	5.2	2.6	4	1.3	1.6	0.8	1
Memory (GB)	11.5	32.4	17	3.5	8.5	7.5	3.8	0.9

"setup-cost-first" principle. Namely, the algorithm first selects a CO with the least cost and selects a CO with the second least cost, and so on; (ii) CCF, which means that the baseline algorithm selects COs according to the "communication-cost-first" principle from a predefined starting CO with the least setup cost. Fig. 3(a) and Fig. 3(b) depict the comparison results of two baselines with our proposed CS algorithm (denote by CS), respectively. The results in Fig. 3(a) show that CS significantly outperforms SCF under all settings of budget, in that the diameter of SCF is several times and even more than that of CS. As the budget of the service provider increases, increasing trend of diameter can be observed in CS. This is because more COs will be selected as the budget increases, which will be more likely to obtain a larger diameter among these COs. When it comes to SCF, the diameter has no trend since the CO is selected according to its setup cost regardless of distance. In Fig. 3(b), increasing trend can be observed in both CS and CCF, but their diameters are relatively close. This is because, CCF first selects a CO with minimum setup cost, and then completes the selection process according to the distance and output a distance in just an iteration. While CS executes CO traversal for all COs in the network, and pick a minimum diameter from the diameters of all iterations. Both of the two figures show the effectiveness of our CS algorithm.

C. Efficiency of VNF assignment

Asymptotic optimality of the algorithm. Fig. 4 plots the physical utilization at the four COs with parameter $\epsilon = \{3, 5, 7\}$, respectively, in the algorithm. The algorithm runs continuously and reaches at a steady state. We use Gurobi [23] as the LP solver with 16 parallel threads to calculate the optimal utilization $\rho = 0.46$, and plot it on the figures for comparison. According to the configuration of physical servers in the four COs and the resource requirements of eight types of VNFs, the physical server utilization will fluctuate up and down at the optimal value in all COs. All the curves in Fig. 4 verify this fact and also conform to the asymptotic optimality of the shadow routing based algorithm. Another fact that should be noted is that the algorithm work stably with different values of ϵ. In other words, the shadow algorithm is nearly not influenced by the setting of parameter ϵ, as long as it is in a reasonable range as discussed in Sec. VI-C. For bandwidth utilization, as the curves show the same character as the parameter ϵ changes, Fig. 5 just plots one case of the bandwidth utilization of the four COs with $\epsilon = 5$. Unlike CPU and memory resources are binding in a CO, bandwidth resource is not bound with any other types of resources in a CO. As a result, Fig. 5 shows that all bandwidth utilizations are below the optimum.



(a) CS vs. SCF

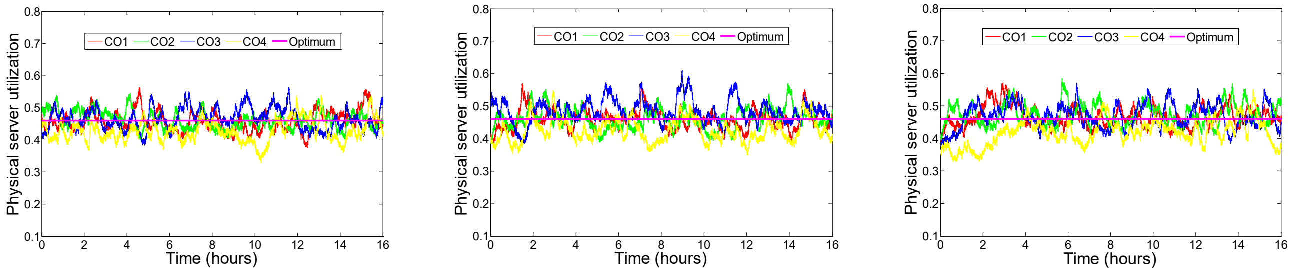
(b) CS vs. CCF

Fig. 3. Diameter comparisons of two baselines with the proposed CS algorithm.

With and without bandwidth constraint. For purpose of validating the necessity of bandwidth constraint in VNF assignment. We implement an algorithm that assigns VNFs only under the constraint of computing resources and pretends that all traffic demands of all service chains can be satisfied. In this case, the physical server utilization of CO1 with $\epsilon = 5$ is shown in Fig. 6. For comparison, the curve with bandwidth constraint and optimum curve are also shown in the figure. Without surprise, the algorithm without bandwidth constraint incurs about 17.4% (i.e., ρ increases from 0.46 to 0.54) increase in the optimum utilization level. This can be explained by the fact that a CO can pack more VNFs to its physical servers for lack of bandwidth constraint, after reaching the steady state. Then the physical server utilization is capable of exceeding the optimum as more physical servers will be non-idle for holding VNFs. In other words, the algorithm will be inaccurate in the packing of VNFs to physical servers, and thus lead to be less efficient.

Adaptiveness of the algorithm. In order to verify the adaptiveness of the shadow algorithm, especially its reactiveness to the changes of network demands, we make some adjustments in the algorithm to achieve this. Specifically, the average inter-arrival time of the incoming VNFs is changed from 2 seconds to 1.5 seconds after the 8th hour in the simulation, while the probability distribution of the type of each arriving VNF remains unchanged. This decrease in inter-arrival time means the increase of arrival rate of all types of VNFs. As expected, the physical server utilization experiences a sudden increase after the 8th hour, and the optimal utilization level (ρ) proceeds accordingly, as shown in Fig. 7. These results are sufficient to demonstrate that the algorithm is able to rapidly react to the sudden change (increase or decrease), and reach a new equilibrium level of the utilization, where the physical server utilization of all COs fluctuates up and down at the optimal level. It further confirms that the algorithm runs continuously and there is no need to rerun the algorithm from a new starting point induced by the input rate change.

Comparison with heuristic algorithm. As no previous work addressed the VNF assignment in geo-distributed NFV networks, we compare our algorithm with heuristic algorithm in [20] for assigning VMs to distributed data centers. In our case, the main idea of the heuristic algorithm is that it first partitions the VNFs into disjoint sets (e.g., belonging to the



(a) coefficient $\epsilon = 3$

(b) coefficient $\epsilon = 5$

(c) coefficient $\epsilon = 7$

Fig. 4. Physical server utilization in different COs

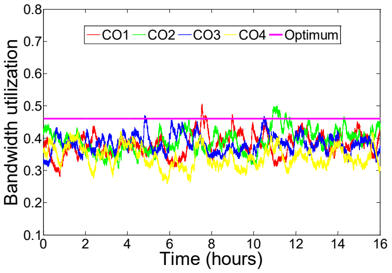


Fig. 5. Bandwidth utilization in different COs when $\epsilon = 5$.

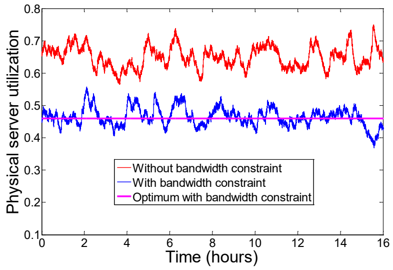


Fig. 6. Physical server utilization in CO1 with and without bandwidth constraint.

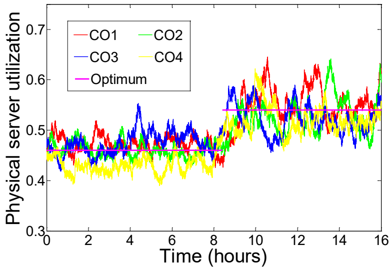


Fig. 7. Physical server utilization with changed inter-arrival time in different COs when $\epsilon = 5$.

same chain), and then schedules each partition in a same CO with maximum available bandwidth resource. Based on this, we plot the bandwidth utilization in CO1, which is obtained hourly from 1st to 10th hour, for both algorithms, as shown in Fig. 8. Bandwidth utilization in shadow algorithm can be observed to keep steady in a range of about 0.3 to 0.4, while in heuristic fluctuates heavily. The latter will consume much bandwidth when a number of VNFs are assigned and there will be much residual bandwidth after these VNFs release the resource. The reason behind this is that once VNFs in a certain partition are scheduled in a CO, they cannot be reallocated to other COs any more. However, there may be more available bandwidth resource in other COs at run time. As a result, load balancing of resource cannot be well achieved across all COs in heuristic compared with shadow algorithm.

Scalability of the algorithm. To verify the scalability of the algorithm, we enlarge the scale of CO from 4 to tens of. Meanwhile, the inter-arrival time of VNFs is adjusted from 2 seconds to 1, 0.5, and 0.25 seconds (i.e., corresponding arrival rate $\lambda = 1, 2, 4$), respectively. For each λ, we calculate the average physical server utilization (or bandwidth utilization) of the relevant number of COs in a same time. It should be noted that the utilization can also rapidly reach the optimal level with increased number of COs, and we just get the average value for illustration . From the result in Fig. 9, we can observe that the average utilization declines with the increasing number of COs, and has a positive correlation with λ. This is because more COs means more available resources and thus low resource utilization with a certain λ. Larger arrival rate λ means more VNFs waiting to be served, which will consume more resources, leading to high resource utilization. It demonstrates that the algorithm can be well scaled with larger number of COs and faster arrival rate of VNFs.

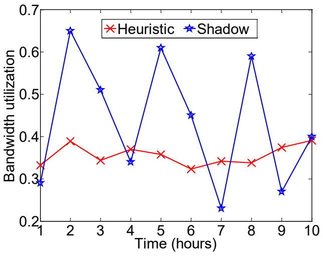


Fig. 8. Comparison of bandwidth utilization in CO1 between heuristic and shadow algorithm.

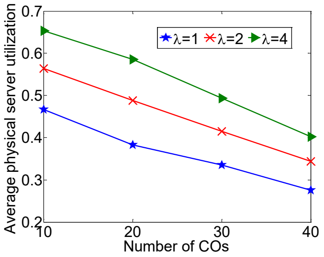


Fig. 9. Average server utilization in different number of COs with different value of λ.

D. Insights of VNF assignment vs. Shadow algorithm

For a large-scale system, in our scenario it means that tens of thousands of customers are requesting for service and a large number of VNF demands are waited to be served. In this case, it is more crucial to achieve a high level (e.g., CO level) of load balancing, rather than single physical server level, across all COs. According to the simulation results, the shadow routing based approach is an excellent choice for achieving this goal, for its outstanding advantages (the ability to handle multiple objectives and constraints) to solve such a problem. Furthermore, the incoming VNFs rates of the customers is usually dynamic, the designed algorithm for assigning VNFs should be online and response to these dynamics. While shadow algorithm overcomes these difficulties as it runs continuously.

VII. Related Work

At the moment NFV emerged, a series of industry standards, proof of concepts and solutions were proposed in the form of published white papers [1], [14], [12], [8], [9], [5], [24]. The document from IETF [14] specifically defines the resource management problem in service chaining, provides four use cases and describes a relevant framework for the problem. The problem of VNF Assignment in distributed telecos networks follows the concepts and frameworks in these white papers.

A related problem to CO selection is finding cliques and other types of subgraphs in a larger graph, which is a classical NP-hard problem. The problem of finding the largest subgraph of a given weighted undirected graph, subject to constraints on maximum degree and diameter has been studied in [25]. The designed algorithm is on the premise of a given degree, thus cannot be applied in our problem. Inspired by the datacenter selection problem in [20], we design a 2-approximation algorithm when triangle inequality holds.

To effectively utilize the resources in NFV, Gu et al. [26] design an efficient auction mechanism for dynamic resource provisioning and pricing of NFV service chains in a datacenter and prove it to be truthful and near-optimal. Yang et al. [6] present and design NFV-RT, a system that dynamically provisions resources in an NFV environment for providing timing guarantees. Kuo et al. [7] discover the crucial relation between the link and server usage, and propose an efficient algorithm for the joint problem to better utilize the network resources. Wang et al. [27] propose a novel metric called dominant load to tackle the multi-resource load balancing problem in NFV. All these works focus on a cloud environment and do not involve the problem of assigning VNFs to multiple COs. A very related work to ours is [17], which considers a shadow routing based approach to the VM placement problem. By comparison, a distinct feature of our work is that we introduce the inherent bandwidth constraint of VNF chains, which cannot be easily obtained, into the common framework [16]. To the best of our knowledge, no previous work has jointly addressed the VNF-CO and VNF-server assignment problem in geo-distributed NFV telelcos networks.

VIII. Conclusion

NFV is an emerging technology that has been widely approved by telecos service providers. In this paper, we studied the CO selection and VNF assignment (CS-VA) problem for distributed NFV in telecos infrastructure. The CS-VA problem is solved in two complementary phases. We first prove the hardness of the CS problem and then design a 2-approximation algorithm. With selected COs, we employ a shadow-routing based approach, which is asymptotically optimal, to jointly solve the VNF-CO and VNF-server problem while achieving network-wide resource load balancing. Simulations show that our CS algorithm outperforms two baselines for minimum diameter and the bandwidth constraint is well introduced into our shadow algorithm, which is more effective than existing heuristics, scalable with large number of COs and highly adaptive to the changing VNF requests.

Another important reason for geo-distributed assignment of VNFs is the reliability problem. If one CO breaks down, the other COs can avoid single point of failure. Therefore, future work will attempt to consider the reliability issues.

References

[1] "Network Functions Virtualization (NFV)," White Paper #3, Oct. 2014. [Online]. Available: http://portal.etsi.org/NFV/NFV_White_Paper3.pdf

[2] Central office. [Online]. Available: https://en.wikipedia.org/wiki/Central_office

[3] Central office lookup tool. [Online]. Available: http://www.marigolddirect.com/lists/co.php

[4] "AT&T Domain 2.0 Vision White Paper." [Online]. Available: https://www.att.com/Common/about_us/pdf/AT&T%20Domain%202.0%20Vision%20White%20Paper.pdf

[5] "Central Office Re-architected as Datacenter (CORD)," White Paper, AT&T, Open Networking Lab, Jun. 2015.

[6] Y. Li, L. Phan, and B. T. Loo, "Network functions virtualization with soft real-time guarantees," in *Proc. IEEE INFOCOM*, 2016.

[7] T. W. Kuo, B. H. Liou, K. C. J. Lin, and M. J. Tsai, "Deploying chains of virtual network functions: On the relation between link and server usage," in *Proc. IEEE INFOCOM*, April 2016, pp. 1–9.

[8] Y. Gittik, "Distributed network functions virtualization: an introduction to D-NFV," White Paper, RAD Data Communications Ltd, Mar. 2014.

[9] "Why distribution matters in NFV," White Paper, Alcatel-Lucent, Aug. 2014.

[10] Z. Zhou, F. Liu, B. Li, B. Li, H. Jin, R. Zou, and Z. Liu, "Fuel cell generation in geo-distributed cloud services: A quantitative study," in *2014 IEEE 34th International Conference on Distributed Computing Systems*, June 2014, pp. 52–61.

[11] "The cost of connectivity 2014," Oct. 2014. [Online]. Available: https://www.newamerica.org/oti/policy-papers/the-cost-of-connectivity-2014/

[12] "Network Functions Virtualization (NFV); Use Cases," Oct. 2013. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/nfv/001_099/001/01.01.01_60/gs_nfv001v010101p.pdf

[13] M. Ghaznavi, N. Shahriar, R. Ahmed, and R. Boutaba, "Service function chaining simplified," *CoRR*, vol. abs/1601.00751, 2016. [Online]. Available: http://arxiv.org/abs/1601.00751

[14] Resource management in service chaining. [Online]. Available: https://tools.ietf.org/pdf/draft-irtf-nfvrg-resource-management-service-chain-03.pdf

[15] A. L. Stolyar and T. Tezcan, "Control of systems with flexible multi-server pools: a shadow routing approach," *Queueing Systems*, vol. 66, no. 1, pp. 1–51, 2010.

[16] A. L. Stolyar, "Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm," *Queueing Systems*, vol. 50, no. 4, pp. 401–457, 2005.

[17] Y. Guo, A. L. Stolyar, and A. Walid, "Shadow-routing based dynamic algorithms for virtual machine placement in a network cloud," in *Proc. IEEE INFOCOM*, April 2013, pp. 620–628.

[18] F. Wang, R. Ling, J. Zhu, and D. Li, "Bandwidth guaranteed virtual network function placement and scaling in datacenter networks," in *2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC)*, Dec 2015, pp. 1–8.

[19] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.

[20] M. Alicherry and T. Lakshman, "Network aware resource allocation in distributed clouds," in *Proc. IEEE INFOCOM*, 2012.

[21] A. Gember, A. Krishnamurthy, S. S. John, R. Grandl, X. Gao, A. Anand, T. Benson, A. Akella, and V. Sekar, "Stratos: A network-aware orchestration layer for middleboxes in the cloud," *CoRR*, vol. abs/1305.0209, 2013.

[22] J. M. Harrison and M. J. López, "Heavy traffic resource pooling in parallelserver systems," *Queueing Systems*, vol. 33, no. 4, pp. 339–368, 1999.

[23] Gurobi optimizer reference manual. [Online]. Available: https://www.gurobi.com/documentation/6.5/refman.pdf

[24] NFV ISG PoC Proposal. [Online]. Available: http://nfvwiki.etsi.org/images/NFVPER%2814%29000011_NFV_ISG_PoC_Proposal_-_Multi-vendor_Distributed_NFV.pdf

[25] A. Dekker, H. Pérez-Rosés, G. Pineda-Villavicencio, and P. Watters, "The maximum degree & diameter-bounded subgraph and its applications," *Journal of Mathematical Modelling and Algorithms*, vol. 11, no. 3, pp. 249–268, 2012.

[26] S. Gu, Z. Li, C. Wu, and C. Huang, "An Efficient Auction Mechanism for Service Chains in The NFV Market," in *Proc. IEEE INFOCOM*, 2016.

[27] T. Wang, H. Xu, and F. Liu, "Multi-resource load balancing for virtual network functions," in *2017 IEEE 37th International Conference on Distributed Computing Systems*, June 2017.