Computer Networks 152 (2019) 12–24

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Virtual network function placement and resource optimization in NFV and edge computing enabled networks

Defang Li*, Peilin Hong, Kaiping Xue, Jianing Pei

Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei, 230027, China

ARTICLE INFO

Article history:
Received 15 July 2018
Revised 17 November 2018
Accepted 28 January 2019
Available online 31 January 2019

Keywords:
Virtual network function placement
Edge computing
Resource optimization

ABSTRACT

Network function virtualization (NFV) and edge computing (EC) are two promising and innovative technologies to accelerate 5G networks. However, placing the service function chains (SFC), each of which consists of a series of ordered virtual network functions (VNFs), into the EC enabled networks is an intractable issue and some new challenges shall arise. Firstly, EC is a hierarchical and geo-distributed structure, which will influence the form of SFCs and make the VNF placement location-related. Secondly, the data processing in EC is hierarchical too, which incurs different latency requirements. In this paper, we study the VNF placement problem considering users' SFC requests (SFCr) in NFV and EC enabled networks. Apart from the above new challenges, the implementation method and chaining of VNFs are also considered, which will raise the need of tradeoff between node resource consumption and bandwidth consumption when placing VNFs. Then the above problem is formulated as an integer linear programming (ILP) model mathematically aiming to minimize the total resource consumption, which is proven to be NP-hard. We get the optimal results when the number of SFCrs is small taking advantage of optimization solver and propose a polynomial time heuristic when the problem scale is large. Simulation results show that the resource consumption derived by our heuristic solution is near to the optimal solution and its performance is very much superior to the contrastive schemes.

1. Introduction

Network function virtualization (NFV) [1,2] has been playing a more and more important role in network designing. It allows network functions (NFs) traditionally delivered on proprietary and application-specific hardware to be realized in software, which are known as the virtual network functions (VNF), and to run on the common off-the-shelf servers [3]. Generally speaking, VNFs are interconnected through a process called Service Function Chaining (SFC) [4], which is responsible for accomplishing user's service. In the NFV enabled network, users' service requests are accomplished by the deployed SFCs. Correspondingly, users' service requests are called as SFC requests (SFCrs) in this paper.

Edge computing (EC) [5–7] is another hot research topic. It is a new form of cloud computing, which intends to reduce the service delay and is the promising technology to support new services, such as Internet-of-Things (IoT), augmented reality, optimized local content distribution, and data caching. Broadly speaking, the popular Fog computing [8,9], Cloudlet [10], and Multi Access Edge Computing (MEC) [7] recently are the different forms of EC in their own areas. EC is a hierarchical and geo-distributed architecture [5] that extends from the edge of the network to the core, including service access routers (SAR), middle servers which are usually in the form of micro datacenters (MDC) and remote cloud datacenter (CDC). By introducing the computation power closer to users, users' requests can be served timely in the MDCs, especially the delay sensitive services.

To meet the need of economic and social development in the future, the next generation (5G) telecommunications technology is charged with the capacity of handling huge volume of traffic, supporting heterogeneity of services, and ultra-reliable low latency communications (URLLC) [11]. Given the characteristics of NFV and EC, we can see that they are two promising technologies to accelerate 5G networks. However, some new challenges shall arise when deploying VNFs to meet the needs in NFV and EC enabled 5G networks.

Firstly, EC is hierarchical. Generally speaking, the delay sensitive part of a service is served in the MDC and the further data mining is usually done in the remote CDC [7,8]. So different VNFs have to be placed on both MDCs and CDC. Furthermore, the latency constraints of VNFs in MDCs and in CDC are different. Specifically, each SFCr has two latency constraints in this scenario, which are the

* Corresponding author.
E-mail addresses: ldf911@mail.ustc.edu.cn (D. Li), plhong@ustc.edu.cn (P. Hong), kpxue@ustc.edu.cn (K. Xue), jianingp@mail.ustc.edu.cn (J. Pei).

https://doi.org/10.1016/j.comnet.2019.01.036
1389-1286/

latency constraint from SAR to MDC and latency constraint from SAR to CDC. Secondly, EC is geo-distributed, which means that MDCs are distributed across a district. Users' SFCrs are from different places, and in order to get served timely, they can access the NFV and EC enabled network from the MDCs near to them. Then this makes the VNFs placement location-related, corresponding VNFs have to be placed to meet the demands in different places.

Besides the above new challenges, the instantiation method of VNFs is also taken into consideration. Traditionally, a VNF is hosted on one virtual machine (VM), and each VM has its own guest operation system (OS) and hypervisor [12], so some basic resources are needed to maintain these supports for the running of the VNF even when it is not in service, and they are called basic resource consumptions (BRCs) in our previous work [13]. In this paper, BRCs are assumed to be fixed when instantiating a VNF, and different VNF instances cannot share the BRCs for the sake of isolation among different instances. Moreover, to save BRCs, the VNF instances are all assumed to support multi-tenancy software architecture [14], which allows multiple VNF requests (VNFr) to be hosted on the same VNF software instance.

In VNF placement problem that considers SFCs, an SFC is inclined to be placed in one MDC, so that most of flows between the VNFs do not go through the network links when chaining the VNFs of the SFC, then the bandwidth consumptions are reduced. However, to achieve above results, many duplicate copies of the same type VNF have to be placed across the network, which will result in a great volume of BRCs. More BRCs mean more node resource consumptions, further, more MDCs, and more capital expenditure and operating expense (CAPEX/OPEX). Actually, BRCs can be reduced by placing less VNF instances. But less VNF instances may lead to longer flow paths of SFCs, which will bring the increase of bandwidth consumptions. So optimization is needed here to achieve a tradeoff between bandwidth consumption and node resource consumption.

Taking all above factors into consideration, we firstly formalize the form of SFCr and SFC in NFV and EC enabled networks, and then formulate the VNF placement problem as an Integer Linear Programming (ILP) model aiming to minimize the total resource consumptions. The model can be seen as a combination of facility location problem (FLP) [15–17] and multi-commodity flow problem (MCFP) [18,19], which are two well known NP-hard problems, so the ILP is NP-hard too. With the help of optimization solver, the optimal results can be figured out when the number of SFCrs is small. However, it is very time-consuming and even infeasible to get the optimal results with the increasing of the number of SFCrs. So a novel Priority based Greedy heuristic nicknamed as PG is proposed to solve the problem in polynomial time.

Our main contributions are as follows:

1. We reveal the new features of VNF placement problem in NFV and EC enabled network, which are hierarchical structure and heterogeneous latency constraints, and formalize the structure of SFCr and SFC in NFV and EC enabled network.
2. We consider the instantiation method and chaining problem of VNFs, and reveal the need of tradeoff between node resource consumption and bandwidth consumption.
3. We make a complete formulation of the problem mathematically, which is modeled as an ILP, and propose an efficient polynomial time heuristic to solve it. Through thorough evaluations, the simulation results show that the resource consumption derived by PG is near to the optimal solution when the number of SFCrs is small and outperforms the contrastive schemes apparently when the number of SFCrs is large.

The remainder of the paper is organized as follows. In Section 2, we provide a literature review. Then the structures of SFCr and SFC in NFV and EC enabled network are formalized and the problem is demonstrated (Section 3). In Section 4, we formulate the problem. After that, the heuristic solution PG is described in detail in Section 5. Section 6 shows the performance evaluation results. Finally, Section 7 concludes our work.

2. Related works

In this part, a review is made about existing works on NFV, edge computing and their expected applications in 5G, respectively.

2.1. NFV And VNF placement

NFV has been standardized by European Telecommunications Standards Institute (ETSI) and Internet Engineering Task Force (IETF) maturely [1,4,20,21], and it has been a hot research spot [3,22,23].

Plenty of researches concentrate their attention on the VNF placement problem. Mathematically, the VNF placement problems are usually formulated as integer linear programming (ILP) [24,25], mixed integer linear programming (MILP) [26], or mixed integer quadratically constraints programming (MIQCP) [18] etc., which are NP-hard in general. From the view of optimization target, the targets can be minimizing the used physical machines [27], minimizing the total resource consumption [25,28] or minimizing the total service delay [26] etc. Particularly, in [18] the authors made an analysis about all three above optimization targets.

Furthermore, many researchers consider the VNF placement problem with other features jointly. In [29], Chuan Pham et al. studied the VNF placement problem for service chains with the purpose of energy and traffic-aware cost minimization. In [24], the authors formalized the network function placement and chaining problem and proposed an ILP model to solve it. It is noteworthy that the VNF containers are geo-distributed in their model. In [30], Zilong Ye et al. considered the problem on how to jointly optimize the topology design and mapping of multiple SFCs such that the total bandwidth consumption is minimized, and they proposed a closed-loop with critical mapping feedback algorithm.

As is mentioned partly in [30], the VNF placement problem considering SFCs has its own features compared with the traditional virtual network (VN) mapping. Firstly, users request SFCs to have their demand served, and the VNFs are ordered in an SFC while the nodes in a VN do not have the order constraint. Secondly, VNFs differ from each other in terms of their functionalities and resource requirements, and their interrelationships need to be considered when placing them together, while the nodes in VN mapping problem usually do not consider the difference from each other in terms of functionality. Thirdly, the nodes belonging to the same VN usually cannot be mapped onto the same substrate node in traditional VN mapping, while in VNF placement problem, it allows the VNFs of the same SFC to be hosted on the same substrate node.

As for BRCs, in our previous work about VNF placement and resource optimization in a single datacenter [13], we have made a clear definition about BRCs, and its influence on resource optimization. In paper [31], the authors also refer some concepts that like BRCs. In their model, they claimed that a server will consume some resources to instantiate a VNF of one specified type, which is very similar to the BRCs in this paper.

Different from above mentioned works, which usually ignore the software characteristic of VNFs, we consider the multi-tenancy implementation of VNF instances. In multi-tenancy architecture, multiple VNFrs of the same type can be hosted on the same VNF instance. As a consequence, the number of VNF instances can be reduced, then BRCs are reduced. However, the problems about which VNFrs should be hosted on the same VNF instance and where to place these VNF instances need to be well addressed. These two problems are also the focuses of our work. Mathematically speaking, we involve these considerations into the problem formulations, which will incur many differences from traditional VNF placement problem.

2.2. Edge computing

In [5], M. Satyanarayanan described the emergence of edge computing, its architecture and innovation. In a broad sense, Fog computing [8,9], Cloudlet [10], and Mobile Edge Computing (MEC) [7] are the different forms of EC in their own areas. Plenty of works have been done about EC.

Academically, in [32], an orchestration architecture for the Fog computing environment is proposed, considering Fogs heterogeneity and dynamics. E. K. Markakis et al. [33] proposed a three-layer architecture by extending the Fog concept to reap the resources of end users devices and couple them into a marketplace where these resources can be traded off. In [7], the authors made a survey about the MEC on its applications, different architectures and different methods to undertake the computation offloading. In [34], X. Sun et al. proposed the latency-aware workload offloading strategy to allocate mobile user application workload in the Cloudlet network. In [35], the authors jointly considered the Cloudlet selection and delay minimization in Fog network.

From the view of industry standards, ETSI has released a series of white paper about the MEC service scenarios [36], MEC deployments in 4G and evolution towards 5G [37], and the pairing between cloud radio access network (Cloud RAN) and MEC [2] and so on.

In our paper, we consider the influence of the EC features, namely latency sensitive, hierarchical and geo-distributed, on the VNF placement problem. Specifically, we formalize the structure of SFCs in EC enabled network, and quantify above mentioned features into the mathematical formulations of the VNF placement problem.

2.3. NFV, Edge computing and 5G

A few works have been done about VNF placement in NFV and EC enabled networks.

The authors in [38] thought that the convergence of NFV, 5G, and Edge computing is unavoidable and introduced an open and converged architecture based on NFV management and orchestration (MANO) that offers uniform management of IoT services spanning the continuum from the cloud to the edge. In [11], B. Blanco et al. emphasized the role played by SDN, NFV and MEC, and analyzed the main open issues of these technologies in relation to 5G. In [39], the authors thought that softwarization is a systemic transformation and proposed an edge operating system combining SDN, NFV and cloud and edge-fog computing. In [40], the authors considered the service chaining optimization in mobile edge computing. They proposed a NFV service chaining scheme based on the popularity of VNFs. In [41], F. Callegati et al. analyzed the complexity of the SDN control plane within a cloud-based edge network implementing NFV and presented a proof-of-concept implementation with the Mininet emulation platform. In [42], F. B. Jemaa et al. studied the VNF placement and provisioning optimization strategies over an edge-central carrier cloud infrastructure, considering the Quality of Service (QoS) requirements (i.e., response time, latency constraints and real-time requirements).

According to the white paper on NFV priorities for 5G by ETSI [43], users' requirements have significantly different characteristics

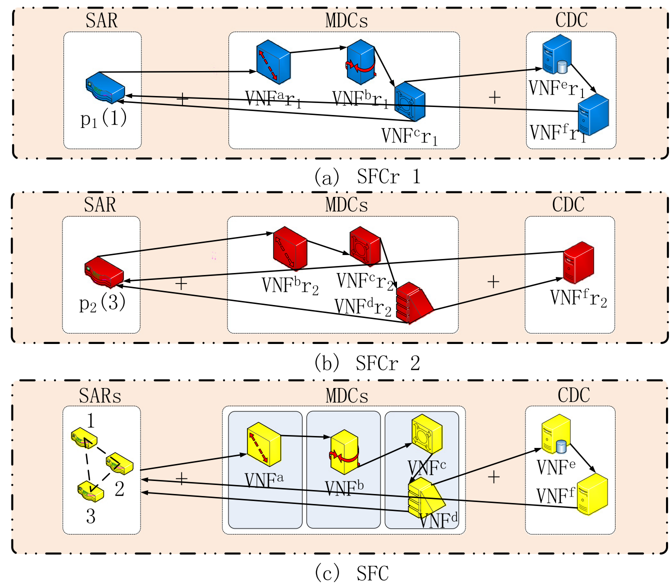


Fig. 1. SFCr and SFC in NFV and EC enabled environment.

and combinations of high scalability, ultra-low latency and high volume of traffic. To handle these heterogeneous QoS requirements and low latency, NFV and EC are believed to be the key technology enablers. Also they have made the efforts on the deployment of MEC on NFV environment [44].

Compared with existing works that mainly focus on the architecture converging of NFV, EC and 5G, we concentrate on the VNF placement problem in the emerging architecture, and the solution designing aiming to improve the network resource utilization.

In summary, our work researches the VNF placement problem in EC enabled network. To improve the network resource utilization, the software characteristics of VNFs are considered, which involve the BRCs and multi-tenancy implementation of VNFs. To solve the problem, we make a complete formulation about the problem (ILP model), and propose a priority based greedy solution, which has polynomial time complexity.

3. Problem statement

In this section, we introduce the structure of SFCr, SFC and substrate topology in NFV and EC enabled network firstly, and then make a description about our problem.

3.1. SFCr and SFC in NFV and EC enabled environment

In the NFV and EC enabled networks, users' SFCrs are hierarchical. Fig. 1(a) and (b) show two SFCrs. As we can see, each SFCr consists of three parts, including an SAR, part of SFCr in MDCs that are responsible for the latency sensitive part of the service, and part of SFCr in the CDC that deal with further data processing. Moreover, the back-flow of each SFCr includes two parts, which are the flow returning back from MDCs and flow returning back from CDCs. Corresponding to the VNFs in SFC, the nodes in SFCr are called VNF requests (VNFr). $VNF^i r$, $i = a, b, c, d, e, f$ in Fig. 1(b) and (c) indicates different kinds of VNFrs, respectively.

Owing to the characteristic of SFCr, the structure of SFC in NFV and EC enabled environment is hierarchical too. As Fig. 1(c) shows, SFC consists of three parts too, namely SARs, part of SFC in MDCs, and part of SFC in CDC. It is worth noting that the VNFs that should be hosted in MDCs may be distributed in several different ones. VNF^i, $i = a, b, c, d, e, f$ in Fig. 1(c) indicates different kinds of VNFs, respectively.

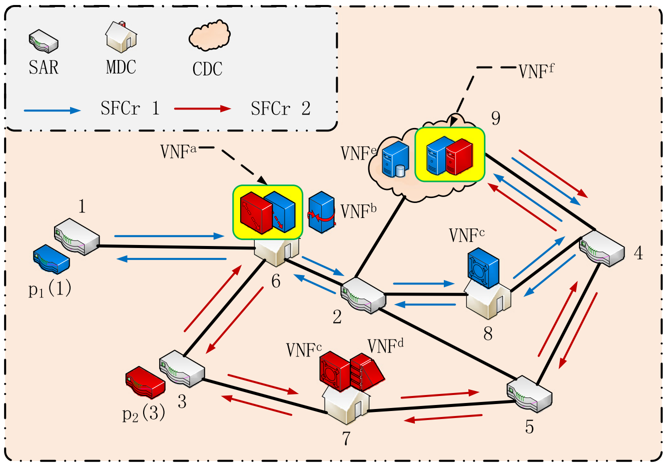


Fig. 2. Problem description.

In this paper, we consider mapping a series of SFCrs to MDCs and CDCs in the NFV and EC enabled networks and placing the related VNFs to accomplish the services. A 6-tuple $(p_\gamma, \Phi_\gamma, \Psi_\gamma, E^v_\gamma, D^M_\gamma, D^C_\gamma)$ is used to indicate an SFCr. p_γ, Φ_γ, and Ψ_γ indicates the SAR of SFCr γ, part of SFCr γ in MDCs, and part of SFCr γ in CDCs, respectively. E^v_γ indicates the logical links between VNFrs of SFCr γ. For example, (VNF^{a}r$_1$, VNF^{b}r$_1$) in Fig. 1(a) is one logical link in SFCr 1. For each SFCr γ, there are two latency constraints, one is the tolerated propagation latency from SAR to MDC, D^M_γ, which is to ensure the user can get served timely; the other one is the entire tolerated propagation latency from SAR to CDC, D^C_γ.

The SAR of each SFCr is corresponding to one specified SAR in the substrate network because of the geographical cause, e.g. SAR p_1 of SFCr 1 is attached to SAR 1 in Fig. 1(c), and SAR p_2 of SFCr 2 is attached to SAR 3.

3.2. Substrate topology

The substrate topology is a hierarchical and geo-distributed structure, including SARs, MDCs and CDCs, which is modeled as an undirected graph. Fig. 2 shows an instance of the substrate topology. A 4-tuple (P, R, G, E^s) is used to indicate the substrate topology, in which P indicates the set of all SARs, R indicates the set of all MDCs, G indicates the set of all CDCs, and E^s indicates the set of all substrate links. Without loss of generality, n^s_u and n^s_v are used as the symbols of two nodes in the substrate network respectively. Then (n^s_u, n^s_v) indicates a substrate link. Each link has a propagation delay, $d_{n^s_u,n^s_v}$.

In the NFV and EC enabled environment, users access service through the SARs, and then the flows of their SFCrs are directed to the MDCs and CDCs through the network.

3.3. Multi-tenancy and NFV

Multi-tenancy is a software architecture principle in the realm of Software as a Service (SaaS) business model [14], which allows multiple tenants to share the same software instance. Compared with the single-tenant architecture, in which each tenant gets his own instance of application, the multi-tenant architecture can lead to higher resource utilization, lower service price, and more efficient management for the cloud service providers [45]. In [46], they thought that mobile network operators could share VNFs while maintaining separate logical data and control planes, and a framework is evaluated utilizing open-source tools and virtual tenant networks techniques.

In this paper, multi-tenancy technology is assumed to be applied to the implementation of VNFs, which means that one instance of VNF can host multiple VNFr of the same type. Because BRCs occur when instantiating VNFs, so the less VNF instances, the less BRCs. And then the total node resource consumptions decrease.

3.4. Problem description

Fig. 2 shows the mapping results of the two SFCrs in Fig. 1.

SFCr 1 and SFCr 2 access the service from SAR 1 and SAR 3, respectively. The blue lines describe the flow directions of SFCr 1 and the red lines describe the flow directions of SFCr 2. It is assumed that VNF^{a}r$_1$, VNF^{a}r$_2$ and VNF^{b}r$_1$ are mapped on MDC 6; VNF^{c}r$_2$ and VNF^{d}r$_2$ are mapped on MDC 7; VNF^{c}r$_1$ is mapped on MDC 8; VNF^{e}r$_1$, VNF^{f}r$_1$, and VNF^{f}r$_2$ are all mapped on CDC 9. As a consequence, an instance of VNF^b has to be placed on MDC 6; an instance of VNF^c and an instance of VNF^d on MDC 7; an instance of VNF^c on MDC 8; an instance of VNF^e on CDC 9. However, only one instance of VNF^a is needed rather than two on MDC 6, and one instance of VNF^f on CDC 9, owing to the multi-tenancy technology.

Based on above results, the consumed BRCs will be 7 shares, namely 2 shares in MDC 6; 2 shares in MDC 7; 1 share in MDC 8 and 2 shares in CDC 9. If the mapping position of VNF^{c}r$_1$ changes from MDC 8 to MDC 7, then MDC 8 will not need to place the instance of VNF^c. As a result, 1 share of BRCs will be saved and MDC 8 is free, which means that a great of CAPEX/OPEX will be saved. However, the flow path of SFCr 1 may change from $1\to6\to2\to8\to4\to9\to4\to8\to2\to6\to1$ to $1\to6\to3\to7\to5\to4\to9\to4\to5\to7\to3\to6\to1$, which can be seen that 2 more hops of link need to be passed through, so bandwidth consumptions increase.

From the above simple case, we can figure out that less instances of VNFs will reduce the node resource consumption, however, bandwidth consumption may increase because of the longer service path. So it needs a balance between the number of VNF instances, i.e, node resource consumption and the bandwidth consumption. Moreover, the latency constraints must not be violated when mapping SFCrs to the network.

In summary, we mean to design a resource efficient mapping scheme for a set of SFCrs to accommodate them into the NFV and EC enabled network, then place the corresponding VNFs to accomplish the services in this paper.

4. Problem formulation

4.1. FLP And MCFP

In facility location problem (FLP) [15–17], we have a set of clients and a set of facilities, then there needs a way to connect each client to a facility that minimizes the total cost. The cost usually contains two parts, which are the connection costs and facility costs. In our model, MDCs can be treated as the facilities and SFCrs as the clients, bandwidth consumption as the connection cost, and node resource consumption as the facility cost. Then the VNF placement problem in NFV and EC enabled networks can be seen as an FLP. From another view, there are massive network flows in the SFCrs, and all these flows need to be mapped into the network, so the mapping process of flows in the VNF placement problem can be seen as a multi-commodity flow problem (MCFP) [18,19].

4.2. Mathematical formulation

In this subsection, the VNF placement problem in NFV and EC enabled environment is formulated as an ILP model, which is a

Table 1
Notations.

Symbols	Description
Topology related	
(P, R, G, E^s)	The 4-tuple indicating the substrate network. P indicates the set of SARs, R indicates the set of MDCs, G indicates the set of CDCs, and E^s indicates all the substrate links.
n_u^s, n_v^s	Two substrate nodes in the substrate network.
(n_u^s, n_v^s)	The substrate link between n_u^s and n_v^s.
$(p_\gamma, \Phi_\gamma, \Psi_\gamma, E_\gamma^v, D_\gamma^M, D_\gamma^C)$	The 6-tuple indicating an SFCr. p_γ indicates the SAR of SFCr γ, Φ_γ indicates the part of SFCr γ that should be in MDCs. Ψ_γ indicates the part of SFCr γ that should be in CDC. E_γ^v indicates the logical links between VNFrs of SFCr γ. D_γ^M is the tolerated propagation latency from SAR to MDC, and D_γ^C is the entire tolerated propagation latency of SFCr γ.
n_i^v, n_j^v	Two nodes in an SFCr.
(n_i^v, n_j^v)	The logical link between n_i^v and n_j^v, $(n_i^v, n_j^v) \in E_\gamma^v$.
Variables	
Γ	The SFCr set.
Λ	The specifies of all the VNFs, λ is an element of Λ.
ρ	The cost of activating a new MDC.
$\mathrm{brc}_\lambda^{\mathrm{cpu}}$, $\mathrm{brc}_\lambda^{\mathrm{mem}}$	The CPU and memory BRCs when instantiating a new VNF λ on one MDC or CDC, respectively.
b_{γ, n_i^v, n_j^v}	The bandwidth consumption of logical link (n_i^v, n_j^v) in SFCr γ.
$\mathrm{cpu}_{\gamma, n_i^v}$, $\mathrm{mem}_{\gamma, n_i^v}$	The CPU and memory consumption of VNFr n_i^v in SFCr γ, respectively.
$C_{n_u^s}^{\mathrm{cpu}}$, $C_{n_u^s}^{\mathrm{mem}}$	The CPU and memory capacity of substrate node n_u^s, respectively.
$C_{(n_u^s, n_v^s)}^{\mathrm{link}}$	The link capacity of substrate link (n_u^s, n_v^s).
$d_{n_u^s, n_v^s}$	The propagation delay of link (n_u^s, n_v^s).
Binaries	
x_{γ, n_i^v, n_u^s}	Whether n_i^v in SFCr γ is mapped on substrate node n_u^s.
$l_{\gamma, n_i^v, \lambda}$	Whether VNFr n_i^v in SFCr γ demands VNF λ.
$y_{\gamma, n_i^v, n_j^v, n_u^s, n_v^s}$	Whether the mapping path of logical link (n_i^v, n_j^v) of SFCr γ goes through the substrate link (n_u^s, n_v^s).
z_{λ, n_u^s}	Whether an instance of VNF λ is placed on substrate node n_u^s, $\lambda \in \Lambda$, $n_u^s \in R\&G$.
$h_{n_u^s}$	Whether MDC n_u^s is activated, $n_u^s \in R$.

combination of FLP and MCFP. The main notations are listed in Table. 1. There are similar formulations in [13,24,25].

Firstly, one node (a VNFr or the SAR) in an SFCr must be mapped on one and only one node in the substrate network:

$$\sum_{u=0}^{|P|+|R|+|G|-1} x_{\gamma, n_i^v, n_u^s} = 1, n_u^s \in P \cup R \cup G, \gamma \in \Gamma, n_i^v \in p_\gamma \cup \Phi_\gamma \cup \Psi_\gamma. \tag{1}$$

x_{γ, n_i^v, n_u^s} indicates whether n_i^v in SFCr γ is mapped on substrate node n_u^s. $|P|$, $|R|$, $|G|$ are the total number of SARs, MDCs and CDCs, respectively.

Specially, it is stipulated that the SAR of each SFCr has to be mapped on one specified SAR owing to the geographical cause. It is the consistent one-to-one match relationship, so if n_i^v is an SAR (p_γ):

$$x_{\gamma, n_i^v, n_u^s} = \begin{cases} 1, & n_u^s \textit{ is the attachment of } p_\gamma, \ \gamma \in \Gamma, \\ 0 & \textit{otherwise}. \end{cases} \tag{2}$$

Besides, VNFrs in Φ_γ can only be mapped on MDCs, and VNFrs in Ψ_γ can only be mapped on CDCs for each SFCr γ. So there come the following constraints:

$$x_{\gamma, n_i^v, n_u^s} = \begin{cases} \{0, 1\}, & n_i^v \in \Phi_\gamma \textit{ and } n_u^s \in R, \\ \{0, 1\}, & n_i^v \in \Psi_\gamma \textit{ and } n_u^s \in G, \\ 0, \textit{otherwise}. \end{cases} \tag{3}$$

Then BRCs related variable, z_{λ, n_u^s}, and MDC related variable, $h_{n_u^s}$, are introduced in. Firstly:

$$z_{\lambda, n_u^s} = \begin{cases} 1, & \sum_{\gamma=0}^{|\Gamma|-1} \sum_{i=0}^{|\Phi_\gamma|+|\Psi_\gamma|-1} x_{\gamma, n_i^v, n_u^s} \cdot l_{\gamma, n_i^v, \lambda} \geq 1, \\ 0 & \textit{otherwise}. \end{cases} \tag{4}$$

where $n_u^s \in R \cup G$, $|\Gamma|$ is the number of total SFCrs and $|\Phi_\gamma| + |\Psi_\gamma|$ is the number of VNFrs in SFCr γ. Furthermore, l_{γ, n_i^v} is not variable, because the type of one VNFr is known beforehand. So $\sum_{\gamma=0}^{|\Gamma|-1} \sum_{i=0}^{|\Phi_\gamma|+|\Psi_\gamma|-1} x_{\gamma, n_i^v, n_u^s} \cdot l_{\gamma, n_i^v, \lambda} \geq 1$ is not the quadratic constraint. Eq. (4) is a conditional constraint. However, referring to the theorems in [47], we can utilize Eq. (5) to linearize Eq. (4):

$$0 < 1 - \sum_{\gamma=0}^{|\Gamma|-1} \sum_{i=0}^{|\Phi_\gamma|+|\Psi_\gamma|-1} x_{\gamma, n_i^v, n_u^s} \cdot l_{\gamma, n_i^v, \lambda} + \mathbb{K} \cdot z_{\lambda, n_u^s} < \mathbb{K} - 1 \tag{5}$$

$\mathbb{K}$ is a big constant, and usually it determines the solution space of Eq. (5). The bigger the $\mathbb{K}$ is, the larger the solution space will be. Here $\mathbb{K}$ is set to be $\sum_{\gamma=0}^{|\Gamma|-1}(|\Phi_\gamma| + |\Psi_\gamma| + 1)$ in our model, which is the number of all VNFrs, to make Eq. (5) effective while narrowing down the solution space.

$x_{\gamma, n_i^v, n_u^s} \cdot l_{\gamma, n_i^v, \lambda}$ indicates that whether SFCr γ has a VNFr demanding for VNF λ and the VNFr is hosted on substrate node n_u^s, if yes, $x_{\gamma, n_i^v, n_u^s} \cdot l_{\gamma, n_i^v, \lambda} = 1$. So Eq. (4) means that if more than one (including one) VNFrs demanding for VNF λ are mapped on n_u^s (MDC or CDC), then the corresponding MDC or CDC must place an instance of VNF λ.

$$h_{n_u^s} = \begin{cases} 1, & \sum_{\gamma=0}^{|\Gamma|-1} \sum_{i=0}^{|\Phi_\gamma|-1} x_{\gamma, n_i^v, n_u^s} \geq 1, n_u^s \in R. \\ 0 & \textit{otherwise}. \end{cases} \tag{6}$$

Eq. (6) means that if there is more than one VNFr mapped on one MDC, then the MDC has to be activated, and we can utilize the similar method as that for Eq. (4) to linearize Eq. (6). As for the CDCs, they are assumed to be always in operation.

Nextly, the resource constraints are considered. We assume that the network is un-utilized initially, so the maximum available resource on each node or link equals the resource capacity of that node or link.

Firstly, the total CPU and memory consumption by the VNFrs on one MDC cannot exceed CPU and memory capacity of that MDC:

$$\sum_{\gamma=0}^{|\Gamma|-1} \sum_{i=0}^{|\Phi_\gamma|-1} \mathrm{cpu}_{\gamma, n_i^v} \cdot x_{\gamma, n_i^v, n_u^s} + \sum_{\lambda=0}^{\Lambda-1} \mathrm{brc}_\lambda^{\mathrm{cpu}} \cdot z_{\lambda, n_u^s} \leq C_{n_u^s}^{\mathrm{cpu}}, n_u^s \in R. \tag{7}$$

$$\sum_{\gamma=0}^{|\Gamma|-1}\sum_{i=0}^{|\Phi_\gamma|-1}\mathrm{mem}_{\gamma,n_i^v}\cdot x_{\gamma,n_i^v,n_u^s}+\sum_{\lambda=0}^{\Lambda-1}\mathrm{brc}_\lambda^{\mathrm{mem}}\cdot z_{\lambda,n_u^s}\le C_{n_u^s}^{\mathrm{mem}},\ n_u^s\in R. \tag{8}$$

In Eq. (7), $\sum_{\gamma=0}^{|\Gamma|-1}\sum_{i=0}^{|\Phi_\gamma|-1}\mathrm{cpu}_{\gamma,n_i^v}\cdot x_{\gamma,n_i^v,n_u^s}$ indicates the total CPU consumption by the VNFrs mapped on MDC n_u^s, $\sum_{\lambda=0}^{|\Lambda|-1}\mathrm{brc}_\lambda^{\mathrm{cpu}}\cdot z_{\lambda,n_u^s}$ indicates the total CPU BRCs when MDC n_u^s instantiates the related VNFs. We do not consider the resource consumption on SARs and assume that the resources on CDCs are always enough, so there are no constraint equations about SARs and CDCs.

Then the link variable is as follows:

$$y_{\gamma,n_i^v,n_j^v,n_u^s,n_v^s}=\begin{cases}1, & condition\ is\ satisfied,\\ 0 & otherwise.\end{cases}. \tag{9}$$

The *condition* in Eq. (9) is that the mapping path of logical link (n_i^v, n_j^v) in SFCr γ goes through substrate link (n_u^s, n_v^s), $(n_u^s, n_v^s)\in E^s$.

Generally speaking, each logical link in one SFCr is corresponding to a path in the substrate network. However, in the VNF placement problem, two different nodes in one SFCr can be mapped on the same substrate node. In other words, the flow of logical link (n_i^v, n_j^v) in SFCr γ may be limited in one MDC or CDC, not going through any external substrate link. So we have the following constraint:

$$\sum_{(n_u^s,n_v^s)\in E^s} y_{\gamma,n_i^v,n_j^v,n_u^s,n_v^s}\ge 0,\ \ (n_i^v,n_j^v)\in E_\gamma^v,\ \ \gamma\in\Gamma. \tag{10}$$

0 indicates that the flow of corresponding logical link is limited in one MDC or CDC.

For each link in the substrate network, the following capacity constraint must be satisfied:

$$\sum_{\gamma=0}^{|\Gamma|-1}\sum_{(n_i^v,n_j^v)\in E_\gamma^v} b_{\gamma,n_i^v,n_j^v}\cdot y_{\gamma,n_i^v,n_j^v,n_u^s,n_v^s}\le C_{(n_u^s,n_v^s)}^{\mathrm{link}},\ (n_u^s,n_v^s)\in E^s,\ \ \gamma\in\Gamma. \tag{11}$$

Subsequently, the latency constraints of each SFCr are defined. In the NFV and EC enabled networks, time sensitive services are handled in MDCs, and the remote CDC are responsible for statistic analysis generally [8,38]. So there are two latency constraints for each SFCr:

$$\sum_{i=0}^{|\Phi_\gamma|-1}\sum_{j=0}^{|\Phi_\gamma|-1}\sum_{(n_u^s,n_v^s)\in E^s} d_{n_u^s,n_v^s}\cdot y_{\gamma,n_i^v,n_j^v,n_u^s,n_v^s}\le D_\gamma^M,\ (n_i^v,n_j^v)\in E_\gamma^v,\ \ \gamma\in\Gamma. \tag{12}$$

$$\sum_{i=0}^{|\Phi_\gamma|+|\Psi_\gamma|-1}\sum_{j=0}^{|\Phi_\gamma|+|\Psi_\gamma|-1}\sum_{(n_u^s,n_v^s)\in E^s} d_{n_u^s,n_v^s}\cdot y_{\gamma,n_i^v,n_j^v,n_u^s,n_v^s}\le D_\gamma^C, \tag{13}$$

$(n_i^v,n_j^v)\in E_\gamma^v,\ \ \gamma\in\Gamma.$

Eq. (12) ensures that users can get served timely, and Eq. (13) ensures that the total latency should be restricted in a certain threshold. Only the propagation delay is considered in this paper, but it is feasible to add the processing delay and queuing delay on each VNF in the formulations.

Last but not the least, for each SFCr, the following flow constraints must be satisfied [13]:

$$\sum_{n_v^s}^{P\cup R\cup G} y_{\gamma,n_i^v,n_j^v,n_u^s,n_v^s}\in\{0,1\},\ \ \gamma\in\Gamma,\ (n_u^s,n_v^s)\in E^s,\ \ (n_i^v,n_j^v)\in E_\gamma^v. \tag{14}$$

$$\sum_{n_v^s}^{P\cup R\cup G} y_{\gamma,n_i^v,n_j^v,n_v^s,n_u^s}\in\{0,1\},\ \ \gamma\in\Gamma,\ (n_v^s,n_u^s)\in E^s,\ \ (n_i^v,n_j^v)\in E_\gamma^v. \tag{15}$$

$$\sum_{n_v^s}^{P\cup R\cup G} y_{\gamma,n_i^v,n_j^v,n_u^s,n_v^s}-\sum_{n_v^s}^{P\cup R\cup G} y_{\gamma,n_i^v,n_j^v,n_v^s,n_u^s}=x_{\gamma,n_i^v,n_u^s}-x_{\gamma,n_j^v,n_u^s},\ \ \gamma\in\Gamma,$$
$$(n_u^s,n_v^s)\in E^s,\ \left(n_i^v,n_j^v\right)\in E_\gamma^v. \tag{16}$$

Eq. (14) indicates whether a logical link is mapped on one of the substrate links that leave out node n_u^s, and Eq. (15) indicates whether the logical link is mapped on one of the substrate links that go in node n_u^s. The two equations also ensure that one logical link can only be mapped on a single path. Eq. (16) ensures that the path in the substrate network is consistent for a logical link. More specifically, if n_i^v and n_j^v are hosted on two different substrate nodes (MDCs), then:

$$x_{\gamma,n_i^v,n_u^s}-x_{\gamma,n_j^v,n_u^s}=\begin{cases}1, & n_i^v\ is\ on\ n_u^s,\\ -1, & n_j^v\ is\ on\ n_u^s,\\ 0, & otherwise.\end{cases}. \tag{17}$$

If n_i^v and n_j^v are mapped on the same substrate node (MDC), then there will always be $x_{\gamma,n_i^v,n_u^s}-x_{\gamma,n_j^v,n_u^s}=0$.

After all the above constraints, the total CPU resource consumptions can be figured out:

$$\mathrm{CPU}_c=\sum_{\gamma=0}^{|\Gamma|-1}\sum_{i=0}^{|\Phi_\gamma|-1}\sum_{u=0}^{|R|+|G|-1}\mathrm{cpu}_{\gamma,n_i^v}\cdot x_{\gamma,n_i^v,n_u^s} \tag{18}$$

The total memory resource consumptions are:

$$\mathrm{MEM}_c=\sum_{\gamma=0}^{|\Gamma|-1}\sum_{i=0}^{|\Phi_\gamma|-1}\sum_{u=0}^{|R|+|G|-1}\mathrm{mem}_{\gamma,n_i^v}\cdot x_{\gamma,n_i^v,n_u^s} \tag{19}$$

The total CPU BRCs are:

$$\mathrm{BRC}_c^{\mathrm{cpu}}=\sum_{\lambda=0}^{\Lambda-1}\sum_{u=0}^{|R|+|G|-1}\mathrm{brc}_\lambda^{\mathrm{cpu}}\cdot z_{\lambda,n_u^s} \tag{20}$$

The total memory BRCs are:

$$\mathrm{BRC}_c^{\mathrm{mem}}=\sum_{\lambda=0}^{\Lambda-1}\sum_{u=0}^{|R|+|G|-1}\mathrm{brc}_\lambda^{\mathrm{mem}}\cdot z_{\lambda,n_u^s} \tag{21}$$

The total bandwidth consumptions are:

$$\mathrm{Band}_c=\sum_{\gamma=0}^{|\Gamma|-1}\sum_{(n_i^v,n_j^v)\in F_\gamma^v}\sum_{(n_u^s,n_v^s)\in E^s} b_{\gamma,n_i^v,n_j^v}\cdot y_{\gamma,n_i^v,n_j^v,n_u^s,n_v^s} \tag{22}$$

The total cost of activating MDCs is:

$$\mathrm{MDC}_c=\sum_{u=0}^{|R|-1}\rho\cdot h_{n_u^s} \tag{23}$$

In the mapping process, MDCs should be used as less as possible. Because an activated MDC is associated with a series of costs, including building, power supply, hardware equipments, cooling system etc. In order to be compatible with the resource cost, a value ρ is set to indicate the cost of activating an MDC. ρ is far greater than the other cost to ensure that MDCs are activated as less as possible.

Then the total substrate resource cost is:

$$T_c = \alpha * (\text{CPU}_c + \text{BRC}_c^{\text{cpu}}) + \beta * (\text{MEM}_c + \text{BRC}_c^{\text{mem}}) + \rho * \text{Band}_c + \varrho * \text{MDC}_c \tag{24}$$

α, β, ρ and ϱ are the weighted factors to balance different costs. The optimization target is to minimize T_c.

For the conditional constraints in above equations, namely, Eqs. (4) and (6), we can utilize the theorems in [47] to have these equations linearized. Then all equations above can be formulated in an ILP model. As stated above, the VNF placement problem in NFV and EC enabled environment can be seen as a combination of FLP and MCFP, which are two well known NP-hard problems. So the above ILP model is NP-hard. With the help of existing optimization solvers, like Gurobi [48], the optimal results of the ILP model can be worked out when the number of SFCrs is small. However, it is infeasible to achieve the optimal results in foreseeable time when the number of SFCrs is large. Thus a solution that solves the NP-hard problem in polynomial time is in need.

5. Proposed heuristic

In this section, our polynomial time heuristic will be described in detail.

In our system model, there is only one CDC. Because the resources in the CDC are assumed to be infinite. As a result, the dominated difference between multiple CDCs model and single CDC model is the optimization of load balancing among the links from MDCs to CDCs, which is out of scope of our problem and needs another paper to address it in detail.

To save the bandwidth, it is better to map Φ_γ of SFCr γ as a whole to one MDC and map Ψ_γ of SFCr γ as a whole to the CDC. Because the flows inner Φ_γ or Ψ_γ are restricted in the MDCs or CDC, not consuming the external bandwidth resources as a result.

Based on the above discussions, a scheme is designed to map the SFCrs to MDCs and the CDC. However, the above mapping process leads to a result that many duplicate copies of the same type VNF are placed across the distributed MDCs, which results in a great volume of BRCs and plenty of activated MDCs subsequently. Then an adjustment process is made to merge together the VNFrs demanding for the same type of VNF. By merging the VNFrs demanding for the same type of VNF together, the multi-tenancy technology can be utilized to implement the VNFs. By assigning multiple VNFrs of the same type on the same VNF instance, the number of duplicate VNF copies will decrease, and then BRCs are reduced. Nevertheless, the bandwidth consumption may increase in the process. So a balance needs to be made between the node resource consumption and bandwidth consumption in order to minimize the total cost.

We propose a Priority based Greedy solution, nicknamed as PG, consisting of a priority based SFCr mapping algorithm and a further VNFr merging algorithm. In the next two subsections, the SFCr mapping and VNFr merging algorithm are described in detail, respectively.

5.1. SFCr Mapping

Algorithm 1 gives the processes of SFCr mapping.

Because the SAR of each SFCr is corresponding to one particular SAR in the network, so the mapping of SFCrs are location-related. Thus the SFCrs are classified into $|R|$ clusters based on the SAR of each SFCr firstly (Procedure 1). In Procedure 1, $|R|$ clusters are created, and the MDC is the center of each cluster (lines 1–3 in Procedure 1). Then for each SFCr, the total propagation delay (TPD) from its SAR to each MDC is calculated respectively using Shortest Path Algorithm (SPA) (lines 4–6 in Procedure 1). If the TPD of the SFCr is smaller than its latency requirement from SAR to MDC (D_γ^M), then the SFCr can be classified into the corresponding cluster (lines 7–8 in Procedure 1).

For an SFCr, belonging to a cluster means that it can be mapped on the MDC in that cluster. Obviously, there may be many candidate MDCs for one SFCr. Thus, given the number of candidate MDCs, the SFCrs can be classified into the following two types:

Algorithm 1 SFCr Mapping Algorithm.

Input: Set of MDCs: R, Status of the network: Ω_0;
Output: Set of used MDCs: R_1, Status of the network: Ω_1;
Classify the SFCrs into $|R|$ clusters. $\leftarrow$ **Procedure 1**.
Figure out the processing priority of each cluster.
for i from 0 to $|R|-1$ **do**
  Classify the SFCrs in ζ_i into P-SFCrs and R-SFCrs, ζ_i is the i_{th} cluster.
  Map the P-SFCrs belonging to ζ_i to the MDC in ζ_i.
  Place related VNFs in the MDC.
  Establish paths between the SARs, MDCs and CDC.
  Remove all the P-SFCrs from ζ_i.
  while ζ_i is not empty **do**
    Pick out the R-SFCr(s) that has(have) the smallest TPD between its(their) SAR(s) and the MDC in ζ_i, symbolled as ξ.
    if $|\xi| > 1$ **then**
      Choose the R-SFCr whose demanding VNFs have the least differences with the VNFs already in the MDC as the one to be mapped.
    else
      The single R-SFCr in ξ is chosen as the one to be mapped.
    end if
    if MDC in ζ_i can host the chosen R-SFCr **then**
      Map the chosen R-SFCr to the MDC in ζ_i.
      Establish paths between the SAR, MDC and CDC.
      if Need to place new VNFs **then**
        Place new VNFs.
      end if
    else
      Leave the chosen R-SFCr to another cluster that it can be mapped, and set its priority the same as the P-SFCrs in that cluster.
    end if
    Remove the chosen R-SFCr from ζ_i.
  end while
end for

Procedure 1 Cluster SFCrs.

1: **for** i from 0 to $|R|-1$ **do**
2:   Create an empty cluster, ζ_i.
3: **end for**
4: **for** SFCr γ in Γ **do**
5:   **for** i from 0 to $|R|-1$ **do**
6:     Calculate the TPD from the SAR of SFCr γ to the MDC in ζ_i.
7:     **if** TPD $< D_\gamma^M$ **then**
8:       Add SFCr γ to cluster ζ_i.
9:     **else**
10:       Continue.
11:     **end if**
12:   **end for**
13: **end for**

- “Poor” SFCrs (P-SFCr): For those SFCrs that have only one candidate MDC to be mapped into, we call them the “Poor” SFCrs (P-SFCr). For those VNFrs in the P-SFCr, they are called P-VNFrs.
- “Rich” SFCrs (R-SFCr): For those SFCrs that have more than one candidate MDCs to be mapped into, we call them the “Rich” SFCrs (R-SFCr). For those VNFrs in the R-SFCr, they are called R-VNFrs.

After the clustering, we calculate the number of P-SFCrs in each cluster. Then the processing priority of each cluster is determined based on the number of P-SFCrs in it (line 2 in Algorithm 1). The more P-SFCrs, the higher processing priority of the corresponding cluster. For the clusters that have the same number of P-SFCrs, the resource consumptions by the total P-SFCrs in them are calculated respectively, and the cluster with larger resource consumption is given higher processing priority. Then we indicate the priority of each cluster with numbers. 0 indicates the highest priority, then priority decreases with the increasing of numbers. When the processing sequence is determined, the SFCrs then are mapped into the MDCs and CDC one cluster by one cluster (lines 3–27 in Algorithm 1).

During the process of mapping SFCrs to the corresponding MDC, we need to determine a proper mapping sequence to make a better resource utilization of the MDC. Obviously, the priority of P-SFCrs is higher than that of R-SFCrs, because there are no other MDCs to host the P-SFCrs except the current one. For the P-SFCrs, all of them must be mapped to the current MDC firstly (line 5 in Algorithm 1). After the mapping of all the P-SFCrs, the MDC in the cluster needs to be placed some related VNFs to serve the P-SFCrs and the flow paths between SARs, MDCs and CDC are established using SPA (line 7 in Algorithm 1).

Subsequently, the R-SFCrs need to be mapped to the current MDC. Firstly, R-SFCr that has the smallest TPD between its SAR and the current MDC is picked out (line 10 in Algorithm 1). In this way, the R-SFCr can be mapped to its nearest MDC aiming to consume as less bandwidth as possible. For the R-SFCrs that have the same smallest TPD (lines 11–12 in Algorithm 1), the VNFrs in them are compared with the VNFs that already have been placed in the MDC respectively. Then the R-SFCr whose needed VNFs has the least differences with the VNFs already in the MDC (line 12 in Algorithm 1) is chosen to be mapped. In this way, we try not to introduce new types of VNFs in the MDC as much as possible, so as to instantiate as less new VNF instances as possible and the less BRCs as a consequence. For the chosen R-SFCr, it is settled down (lines 16–21 in Algorithm 1) and the next R-SFCr is picked out in the same process. For the R-SFCr that cannot be mapped on the current MDC owing to the shortage of resources, it is left to another cluster can be mapped, and its priority is set the same as that of P-SFCrs in that cluster (line 23 Algorithm 1), trying to map the R-SFCr as soon as possible.

5.2. VNFr Merging

Based on above SFCr mapping results, the bandwidth consumptions are the least. Nevertheless, there are a lot duplicate copies of each type of VNF in the network, so the volume of BRCs is huge. We can merge together the distributed VNFrs that demand for the same type of VNF, and then these VNFrs can share the same instance of VNF utilizing multi-tenancy technology. As a result, the number of VNF instances is reduced, so are the BRCs, number of activated MDCs and the total cost.

Algorithm 2 shows the framework of the VNFr merging algorithm. To describe the process more clearly and briefly, the VNFrs demanding for the same type VNF in one MDC are taken as a whole, and symbolized as ψ.

Algorithm 2 VNFr Merging Algorithm.

Input: Set of used MDCs: R_1, Status of the network: Ω_1;
Output: Set of used MDCs: R_2, Status of the network: Ω_2;

1: Figure out the ψ that can be moved in each used MDCs, and put the MDCs that have movable ψ in a new set, indicated as R_{temp}.
2: Calculate the resource utilization of each used MDC, and pick out the MDC ϖ that has the minimal resource utilization ω.
3: **if** $\omega < \sigma$ **then**
4:   **if** MDC ϖ has no P-SFCrs **then**
5:     Move all R-SFCrs to other used MDCs.
6:     Migrate ψes among the used MDCs. $\leftarrow$ **Procedure 2**.
7:     Record results. $\leftarrow$ **Procedure 3**.
8:   **else**
9:     Migrate ψes among the used MDCs. $\leftarrow$ **Procedure 2**.
10:     Record results. $\leftarrow$ **Procedure 3**.
11:   **end if**
12: **else**
13:   Migrate ψes among the used MDCs. $\leftarrow$ **Procedure 2**.
14:   Record results. $\leftarrow$ **Procedure 3**.
15: **end if**

Firstly, we need to figure out all the ψes that can be moved in each used MDC, and pick out the MDCs that have movable ψ (line 1 in Algorithm 2). The ψ is movable if it satisfies the following conditions:

1. None of the VNFrs in ψ is P-VNFr.
2. None of the R-SFCrs in the original MDC corresponding to ψ violates the latency constraints if VNFrs in ψ are migrated to another MDC.

To reduce the total VNF placement cost, MDCs should be activated as less as possible, because the corresponding cost of activating an MDC is far higher than other costs. So the MDC that has the minimal resource utilization is found out (ϖ) (line 2 in Algorithm 2). If the resource utilization is below a threshold (σ) and the MDC has no P-SFCrs, the corresponding MDC is emptied out by moving all R-SFCrs in it to other used MDCs (lines 3–5 in Algorithm 2). In this way, the MDC with low resource utilization can be turned off rapidly. Then the procedure of migrating ψes among the used MDCs continues (lines 6–15 in Algorithm 2).

Procedure 2 gives the pseudo codes of migrating ψes among used MDCs.

Firstly the ψ with minimal resource demand needs to be figured out in each used MDC (line 3 in Procedure 2). And then these ψes are sorted based on their resource consumptions in ascending order (line 4 in Procedure 2). Nextly, a target MDC that can be hosted on for each VNFr in each ψ should be found (lines 5–7 in Procedure 2). An MDC can be chosen as the destination for the migrated VNFr if the MDC has enough resource left and has been placed the corresponding VNF. If the latency constraints of the R-SFCr that contains the VNFr is not violated, then the VNFr can be moved to the target MDC (lines 8–9 in Procedure 2). Otherwise, the network status should rollback to the start of the migration process and the migration loop of current ψ should be break (lines 11–12 in Procedure 2). After finishing the migration loop of one ψ (lines 6–14 in Procedure 2), it needs to check if the ψ is migrated successfully (lines 15–24 in Procedure 2). If yes, the ψ is marked as processed and the related VNF instance is removed from the corresponding MDC. If not, the ψ is only marked as processed to avoid the repeated processing. When the ψ is migrated successfully, whether T_c is smaller than its previous value needs to be checked (lines 17–18 in Procedure 2). If yes, the previous placement results are replaced with current ones. The *while* loop (lines

Procedure 2 Migrate ψes among used MDCs.

1: Record the current results as the placement results.
2: **while** 1 **do**
3: Find the ψ that demands the minimal node resource in each MDC in R_{temp}, and store them in a list L_ψ.
4: Sort the ψes in L_ψ based on their resource consumptions in ascending order.
5: **for** each ψ in L_ψ **do**
6: **for** each VNFr in ψ **do**
7: Find the target MDC the VNFr can be migrated to.
8: **if** latency constraints of the corresponding R-SFCr is not violated **then**
9: Move the VNFr to the target MDC.
10: **else**
11: Restore the network status to the start of the migration of ψ.
12: Break.
13: **end if**
14: **end for**
15: **if** ψ is migrated successfully **then**
16: Mark the ψ as processed and remove the related VNF from the corresponding MDC.
17: **if** T_c is smaller **then**
18: Replace the results with current ones.
19: **else**
20: Continue.
21: **end if**
22: **else**
23: Mark the ψ in the MDC as processed.
24: **end if**
25: **end for**
26: **if** there is no unprocessed ψ in all used MDCs **then**
27: Break.
28: **else**
29: Continue.
30: **end if**
31: **end while**

2–31 in Procedure 2) will not break until all ψes in all used MDCs are marked as processed (lines 26–30 in Procedure 2).

In the process of Procedure 2, the resource violations are not considered. However, the results derived by Procedure 2 may contain occasions of resource violation, which need to be eliminated. If the resource violations cannot be eliminated, the results derived by Algorithm 2 should be abandoned and the results by Algorithm 1 are taken as the final placement decisions (lines 2–5 in Procedure 3).

Algorithm 3 Record results.

1: Check the resource violations of the results get by the previous processes.
2: **if** there are resource violations **then**
3: Eliminate the resource violations.
4: **if** Elimination process is not successful **then**
5: Restore results derived by Algorithm~1.
6: **else**
7: Record current results.
8: **end if**
9: **else**
10: Record current results.
11: **end if**

When eliminating the resource violations, the resource violations in MDCs are handled with in the first place. The idea is to remove the VNFr one by one from the MDC that has resource violation to the MDCs that have enough resources until the resource violation of the corresponding MDC is eliminated. During the process, the empty MDCs should not be the target MDC to receive the migrated VNFrs. If there is one MDC whose resource violation cannot be eliminated, the elimination process will not be successful.

After the elimination of node resource violations, the link resource violations follow. For each link that has resource violation, the violation volume of bandwidth resource is calculated. Then a path is derived by SPA from the ingress to the egress of link, and the minimal available bandwidth of the links in the path must be greater than the volume of bandwidth violation. If the path exists, the corresponding traffic is re-directed from the link that has resource violation to the path. Otherwise, the process of elimination is failed.

5.3. Complexity analysis

In this part, the time complexity of PG is analyzed. To make the descriptions more concisely, we would like to introduce the following two indicators that will be used frequently:

- $\mathbb{M} = |P| + |R| + |G|$, which is the number of all substrate nodes in the network.
- $\mathbb{N} = \sum_{\gamma=0}^{|\Gamma|-1}(|\Phi_\gamma| + |\Psi_\gamma| + 1)$, which is the number of all VNFrs.

Firstly, for Algorithm 1, the time complexity of Procedure 1 is $|\Gamma| \cdot |R| \cdot \mathbb{M} \cdot \log\mathbb{M}$, in which $\mathbb{M} \cdot \log\mathbb{M}$ is the time complexity of SPA. Then the time complexity of sorting process is $|R| \cdot \log|R|$. Nextly, for lines 3–27 in Algorithm 1, all the SFCrs are traversed, and for each SFCr, 4 paths are computed using SPA, which are SAR to MDC, MDC to CDC, MDC to SAR and CDC to SAR. So the time complexity of the process (lines 3–27 in Algorithm 1) is $|\Gamma| \cdot 4 \cdot \mathbb{M} \cdot \log\mathbb{M}$. At last, the total time complexity of Algorithm 1 is $|\Gamma| \cdot (|R| + 4) \cdot \mathbb{M} \cdot \log\mathbb{M} + |R| \cdot \log|R|$, which is at the level of $\boldsymbol{O}(|\Gamma| \cdot |R| \cdot \mathbb{M} \cdot \log\mathbb{M})$.

As for Algorithm 2, when figuring out the movable ψ (line 1 in Algorithm 2), we have to check every VNFr and calculate the possible paths to the candidate MDCs that it may be migrated to. So the time complexity of this process is $\mathbb{N} \cdot |R| \cdot \mathbb{M} \cdot \log\mathbb{M}$ at the worst case. The time complexity of line 2 in Algorithm 2 is $|R|$ at most.

Another time consuming part is Procedure 2, all the VNFrs have to be traversed and the SPA runs $|R|$ times in each loop at the worst case, then the time complexity of this part is $\mathbb{N} \cdot |R| \cdot \mathbb{M} \cdot \log\mathbb{M}$. As for the process of eliminating resource violations, the least time complexity is 0, and the general situation is $\frac{1}{\tau} \cdot \mathbb{N} \cdot |R| \cdot \mathbb{M} \cdot \log\mathbb{M}$, $\tau > 1$.

So the total time complexity of Algorithm 2 is $(2 + \frac{1}{\tau}) \cdot \mathbb{N} \cdot |R| \cdot \mathbb{M} \cdot \log\mathbb{M} + |R|$, $\tau > 1$, which is at the level of $\boldsymbol{O}(\mathbb{N} \cdot |R| \cdot \mathbb{M} \cdot \log\mathbb{M})$.

Generally speaking, $\mathbb{N} \gg |\Gamma|$, so the total time complexity of PG is $\boldsymbol{O}(\mathbb{N} \cdot |R| \cdot \mathbb{M} \cdot \log\mathbb{M})$.

6. Performance evaluation

In this section, we evaluate the performance of PG and make a thorough analysis to the VNF placement problem in NFV and EC enabled networks. The ILP model formalized in previous section is implemented and run in Gurobi Optimizer version 7.0 [48]. The heuristic and the contrastive schemes, in turn, are implemented in Python. All experiments are performed on a computer with one Intel(R) Core(TM) i3-3220 CPU @ 3.30 GHz and 6 GB of RAM.

6.1. Settings

Our simulation parameters refer to the existing works [13,49,50]. The substrate network topology is generated by BRITE [51] based on the Waxman model [52], and there are 100 SARs, 50 MDCs and 500 links in it. Both the CPU and memory capacity of each MDC are set to be 4000, which are also the maximum available resources of CPU and memory on each MDC. The propagation delay on each link obeys the uniform distribution of (0,2).

In the same experiment, the length of each SFCr is the same. For a single SFCr, it has 1 SAR, 4 VNFrs that should be hosted in MDCs, and 1 VNFr that should be in CDC.

The D_γ^M of each SFCr obeys the uniform distribution of (1,2), and D_γ^C obeys the uniform distribution of (5,10). σ in Algorithm 2 is set to be 0.2. Without losing generality, node resource and bandwidth resource are treated as the same importance in our simulations, so the weighted factors α, β, ρ and ϱ in optimization target T_c are all set to be 1.

6.2. Workloads

In the simulation, both the value of CPU BRCs and memory BRCs are set to be 20, and the bandwidth consumption of each SFCr obeys the uniform distribution of (10,50). The relationship between the size of BRCs when instantiating an VNF and the resource demand by each VNFr may have influences on the performance of different solutions. To reveal the above influences, two kinds of VNFr workloads are evaluated in the simulations:

- SFCr A: The CPU and memory consumption of each VNFr in the SFCr obey the uniform distribution of (40,80).
- SFCr B: The CPU and memory consumption of each VNFr in the SFCr obey the uniform distribution of (4,8).

6.3. Benchmarks

We have two benchmarks:

- RG: RG, which is the abbreviation for Random Greedy, is another heuristic we propose. Firstly, we classify the SFCrs into $|R|$ clusters the same as that in Algorithm 1. Then clusters that should be processed are picked randomly, and SFCrs are mapped to the MDCs based the latency constraints of them. The SFCrs whose latency constraints (D_γ^M) are stricter have higher mapping priority. Nextly, all the SFCrs are traversed and mapped to the MDCs one by one. In this way, all the SFCrs are mapped to the MDCs.
- BSVR: It is the heuristic solution proposed by [50]. BSVR considers VNF reusing and latency minimization. Firstly, BSVR derives several candidate paths for an SFCr to be deployed. Then for each path, it counts the overlap number (ϑ) between VNFs in the path and VNFs the SFCr demands, and calculates the end to end delay of the path (D_p). At last, BSVR picks the path that has the maximum value of ϑ/D_p as the one to host the SFCr.

6.4. Results

For each group of results, 10 times of experiments are conducted to reduce the accidental errors, and error bars represent the 95% confidence intervals in corresponding figures. In each group of results, the final BRCs, bandwidth consumptions and number of activated MDCs are used to evaluate the performance of different solutions.

6.4.1. Performance comparisons versus different SFCr number

In this part, the performance of different solutions are evaluated via the varying number of SFCrs. Fig. 3 shows the results, and the workloads are the mixture of SFCr A and SFCr B with the ratio of 1:1. From the figure, it can be seen that PG performs the best from the aspects of BRCs and number of activated MDCs, which owes to that PG maps the SFCrs in a priority based sequence. If an SFCr that has looser latency constraint is mapped into one MDC firstly, then it may result in a consequence that the SFCr that has stricter latency constraint cannot be mapped into the same MDC, and a new MDC has to be activated nearby. So unnecessary MDCs may be activated if the mapping sequence is inefficient. Moreover, PG merges VNFrs of the same type together, and then utilizes multi-tenancy principle to implement the VNF instances. In this way, PG reduces the number of VNF instances, then less BRCs are incurred. BSVR considers the re-using of VNFs among the MDCs on the mapping path of the corresponding SFC, which is an optimization scheme based on local information. Meanwhile, PG merges the VNFrs of the same type globally. So the results of PG is superior to that of BSVR, in terms of BRCs and number of activated MDCs.

As for the bandwidth consumptions, PG consumes a little more bandwidth resource than the other contrastive schemes. It is because PG instantiates less VNF instances, so when chaining the VNFs, the flows have to go through more links to accomplish an SFC, and then the bandwidth consumptions are more. RG consumes the least bandwidth, it is because most parts of one SFCr are usually mapped on the same MDC, so the flows that inner the SFCrs are limited in the MDCs, leading to much bandwidth saving. BSVR tends to place the VNF instances used by one SFCr into different MDCs, so flows need to go through more links than that by RG, and bandwidth consumptions by BSVR are near to that by PG.

6.4.2. Performance comparisons versus different kinds of SFCr workloads

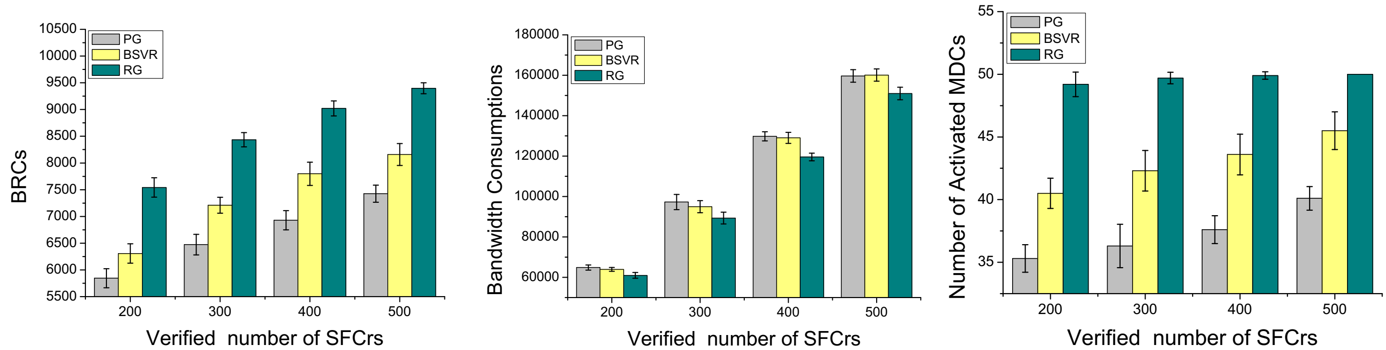
In this part, we try to reveal the influence of different kinds of workloads on the performance of different solutions. Fig. 4 shows the results, and the number of SFCrs is 400 in each group of experiment. In the experiment, we have 3 groups of SFCrs, which are pure SFCr A, mixture of SFCr A and SFCr B with ratio of 1:1 and pure SFCr B, respectively.

From the results, we can derive similar conclusions with Fig. 3. Moreover, we can see that different workloads have very slightly influences on the performance of the solutions. It owes to the distribution characteristic of the EC enabled network and strict latency constraints of users' requests, which restrict that SFCrs are usually mapped into the MDCs nearby. So the VNFs are usually placed on the MDCs nearby, the optimization space is limited, and then the influences of different workloads on the solutions are diminished.

6.4.3. PG Versus optimal results

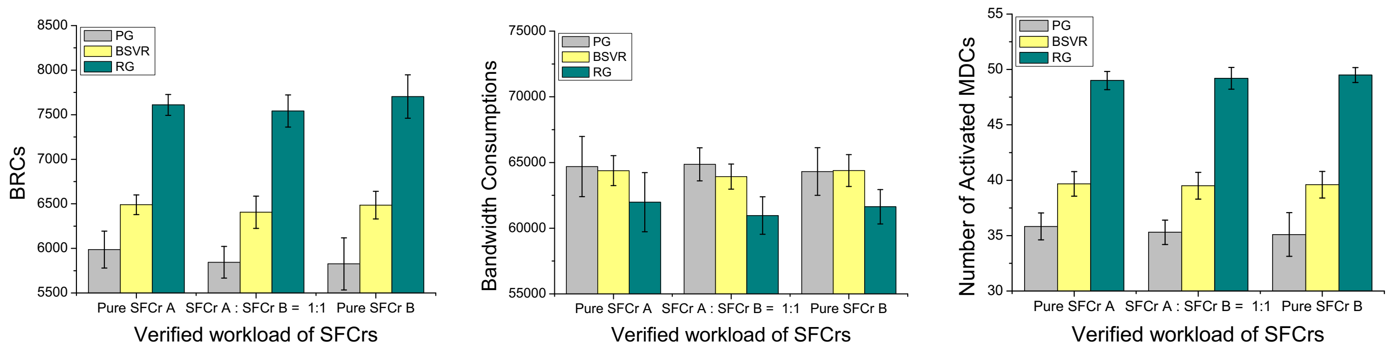
In this part, the performance of PG is compared with the optimal results derived by Gurobi [48] with a substrate network of 30 SARs, 15 MDCs and 150 links, and the number of SFCrs is 30. From Fig. 5, on one hand, it can be seen that BRCs by PG are about 1.6 times of the optimal results, number of activated MDCs is about 1.5 times, and bandwidth consumption is about 1.2 times, which indicate that the performance of PG is near to the optimal solution when the problem is in small scale. On the other hand, it is also superior to the performance of BSVR and RG.

Besides the above comparisons, we also measure the runtime of each solution deriving the results of one placement process. As is shown in Table. 2, the time consumed by Gurobi based solution is far longer than that by other three heuristics. It is because the solution space is exponential, and time complexity of finding the optimal solution is also in exponential level. Moreover, the solution



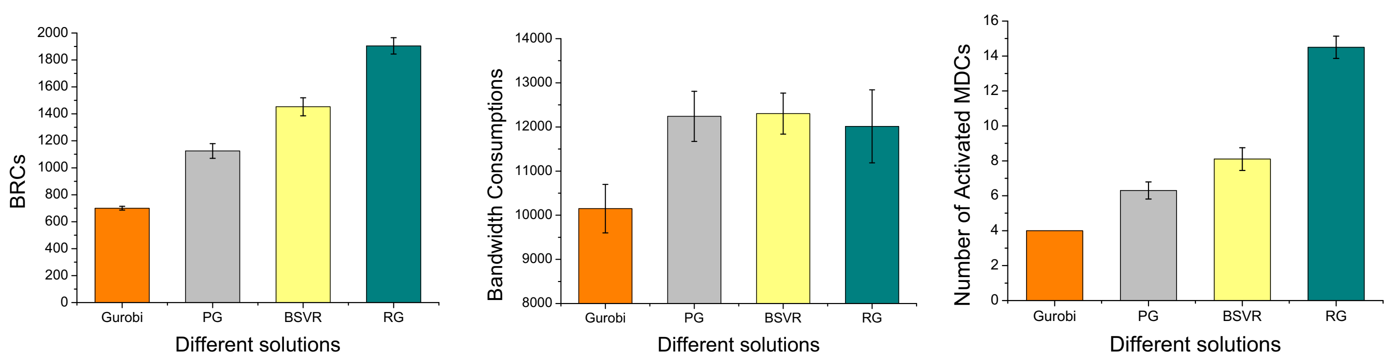
(a) BRCs (b) Bandwidth cost (c) Number of activated MDCs

Fig. 3. Performance comparisons vs different number of SFCrs.



(a) BRCs (b) Bandwidth cost (c) Number of activated MDCs

Fig. 4. Performance comparisons vs verified workloads of SFCrs.



(a) BRCs (b) Bandwidth cost (c) Number of activated MDCs

Fig. 5. Performance comparisons vs Different solutions.

Table 2
Runtime of each solution deriving the results of one placement process (unit: s).

Gurobi	PG	RG	BSVR
$10^2 - 10^5$	18–21	10–13	13–20

space can be varied according to different scenarios, so the convergence time of Gurobi varies a long range. PG, BSVR and RG all have polynomial time complexity, so the consumed time by them is at the same level.

6.4.4. The utilization of network resources

Fig. 6(a)–(c) show the cumulative distribution function (CDF) curves about CPU utilization, memory utilization on MDCs and bandwidth utilization on links, respectively. The number of SFCrs is 400 and the workloads are the mixture of SFCr A and SFCr B with ratio of 1:1.

From the figures, we can see that node resource (CPU and memory) utilization derived by PG is highest. The higher the node resource utilization is, the less the activated MDCs will be. The results about node resource utilization are consistent with that about

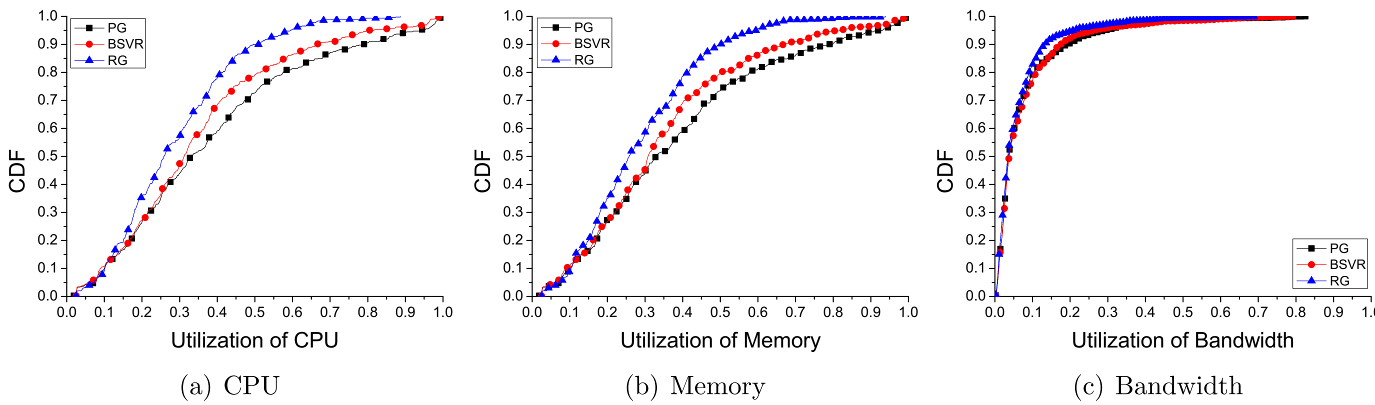


Fig. 6. CDF of different resource utilization.

activated number of MDCs. As for bandwidth utilization, we can see that RG gets the lowest utilization, which also indicates that it consumes the least bandwidth resource.

7. Conclusion

In this paper, we study the VNF placement and resource optimization problem in NFV and EC enabled networks with a set of known SFCrs. The SFCrs are hierarchical and geo-distributed owing to the characteristics of EC. Besides, the latency constraints are heterogeneous and more stricter. Moreover, the instantiation method of VNFs is considered, which is that the VNFs are instantiated utilizing the multi-tenancy principle in this paper. And then the confliction between the node resource consumption and bandwidth consumption is clarified. Mathematically, the VNF placement problem is formulated as an ILP model with the aim of minimizing the total resource consumption in substrate network. The ILP is NP-hard, apart from calculating the optimal results by Gurobi when the problem scale is small, a heuristic solution PG is proposed to solve the problem in polynomial time. In the last of the paper, an evaluation to the heuristic is made in detail, and the simulation results show that the resource cost derived by PG is near to the optimal results when the problem scale is small and it outperforms the contrastive schemes very much when the problem scale is large.

Acknowledgment

The work was supported by National Natural Science Foundation of China (NSFC) under Grant No. 61671420, 61672484.

References

[1] M. Chiosi, D. Clarke, P. Willis, A. Reid, J. Feger, M. Bugenhagen, W. Khan, M. Fargano, C. Cui, H. Deng, et al., Network functions virtualisation: an introduction, benefits, enablers, challenges and call for action, in: SDN and OpenFlow World Congress, 2012, pp. 22–24.
[2] ETSI White Paper No. 23, Cloud RAN and MEC: A Perfect Pairing, European Telecommunications Standards Institute, 2018, [Online]. Available: http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp23_MEC_and_CRAN_ed1_FINAL.pdf.
[3] G. Faraci, G. Schembra, An analytical model to design and manage a green sdn/nfv cpe node, IEEE Trans. Netw. Serv. Manage. 12 (3) (2015) 435–450, doi:10.1109/TNSM.2015.2454293.
[4] J. Halpern, C. Pignataro, Service Function Chaining (SFC) architecture, Technical Report, 2015, doi:10.17487/RFC7665.
[5] M. Satyanarayanan, The emergence of edge computing, Computer 50 (1) (2017) 30–39, doi:10.1109/MC.2017.9.
[6] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: vision and challenges, IEEE Internet Thing. J. 3 (5) (2016) 637–646, doi:10.1109/JIOT.2016.2579198.
[7] P. Mach, Z. Becvar, Mobile edge computing: a survey on architecture and computation offloading, IEEE Commun. Surv. Tutor. 19 (3) (2017) 1628–1656, doi:10.1109/COMST.2017.2682318.
[8] F. Bonomi, R. Milito, P. Natarajan, J. Zhu, Fog computing: a platform for internet of things and analytics, in: Big Data and Internet of Things: A Roadmap for Smart Environments, Springer, 2014, pp. 169–186, doi:10.1007/978-3-319-05029-4_7.
[9] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, ACM, 2012, pp. 13–16, doi:10.1145/2342509.2342513.
[10] M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies, The case for vm-based cloudlets in mobile computing, IEEE Pervas. Comput. 8 (4) (2009), doi:10.1109/MPRV.2009.82.
[11] B. Blanco, J.O. Fajardo, I. Giannoulakis, E. Kafetzakis, S. Peng, J. Pérez-Romero, I. Trajkovska, P.S. Khodashenas, L. Goratti, M. Paolino, et al., Technology pillars in the architecture of future 5g mobile networks: nfv, mec and sdn, Comput. Standard. Interfaces 54 (2017) 216–228, doi:10.1016/j.csi.2016.12.007.
[12] S. Natarajan, R. Krishnan, A. Ghanwani, D. Krishnaswamy, P. Willis, A. Chaudhary, F. Huici, An analysis of lightweight virtualization technologies for NFV, [Online]. Available: https://tools.ietf.org/html/draft-natarajan-nfvrg-containers-for-nfv-03, 2017.
[13] D. Li, P. Hong, K. Xue, J. Pei, Virtual network function placement considering resource optimization and sfc requests in cloud datacenter, IEEE Trans. Parallel Distrib.Syst. (7) (2018) 1664–1677, doi:10.1109/TPDS.2018.2802518.
[14] C.-P. Bezemer, A. Zaidman, Multi-tenant saas applications: maintenance dream or nightmare? in: Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE), ACM, 2010, pp. 88–92, doi:10.1145/1862372.1862393.
[15] M.T. Hajiaghayi, M. Mahdian, V.S. Mirrokni, The facility location problem with general cost functions, Networks 42 (1) (2003) 42–47, doi:10.1002/net.10080.
[16] F. Bari, S.R. Chowdhury, R. Ahmed, R. Boutaba, O.C.M.B. Duarte, Orchestrating virtualized network functions, IEEE Trans. Netw. Serv. Manage. 13 (4) (2016) 725–739, doi:10.1109/TNSM.2016.2569020.
[17] B. Addis, D. Belabed, M. Bouet, S. Secci, Virtual network functions placement and routing optimization, in: Cloud Networking (CloudNet), 2015 IEEE 4th International Conference on, IEEE, 2015, pp. 171–177, doi:10.1109/CloudNet.2015.7335301.
[18] S. Mehraghdam, M. Keller, H. Karl, Specifying and placing chains of virtual network functions, in: Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on, IEEE, 2014, pp. 7–13, doi:10.1109/CloudNet.2014.6968961.
[19] W. Szeto, Y. Iraqi, R. Boutaba, A multi-commodity flow based approach to virtual network resource allocation, in: Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE, 6, IEEE, 2003, pp. 3004–3008, doi:10.1109/GLOCOM.2003.1258787.
[20] Network Functions Virtualization (NFV); Management and Orchestration, European Telecommunications Standards Institute, [Online]. Available: http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_nfv-man001v010101p.pdf.
[21] ETSI GR NFV 001 V1.2.1, Network Functions Virtualization (NFV); Use cases, European Telecommunications Standards Institute, 2017, [Online]. Available: https://www.etsi.org/deliver/etsi_gr/NFV/001_099/001/01.02.01_60/gr_nfv001v010201p.pdf.
[22] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, R. Boutaba, Network function virtualization: state-of-the-art and research challenges, IEEE Commun. Surv. Tutor. 18 (1) (2016) 236–262, doi:10.1109/COMST.2015.2477041.
[23] B. Han, V. Gopalakrishnan, L. Ji, S. Lee, Network function virtualization: challenges and opportunities for innovations, IEEE Commun. Mag. 53 (2) (2015) 90–97, doi:10.1109/MCOM.2015.7045396.
[24] M.C. Luizelli, L.R. Bays, L.S. Buriol, M.P. Barcellos, L.P. Gaspary, Piecing together the nfv provisioning puzzle: efficient placement and chaining of virtual network functions, in: Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on, IEEE, 2015, pp. 98–106, doi:10.1109/INM.2015.7140281.
[25] Z. Xu, W. Liang, A. Galis, Y. Ma, Q. Xia, W. Xu, Throughput optimization for admitting nfv-enabled requests in cloud networks, Comput. Netw. (2018), doi:10.1016/j.comnet.2018.06.015.
[26] L. Qu, C. Assi, K. Shaban, Delay-aware scheduling and resource optimization with network function virtualization, IEEE Trans. Commun. 64 (9) (2016) 3746–3758, doi:10.1109/TCOMM.2016.2580150.

[27] M. Xia, M. Shirazipour, Y. Zhang, H. Green, A. Takacs, Network function placement for nfv chaining in packet/optical datacenters, J. Lightwave Technol. 33 (8) (2015) 1565–1570, doi:10.1109/JLT.2015.2388585.

[28] R. Cohen, L. Lewin-Eytan, J.S. Naor, D. Raz, Near optimal placement of virtual network functions, in: Computer Communications (INFOCOM), 2015 IEEE Conference on, IEEE, 2015, pp. 1346–1354, doi:10.1109/INFOCOM.2015.7218511.

[29] C. Pham, N.H. Tran, S. Ren, W. Saad, C.S. Hong, Traffic-aware and energy-efficient vnf placement for service chaining: joint sampling and matching approach, IEEE Trans. Serv. Comput. (2017), doi:10.1109/TSC.2017.2671867.

[30] Z. Ye, X. Cao, J. Wang, H. Yu, C. Qiao, Joint topology design and mapping of service function chains for efficient, scalable, and reliable network functions virtualization, IEEE Netw 30 (3) (2016) 81–87, doi:10.1109/MNET.2016.7474348.

[31] T. Wen, H. Yu, G. Sun, L. Liu, Network function consolidation in service function chaining orchestration, in: Communications (ICC), 2016 IEEE International Conference on, IEEE, 2016, pp. 1–6, doi:10.1109/ICC.2016.7510679.

[32] M.S. de Brito, S. Hoque, T. Magedanz, R. Steinke, A. Willner, D. Nehls, O. Keils, F. Schreiner, A service orchestration architecture for fog-enabled infrastructures, in: Fog and Mobile Edge Computing (FMEC), 2017 Second International Conference on, IEEE, 2017, pp. 127–132, doi:10.1109/FMEC.2017.7946419.

[33] E.K. Markakis, K. Karras, N. Zotos, A. Sideris, T. Moysiadis, A. Corsaro, G. Alexiou, C. Skianis, G. Mastorakis, C.X. Mavromoustakis, et al., Exegesis: extreme edge resource harvesting for a virtualized fog environment, IEEE Commun. Mag. 55 (7) (2017) 173–179, doi:10.1109/MCOM.2017.1600730.

[34] X. Sun, N. Ansari, Latency aware workload offloading in the cloudlet network, IEEE Commun. Lett. 21 (7) (2017) 1481–1484, doi:10.1109/LCOMM.2017.2690678.

[35] M. Ali, N. Riaz, M.I. Ashraf, S. Qaisar, M. Naeem, Joint cloudlet selection and latency minimization in fog networks, IEEE Trans. Ind. Inf. (2018), doi:10.1109/TII.2018.2829751.

[36] ETSI GS MEC-IEG 004 V1.1.1, Mobile-Edge Computing (MEC); Service Scenarios, European Telecommunications Standards Institute, 2015, [Online]. Available: http://www.etsi.org/deliver/etsi_gs/MEC-IEG/001_099/004/01.01.01_60/gs_MEC-IEG004v010101p.pdf.

[37] ETSI White Paper No. 24, MEC Deployments in 4G and Evolution Towards 5G, European Telecommunications Standards Institute, 2018, [Online]. Available: http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp24_MEC_deployment_in_4G_5G_FINAL.pdf.

[38] F. van Lingen, M. Yannuzzi, A. Jain, R. Irons-Mclean, O. Lluch, D. Carrera, J.L. Perez, A. Gutierrez, D. Montero, J. Marti, et al., The unavoidable convergence of nfv, 5g, and fog: a model-driven approach to bridge cloud and edge, IEEE Commun. Mag. 55 (8) (2017) 28–35, doi:10.1109/MCOM.2017.1600907.

[39] A. Manzalini, N. Crespi, An edge operating system enabling anything-as-a-service, IEEE Commun. Mag. 54 (3) (2016) 62–67, doi:10.1109/MCOM.2016.7432173.

[40] Y. Nam, S. Song, J.-M. Chung, Clustered NFV service chaining optimization in mobile edge clouds, IEEE Commun. Lett. 21 (2) (2017) 350–353, doi:10.1109/LCOMM.2016.2618788.

[41] F. Callegati, W. Cerroni, C. Contoli, G. Santandrea, Dynamic chaining of virtual network functions in cloud-based edge networks, in: Network Softwarization (NetSoft), IEEE, 2015, pp. 1–5, doi:10.1109/NETSOFT.2015.7116127.

[42] F.B. Jemaa, G. Pujolle, M. Pariente, Qos-aware VNF placement optimization in edge-central carrier cloud architecture, in: Global Communications Conference (GLOBECOM), IEEE, 2016, pp. 1–7, doi:10.1109/GLOCOM.2016.7842188.

[43] Network Functions Virtualisation-White Paper on NFV priorities for 5G, European Telecommunications Standards Institute, 2017, [Online]. Available: http://portal.etsi.org/NFV/NFV_White_Paper_5G.pdf.

[44] ETSI GR MEC 017 V1.1.1, Mobile Edge Computing (MEC); Deployment of Mobile Edge Computing in an NFV environment, European Telecommunications Standards Institute, 2018, [Online]. Available: http://www.etsi.org/deliver/etsi_gr/MEC/001_099/017/01.01.01_60/gr_MEC017v010101p.pdf.

[45] S. Pal, A.K. Mandal, A. Sarkar, Application multi-tenancy for software as a service, ACM SIGSOFT Softw. Eng. Notes 40 (2) (2015) 1–8, doi:10.1145/2735399.2735412.

[46] A.M. Medhat, G. Carella, J. Mwangama, N. Ventura, Multi-tenancy for virtualized network functions, in: Network Softwarization (NetSoft), 2015 1st IEEE Conference on, IEEE, 2015, pp. 1–6, doi:10.1109/NETSOFT.2015.7116177.

[47] MATH3902 Operations Research II: Integer Programming Topics, The University of Hong Kong, 2004, [Online]. Available: http://web.hku.hk/~schu/IntegerProgramming.pdf.

[48] Gurobi optimizer reference manual, version 7.0, Gurobi, [Online]. Available: http://www.gurobi.com/documentation/7.0/refman.pdf.

[49] L.F. Bittencourt, J. Diaz-Montes, R. Buyya, O.F. Rana, M. Parashar, Mobility-aware application scheduling in fog computing, IEEE Cloud Comput. 4 (2) (2017) 26–35, doi:10.1109/MCC.2017.27.

[50] D. Li, J. Lan, P. Wang, Joint service function chain deploying and path selection for bandwidth saving and vnf reuse, Int. J. Commun. Syst. (2018), doi:10.1002/dac.3523.

[51] A. Medina, A. Lakhina, I. Matta, J. Byers, Brite: An approach to universal topology generation, in: Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2001. Proceedings. Ninth International Symposium on, IEEE, 2001, pp. 346–353, doi:10.1109/MASCOT.2001.948886.

[52] B.M. Waxman, Routing of multipoint connections, IEEE J. Sel. Areas Commun. 6 (9) (1988) 1617–1622, doi:10.1109/49.12889.

Defang Li was born in 1991. He received his B.S. degree from the Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC), in 2014 and he is pursuing for his Ph.D in USTC with his advisor Peilin Hong now. His research interests include SDN, NFV, and the network resource orchestration and management.

Peilin Hong was born in 1961. She received her B.S. and M.S. degrees from the Department of Electronic Engineering and Information Science (EEIS), University of Science and Technology of China (USTC), in 1983 and 1986. Currently, she is a Professor and Advisor for Ph.D. candidates in the Department of EEIS, USTC. Her research interests include next-generation Internet, policy control, IP QoS, and information security. She has published 2 books and over 100 academic papers in several journals and conference proceedings.

Kaiping Xue received his B.S. degree from the Department of Information Security, University of Science and Technology of China (USTC), in 2003 and received his Ph.D. degree from the Department of Electronic Engineering and Information Science (EEIS), USTC, in 2007. From May 2012 to May 2013, he was a postdoctoral researcher with Department of Electrical and Computer Engineering, University of Florida. Currently, he is an Associate Professor in the Department of Information Security and Department of EEIS, USTC. He currently serves on the editorial board of Ad Hoc Networks. His research interests include next-generation Internet, distributed networks and network security.

Jianing Pei was born in 1992. He received his B.S. degree from the Department of Information and Electrical Engineering (IEE), China University of Mining and Technology (CUMT), in 2015 and he is pursuing for his M.S. degree in USTC with his advisor Peilin Hong now. His research interests include SDN, NFV, and the network resource orchestration and management.