Computer Communications 151 (2020) 449–462

Contents lists available at ScienceDirect

Computer Communications

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

Energy efficient network service deployment across multiple SDN domains

Chuangchuang Zhang [b], Xingwei Wang [a,b,*], Anwei Dong [b], Yong Zhao [b], Qiang He [b], Min Huang [c,a]

[a] State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, 110819, China
[b] College of Computer Science and Engineering, Northeastern University, Shenyang, 110819, China
[c] College of Information Science and Engineering, Northeastern University, Shenyang, 110819, China

ARTICLE INFO

Keywords:
Network service deployment
Multiple SDN domains
Energy saving
Load balancing

ABSTRACT

The emergence of Software Defined Networking (SDN) and Network Function Virtualization (NFV) enables flexible service provisioning and deployment. However, with the continuous expansion of network scale and sharp increasing of end users, how to flexibly provide network services across multiple SDN domains for users is becoming a critical issue. A major challenge in the multi-domain network service provisioning is the network service deployment method taking into account energy efficiency. In this paper, we study the problem of how to optimally deploy network services across multiple SDN domains with the target of saving energy while achieving the load balancing of multi-domain networks. Specifically, firstly, we propose a novel multi-domain network service deployment framework by integrating SDN architecture and NFV technology, which can intelligently deploy virtual network functions (VNFs) into multi-domain networks. Secondly, we formulate this problem as a multi-objective optimization model to achieve the minimization of energy consumption and load balancing of multi-domain networks. Furthermore, we present a heuristic network service deployment algorithm to solve it. Finally, simulation results demonstrate that the proposed heuristic service deployment algorithm is efficient and outperforms comparison algorithms in terms of energy consumption and load balancing degree.

1. Introduction

In recent years, a large number of middleboxes (e.g., Network Address Translator (NAT), Firewall (FW) and Intrusion Detection System (IDS)) have been developed and deployed in current networks to provide various network services. However, it is inefficient for service provisioning due to the ossified architecture of traditional network and middleboxes's heavy dependence on specific hardware equipment [1]. Fortunately, the emergence of Network Function Virtualization (NFV) brings new opportunities to address the challenge [2,3]. It implements network functions in software rather than hardware devices, running on virtual machines hosting on commodity servers [4]. One network service in NFV can be represented as one Service Function Chain (SFC), which is composed of a series of Virtual Network Functions (VNFs) in a given order. To provide efficient service, the SFC is deployed in several different servers and traffic is steered to pass through a set of VNFs in a specific sequence. NFV supports flexible service provisioning, significantly reducing Capital Expenditures (CAPEX) and Operational Expenditures (OPEX). On the other hand, Software Defined Networking (SDN), as an emerging networking paradigm, achieves the separation of control plane and data plane, and provides logically centralized control capacity as well as powerful programmable capability [5–10]. By integrating SDN with NFV, flexible network service deployment can be realized [11]. Specifically, when we deploy network services, the SDN controller can provide elastic and efficient resource management for VNFs to optimize network performance by leveraging the centralized control capacity.

One of the key issues in NFV is the optimal deployment of SFC taking into account energy efficiency [12]. Recently, energy efficiency has currently become a critical issue with the explosive growth of network traffic and electricity consumption. In such environment where SFCs are created and VNFs are deployed on virtual machines that utilize the resources provided by the physical servers, the decision on deployment location of SFC has a significant impact on efficient use of resource and energy consumption. High power consumption results in energy waste and excessive carbon emissions, and exacerbates greenhouse effect. The Internet devices and network infrastructures need to be significantly more energy efficient and nowadays network operators and service providers are required to pay more attention to energy consumption [13]. Energy efficiency has been considered when dealing with the SFC deployment in current studies. Despite these efforts [14–19], most existing works focus on single domain networks rather than multi-domain networks. However, the rapidly growing network traffic and constant enlargement of network scales pose serious challenges to single domain networks in terms of energy saving and load balancing.

* Corresponding author.
E-mail address: wangxw@mail.neu.edu.cn (X. Wang).

https://doi.org/10.1016/j.comcom.2020.01.019
Received 12 February 2019; Received in revised form 16 December 2019; Accepted 8 January 2020
Available online 15 January 2020

Internet has evolved from single-domain networks to multi-domain networks in recent years. With the sharp increasing of network users and continuous expansion of network scale, the Internet is composed of several different domains which are managed by different operators and service providers. In such networks, each domain presents different internal policies, such as resource usage and energy consumption. As a result, the complexity of the SFC deployment problem in multi-domain networks has increased manifold. The main challenge of provisioning a SFC request in multi-domain networks is the lack of global information for all the domains. Moreover, unlike SFC deployment in single domain networks, the VNFs in multi-domain networks are deployed into several different domains due to limited resource capacity of each domain, and virtual links between VNF nodes are mapped into inter-domain physical links if source and destination are in two different domains. The servers in different domains consume different resources to run the VNFs for service provisioning. During service provisioning, servers consume enormous amounts of powers and generate lots of energy consumption. This imposes serious challenges to service providers and end users. First, the growth of network devices' power significantly increases service provider's OPEX. Second, excessive energy consumption affects network performance and degrades user's QoE. Hence, it is necessary to design an energy efficient multi-domain service deployment scheme.

To efficiently save energy, the traditional SFC deployment methods [20–24] try to share physical server as much as possible for activating fewer servers. Different VNFs may be consolidated on the same physical servers in a same domain, and all the VNFs may not be uniformly deployed in multiple domains. It results in load imbalance of the multi-domain network system, thereby affecting the overall QoE of end users. Thus, it is desirable to achieve load balancing when we determine an optimal multi-domain SFC deployment scheme to maximize energy efficiency.

Recently, several proposals have been conducted to address the multi-domain SFC deployment issues [25–30]. However, there are few efforts to improve the energy efficiency of the SFC placement in multi-domain networks. Although several attempts in [26–28] have been made, they ignore the load balancing of multi-domain networks. Furthermore, the multi-domain SFC deployment requires an efficient allocation of resources for different service requests at a reduced cost. Thus, it is expected that an energy efficient solution can dynamically adapt to the usage of resources according to service demands.

To this end, in this paper, we study the multi-domain network service deployment problem by jointly taking energy consumption and load balancing into account. An intelligent multi-domain network service deployment framework is proposed to jointly optimize energy consumption of the servers and load balancing of the multi-domain network system. Our main contributions are as follows.

- By integrating SDN architecture and NFV technology, we present a novel energy efficient network service deployment framework to intelligently deploy the SFC into multi-domain networks, wherein the global network knowledge and centralized decision making mechanism of SDN make it efficiently realize energy optimization and load balancing.
- We formulate the problem of multi-domain network service deployment as a multi-objective optimization model with the target of minimizing energy consumption and load balancing degree by considering resource allocation as the constraint.
- To solve the multi-objective optimization model, we propose a heuristic service deployment algorithm.
- We conduct extensive simulations for performance evaluation. And simulation results demonstrate that the proposed heuristic algorithm can efficiently reduce energy consumption and achieve the load balancing of the multi-domain network system.

The remainder of this paper is organized as follows. Section 2 reviews the related work. The system framework is proposed in Section 3. Section 4 introduces the problem formulation, and the heuristic deployment algorithm is presented in Section 5. Section 6 shows the performance evaluation. Finally, we conclude this paper in Section 7.

2. Related work

The energy efficient SFC deployment problem has been studied [14–24] over the past decades. It is generally formulated as an optimization model by considering different constraints in most existing work, and different optimization approaches have been proposed to save energy.

In [14], Tajiki et al. formulated the resource allocation problem of SFC deployment as an Integer Linear Programming (ILP) model by jointly considering energy consumption and SFC parameters. In [15], Kar et al. presented an energy saving model for dynamic VNF placement and formulated it as an optimization model by considering server capacity and delay. Jang et al. [16] established a multi-objective optimization model with the aim of acceptable flow rate maximization and energy cost minimization, and transformed it into a single objective optimization model. Huin et al. [17] studied the problem of energy consumption minimization while satisfying link and node capacity constraints as well as SFC constraints. Yang et al. [18] addressed the energy aware SFC placement problem in data centers by jointly considering the energy consumed by servers and switches. Tajiki et al. [19] proposed a resource allocation architecture for VNFs by jointly considering VNF placement and QoS routing.

Similarly, Eramo et al. [20] proposed a VNF migration policy to minimize consolidation and migration energy cost, and formulated it as an ILP model. Pham et al. [21] studied the traffic aware and energy efficient SFC placement problem and formulated it as a combinatorial NP-hard model. The authors in [22] presented a robust VNF placement and routing optimization model to minimize the energy consumption caused by computing and network infrastructure. Sun et al. [23] designed a first-fit and greedy algorithm based SFC deployment method to reduce energy consumption. In [24], the VNF placement and chaining problem was formulated as a decision tree model with aim at saving energy. Meanwhile, an extension of the Monte Carlo tree search method was employed to achieve resource consolidation and VNF sharing between multiple tenants.

Although many significant efforts have been input to save energy, they only focus on SFC deployment in single domain networks rather than in multi-domain networks. Most works try to save energy by minimizing the total number of active servers or simply closing idle network devices. For example, the authors in [18] consolidated SFCs in the minimum set of servers, and used as few switches and links as possible to save energy. Similarly, Kim et al. [31] dynamically consolidated the VNFs with low service traffic into other servers to reduce the number of idle servers, thereby minimizing energy consumption. In [32], by leveraging the adaptive rate strategy, servers dynamically adjusted the processing capacity based on the coming workload to reduce energy consumption. However, different from single domain networks, the VNFs are deployed in different SDN domains in multi-domain network. Although, SFCs are consolidated in the minimum set of servers, and as few switches and links as possible are used, it easily results in load imbalance, thereby affecting network performance and user's QoE. Most existing approaches are not unfeasible for our problem since energy efficiency and load balancing of multi-domain networks are not together guaranteed.

To cope with load balancing, several researches have conducted recently. Wang et al. [33] explored the problem of multi-resource load balancing for VNFs, and formulated it as an optimization problem. Specifically, the dominant load that is defined as the maximum load for all resource types was proposed as the load balancing metric. Thai et al. [34] proposed a load balancing system for VNF chaining, to diminish control and data plane overheads in existing service chaining solutions by steering network traffic to different VNFs via multiple paths. Fei et al. [35] proposed a novel framework for VNF assignment in geo-distributed NFV infrastructure to achieve cost efficiency and balancing of both computing and bandwidth resource. Carpio et al. [36] tried to achieve load balancing by optimizing VNF placement with replication.

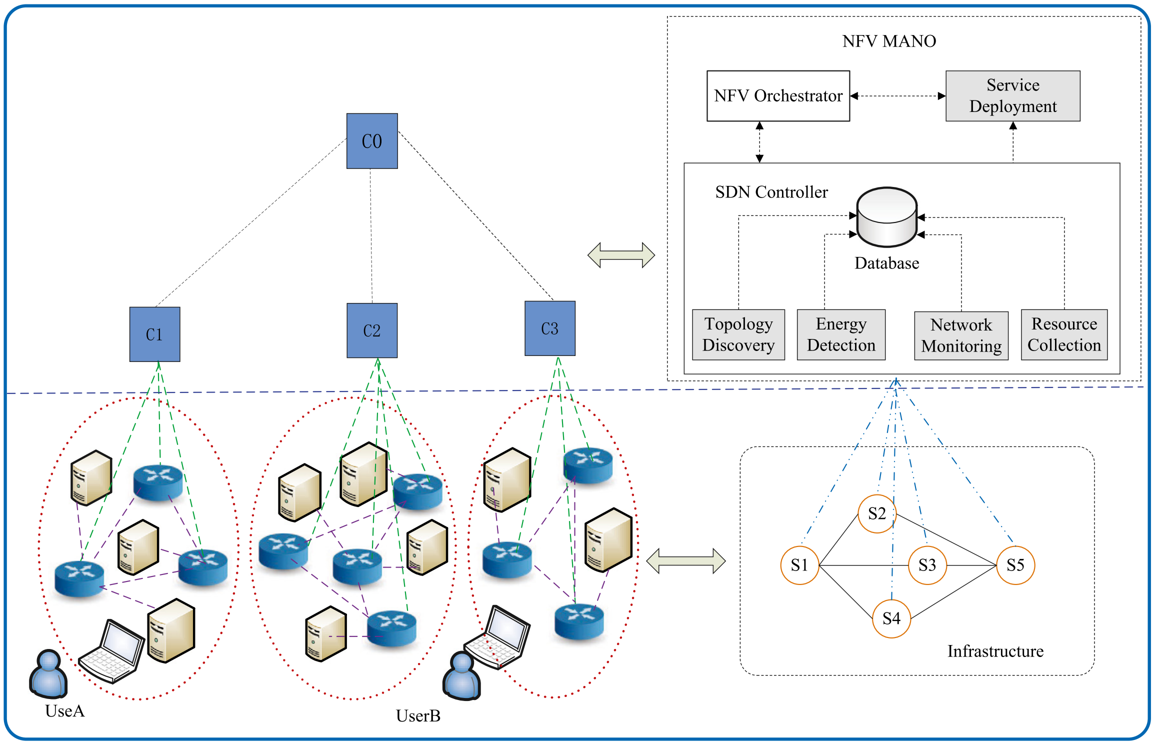


Fig. 1. System framework.

The above attempts address the load balancing problem efficiently. However, their researches are limited to single domain networks and ignore energy efficiency optimization. Different from the above works, this paper tries to optimize SFC deployment across multiple SDN domains with the aim of energy saving while achieving load balancing of multi-domain networks. In particular, the load balancing degree is proposed as the load balancing metric. The load balancing degree represented by CPU utilization ratio is defined as the variance of domain loads, described as in Eq. (6). When the VNFs are deployed in different SDN domains more evenly, the load balancing degree becomes smaller; otherwise, the VNFs are centrally deployed in some specific SDN domains, the load balancing degree becomes larger.

With regard to the multi-domain SFC deployment, there are few related researches at present. Our previous work in [25] focused on SFC deployment in multi-domain SDN networks to optimize service cost. However, we do not consider optimizing energy efficiency in [25]. Sun et al. [26] formulated the online multi-domain SFC provisioning problem as an ILP model to minimize energy consumption, and presented a heuristic algorithm to solve it. Xu et al. [27] proposed a hierarchical control architecture for multi-domain service function chaining, and designed an energy aware SFC placement algorithm and an energy aware migration algorithm to reduce energy consumption. Their main idea is to occupy as few resources as possible to accommodate as many as service request. Kaur et al. [28] established a multi-objective optimization model with the target of the maximum deployment of VNFs and the minimum energy consumption taking into account resource constraints, and presented an improved NSGA-II algorithm to solve it.

Similarly, we study the multi-domain SFC deployment problem, and establish a multi-objective optimization model. However, different from their optimization objectives, we aim to jointly optimize energy consumption and load balancing of multi-domain networks by considering resource allocation. To solve the optimization model, we present a heuristic service deployment algorithm.

3. System framework

The proposed energy efficient multi-domain network service deployment framework is illustrated in Fig. 1. Similar to [19], the overall framework consists of two parts, i.e., orchestration part and network infrastructure. In the orchestration part, we only consider VNF Management and Orchestration (MANO). It is composed of three components (i.e., NFV Orchestrator component, Service Deployment component and SDN Controller component) that interact with each other to perform the SFC deployment.

(1) **NFV Orchestrator component**

It corresponds to the NFV Orchestrator (NFVO) component in the NFV architectural framework defined by European Telecommunications Standards Institute (ETSI) [4], and manages the VNF lifecycle. It is responsible to receive service requests from the SDN controller component and obtain the SFC deployment solution by communicating with the service deployment component. After that, it sends the multi-domain SFC deployment solution to the SDN controller component for service provisioning.

(2) **SDN Controller component**

The SDN controllers can be divided into two categories. One is the centralized controller (e.g., $C0$ in Fig. 1) and the other is the domain controller (e.g., $C1$, $C2$ and $C3$ in Fig. 1). The whole network is composed of multiple SDN domains, for example, there are three SDN domains in Fig. 1. All the domain controllers are controlled and managed by the centralized controller. The domain controller is responsible to control and manage all the underlying network devices in each domain including forwarding devices and physical servers. The former is responsible to forward data flow, and the latter is used to instantiate the VNFs.

To support energy efficient service provisioning, depicted in Fig. 1, four functionality modules (i.e., topology discovery module, energy detection module, network monitoring module and resource collection module) and one knowledge database are added into the SDN controller component. Their main functions are described as follows.

- **Topology discovery module**: It can get the global topology view of multi-domain networks to control all the underlying network devices. The obtained topology is stored in the knowledge database, and used to compute the optimal deployment location of each VNF.

- **Energy detection module**: It is responsible to detect the energy consumption of different servers in each SDN domain. The energy consumption information of each SDN domain is recorded in the knowledge database, and used to realize energy saving and load balancing of multi-domain networks.
- **Network monitoring module**: It is responsible to monitor the change of network status and workload of each server. Once network status or server workload changes, they are stored in the knowledge database, and used to calculate an optimal deployment solution for each service request.
- **Resource collection module**: It is responsible to collect resource information of each server. It can provide useful resource information to guide the resource allocation for the multi-domain SFC deployment.
- **Knowledge database**: It is responsible to gather the information on different types of VNFs and their resource requirements, energy consumption of each server, network status, server load, etc. Such information is useful to achieve the energy efficient multi-domain SFC deployment.

In summary, the SDN controller component is responsible to discover network topology, detect server energy change, monitor network status and collect resource capacity information. It sends these information to the service deployment component. On the other hand, it receives service requests from end users (e.g., $UserA$ and $UserB$ in Fig. 1) and forwards the service requests to the NFV orchestrator component. Moreover, it updates flow tables in forwarding devices based on the multi-domain SFC deployment solution from the NFV orchestrator component.

(3) **Service deployment component**

To deploy the network service, the NFV orchestrator component sends the SFC and its resource demand to the service deployment component. The service deployment component determines the optimal servers to deploy all the VNFs by using the proposed heuristic service deployment algorithm. In order to achieve energy efficient network service deployment, the service deployment component requires real time resource and workload status information for the optimal server selection. The information can be obtained via the SDN controller component. Upon determining the optimal server candidates, the service deployment component sends the service deployment solution to the NFV orchestrator component.

The proposed multi-domain service deployment framework is aligned with the NFV framework proposed by ETSI, which is composed of NFV Infrastructure (NFVI), VNFs and NFV management and Orchestration (MANO) [11]. These components can be also identified in our proposed framework. In the proposed service deployment framework, NFVI refers to the combination of hardware and software resources distributed in multiple SDN domains. VNFs refer to the virtual network functions deployed on virtual machines, running on the physical servers. NFV MANO covers the lifecycle management of physical resources as well as the management and orchestration of VNF instances. In the proposed service deployment framework, NFV MANO is mainly implemented by the NFV orchestrator component, the service deployment component and the SDN controller component. The NFV orchestrator component performs the same functionalities as the NFV orchestrator defined by ETSI. The service deployment component is responsible to determine the service deployment solution. In fact, the service deployment solution can be also determined by the SDN controller or the NFV orchestrator. However, to highlight the importance of energy efficiency SFC deployment function, we set the service deployment component as a single component. The SDN controller component is responsible to maintain the interaction between VNFs and network resource.

Moreover, the Virtualized Infrastructure Manager (VIM) and VNF manager (VNFM) in NFV MANO proposed by ETSI remain the same in our proposed framework. Although the VIM proposed by ETSI is responsible to manage network resources, the SDN controller in our proposed framework plays the role of resource collection and allocation by leveraging its centralized control capacity.

The main innovation of the proposed system framework is that we integrate NFV technology and SDN architecture to realize the optimal deployment of multi-domain network service (in forms of SFC) in an energy efficient manner. Specifically, the global network topology view, resource status and energy consumption information of each SDN domain can be used to guide the SFC deployment in multi-domain networks. It can achieve energy consumption minimization of server nodes and load balancing degree minimization by taking advantages of the global network knowledge and centralized decision making mechanism of SDN.

The specific workflow of the proposed multi-domain service deployment framework is described as follows.

(1) When an end user enters multi-domain networks, it sends a service request to the SDN controller for service provisioning.

(2) The SDN controller forwards the service request to the NFV orchestrator, and the NFV orchestrator transforms this service request to a SFC.

(3) The NFV orchestrator sends resource demands of the SFC to the service deployment component.

(4) The service deployment component obtains the current global network resource and energy status by communicating with the SDN controllers.

(5) According to current resource status and server workload information, the service deployment component determines the appropriate server candidates by executing the proposed heuristic service deployment algorithm.

(6) The service deployment component sends the SFC placement solution to the NFV orchestrator, and the NFV orchestrator forwards it to the SDN controller.

(7) The CPU, storage and memory resources are allocated to the corresponding servers according to resource demand of VNFs, and flow tables in the underlying forwarding devices are updated.

(8) When network status or server workload changes, they are recorded in the knowledge database.

4. System model and problem formulation

In this section, we introduce the system model and problem formulation. To facilitate the understanding, some key mathematical symbols used in the paper are summarized in Table 1.

4.1. Assumptions

We make four assumptions on the multi-domain network service deployment problem as follows. Firstly, multiple VNFs can be deployed on the same server node, but one VNF cannot be split to deploy on multiple servers. Secondly, we assume that the resource capacity of each serve is limited, but the multi-domain networks have abundant resource capacities for each service request. Thirdly, we only take into account three kinds of resources (i.e., CPU, storage and memory resource) for server resources. Finally, we assume bandwidth and delay demand of each service request can be satisfied. Therefore, bandwidth and delay demand of each service request is not considered in this paper.

4.2. System model

4.2.1. Network model

The multi-domain physical network is modeled as an undirected graph $G = (V, E)$, where V is the set of physical nodes, and L is the set of physical links. The nodes are divided into two categories, i.e., forwarding nodes (e.g. switch) and server nodes, denoted as V_{FN} and V_{SN}, respectively. Then, V can be expressed as $V = V_{FN} \bigcup V_{SN}$.

Table 1
Summary of key mathematical symbols.

Notation	Description
G	The multi-domain physical network
G_i	The ith SDN domain
N_G	The total number of SDN domains
V	The set of nodes
V_{FN}	The set of forwarding nodes
V_{SN}	The set of server nodes
V_{SN}^i	The set of server nodes in G_i
$V_{SN}^{i,j}$	The jth server node in G_i
$rp_{cpu}^{i,j}$	The CPU capacity of server $V_{SN}^{i,j}$
$rp_{stor}^{i,j}$	The storage capacity of server $V_{SN}^{i,j}$
$rp_{mem}^{i,j}$	The memory capacity of server $V_{SN}^{i,j}$
$bw_{i,m,j,n}$	The bandwidth resource capacity of link $e_{i,m,j,n}$
f	The set of different types of VNFs
f_i	The ith type of VNF
rd_{cpu}^i	The CPU demand of VNF f_i
rd_{stor}^i	The storage demand of VNF f_i
rd_{mem}^i	The memory demand of VNF f_i
F	The set of service requests
F_i	The ith SFC
N_F	The total number of SFC
$F_{i,j}$	The jth VNF in F_i
$x_{mn}^{i,j}$	A binary variable. 1 if $F_{i,j}$ is deployed on server $V_{SN}^{m,n}$; 0 otherwise
$y_{i,j}^k$	A binary variable. 1 if $F_{i,j}$ belongs to f_k; 0 otherwise

With respect to each SDN domain, let $G^i = (V^i, E^i)$ denote the ith SDN domain, where V^i and E^i represent the set of physical nodes and links in the ith SDN domain, respectively. Thus, $G = \bigcup_i^{N_G} G^i$, where N_G is the total number of SDN domains. The set of server nodes in domain G^i is defined as V_{SN}^i, and the jth server in domain G^i is denoted by $v_{SN}^{i,j} \in V_{SN}^i \subseteq V_{SN}$.

The servers in different SDN domains have different resource capacities, including computing, storage and memory capacity. Therefore, each server $v_{SN}^{i,j}$ is associated with three attributes $rp_{cpu}^{i,j}$, $rp_{stor}^{i,j}$ and $rp_{mem}^{i,j}$ to represent three available resource capacities, respectively.

4.2.2. Service request model

Let $f = \{f_1, f_2, f_3, \ldots, f_{N_f}\}$ be the set of different types of VNFs, where f_i is the ith type of VNF, and N_f is the total number of all the VNF types. Different types of VNFs have different resource demands. Running the VNF consumes various types of server resources. Thus, we define CPU, storage and memory resource demand of VNF f_i by rd_{cpu}^i, rd_{stor}^i and rd_{mem}^i, respectively.

We define $F = \{F_1, F_2, F_3, \ldots, F_{N_F}\}$ to identify the set of service requests, where $F_{i,j} \in F_i \in F$ denotes the jth VNF in the ith SFC F_i, and N_F denotes the total number of service requests.

4.3. Problem formulation

4.3.1. Decision variables

The following two decision variables are defined for the multi-domain SFC deployment decision.

$x_{m,n}^{i,j}$: A binary variable. 1 if VNF $F_{i,j}$ is deployed on server $V_{SN}^{m,n}$; 0 otherwise.

$y_{i,j}^k$: A binary variable. 1 if VNF $F_{i,j}$ belongs to f_k; 0 otherwise.

4.3.2. Energy consumption

In most existing works, the energy consumption is generally composed of energy caused by servers and link energy consumption. However, according to [18], the link energy consumption only accounts for a small proportion of total energy consumption [18]. Therefore, some researches, e.g. [16,18,19,21,26,31], only focus on optimizing server energy consumption in the network service function chain deployment. Similarly, we only consider the energy consumption caused by server nodes in this paper. In multi-domain networks, there exist several heterogeneous clusters that use different kinds of servers with different energy consumption. However, according to [31], the energy consumption of a physical server increases linearly depending on the CPU utilization of the server. Thus, we make the assumption that the energy consumption of the server is proportional to the total number of CPU resources required by the VNFs.

There are three kinds of working states for a server, i.e., in idle mode, or in active mode, or in off mode. In off mode, the server is shut down and stops working without any energy consumption. In idle mode, the server waits for working and generates some energy even if it does not host any VNFs. Once a VNF is deployed on the server, the server is transformed from idle state to active state to run the VNF. With the increase of server workload, the server will generate more energy consumption. The energy consumption of server $v_{SN}^{m,n}$ can be expressed as follows.

$$e_{m,n} = \begin{cases} e_{idle}^{m,n} & \text{in idle mode} \\ e_{idle}^{m,n} + e_{active}^{m,n} & \text{in active mode} \\ 0 & \text{in off mode} \end{cases} \tag{1}$$

$$e_{active}^{m,n} = p_{m,n} \cdot e_{activemax}^{m,n} \tag{2}$$

$$p_{m,n} = \frac{cp_{m,n}}{cp_{max}^{m,n}} \tag{3}$$

$$cp_{m,n} = \sum_{F_{i,j} \in F_i} \sum_{F_i \in F} \sum_{f_k \in f} x_{m,n}^{i,j} \cdot y_{i,j}^k \cdot rd_{cpu}^k \tag{4}$$

where $e_{idle}^{m,n}$ is the energy consumed by server $v_{SN}^{m,n}$ in idle state. $e_{active}^{m,n}$ is the energy consumption caused by CPU resource consumption in active state. $p_{m,n}$ is the CPU utilization of server $v_{SN}^{m,n}$. $cp_{m,n}$ is the total number of CPU resources consumed by server $v_{SN}^{m,n}$. $cp_{max}^{m,n}$ is the maximum CPU resource capacity which server $v_{SN}^{m,n}$ consumes. $e_{activemax}^{m,n}$ is the maximum energy caused by server $v_{SN}^{m,n}$ consuming CPU resource.

Thus, the total energy consumption generated by all the SFCs F can be formulated as

$$EC = \sum_m \sum_n e_{m,n} \tag{5}$$

4.3.3. Load balancing degree

In order to evaluate the load balancing of multi-domain networks, we define the concept "load balancing degree". We use the CPU utilization ratio of the server to represent the load balancing degree, which can be computed by Eq. (4). The smaller the load balancing degree is, the better load balancing effect is. It can be observed from Eq. (4) that its value is between 0 and 1. When the load balancing degree is close to 1, it indicates that the multi-domain network system is in serious load imbalance. When the load balancing degree is close to 0, it indicates that the multi-domain network system is in near optimal load balancing.

$$LBD = \frac{1}{N_G} \sqrt{\sum_{m=1}^{N_G} \| ld_m - \bar{ld} \|^2} \tag{6}$$

$$\bar{ld} = \frac{1}{N_G} \sum_{m=1}^{N_G} ld_m \tag{7}$$

$$ld_m = \frac{1}{N_{SN}^m} \sum_{n=1}^{N_{SN}^m} p_{m,n} \tag{8}$$

where $\bar{ld}$ is the average server load of all SDN domains. N_G is the total number of SDN domains. ld_m is the total loads of servers in domain G^m. N_{SN}^m is the total number of servers in domain G^m.

4.3.4. The multi-objective optimization model

Our aim is to minimize the total energy consumption cost while achieving the load balancing of multi-domain networks. Hence, the proposed multi-objective optimization model is formulated as follows.

$$min \quad EC \tag{9}$$

$$min \quad LBD \tag{10}$$

$$s.t. \begin{cases} \sum_m \sum_n x_{m,n}^{i,j} = 1 & \text{(a)} \\ \sum_{F_{i,j} \in F_i} \sum_{F_i \in F} \sum_{f_k \in f} x_{m,n}^{i,j} \cdot y_{i,j}^k \cdot rd_{cpu}^i \leq rp_{cpu}^{m,n} & \text{(b)} \\ \sum_{F_{i,j} \in F_i} \sum_{F_i \in F} \sum_{f_k \in f} x_{m,n}^{i,j} \cdot y_{i,j}^k \cdot rd_{stor}^i \leq rp_{stor}^{m,n} & \text{(c)} \\ \sum_{F_{i,j} \in F_i} \sum_{F_i \in F} \sum_{f_k \in f} x_{m,n}^{i,j} \cdot y_{i,j}^k \cdot rd_{mem}^i \leq rp_{mem}^{m,n} & \text{(d)} \\ x_{m,n}^{i,j} \in \{0,1\} & \text{(e)} \\ y_{i,j}^k \in \{0,1\} & \text{(f)} \end{cases} \tag{11}$$

Constraint (11)(a) ensures that each VNF should be deployed on just one server node instead of multiple server nodes. For each server, it needs to consume enough resources to run the deployed VNFs for service provisioning. Thus, constraint (11)(b) ensures that the CPU resource allocated to the server should meet the CPU demand of the deployed VNFs. Similarly, constraints (11)(c) and (11)(d) guarantee that the server has enough storage and memory resources to run the VNFs, respectively. Moreover, the two decision variables should satisfy the integrality constraints in formulas (11)(e) and (11)(f).

5. The heuristic deployment algorithm

In order to address the above optimization problem, we proposed a novel heuristic network service deployment algorithm. In this section, we describe the proposed heuristic algorithm and analyze its complexity.

5.1. Algorithm description

High energy consumption and severe load imbalance result in server performance degradation and affects user's service level agreements. Therefore, in order to avoid excessive energy consumption of each server $V_{SN}^{m,n}$, a maximum energy consumption threshold value (E_{max}) is set. Specifically, when we deploy a VNF $F_{i,j}$ in multi-domain networks G, we select the server $V_{SN}^{m,n}$ with the minimum additional energy consumption cost to deploy each VNF $F_{i,j}$. If the total energy consumption generated by the selected server $V_{SN}^{m,n}$ exceeds the set threshold value E_{max}, we re-select another suitable server to deploy the VNF $F_{i,j}$.

On the other hand, in order to achieve the load balancing of multi-domain networks G, a maximum workload threshold value (L_{max}) and a minimum workload threshold value (L_{min}) are set to avoid each server $V_{SN}^{m,n}$ from getting overloaded and underloaded, respectively. Specifically, when we select server $V_{SN}^{m,n}$ for a VNF $F_{i,j}$, we judge whether its workload is higher than the set threshold value L_{max} or not. If yes, we will re-select another suitable server to deploy the VNF. After completing the SFC deployment, we make SFC deployment adjustment to reduce the number of used servers. Specifically, the servers whose workload is smaller than the set threshold value (L_{min}) are switched off, and all the VNFs deploying on the underloaded servers are re-deployed on the other appropriate servers.

Moreover, we deploy the VNFs in all the SDN domains as uniformly as possible by means of the round-robin scheduling algorithm. Specifically, at different moments, each SDN domain G^i has different number of deployed VNFs and different server workloads. In order to keep the load balancing of multi-domain networks G, we set that the probability of it being selected for next VNF deployment is smaller if an SDN domain has bigger workload. Upon determining a candidate SDN domain, we select an optimal server to deploy the VNF.

The main idea of the proposed heuristic service deployment algorithm is that we deploy the VNFs on the servers which can generate the smallest additional energy consumption to reduce the total server energy consumption. Meanwhile, we deploy them into all the SDN domains as uniform as possible to achieve the load balancing of multi-domain networks. The pseudo code of the proposed heuristic service deployment algorithm is described in Algorithm 1.

Algorithm 1 The Heuristic Network Service Deployment Algorithm

```
INPUT: G: The multi-domain networks
       F: The set of service requests
OUTPUT: EC: The total energy consumption
        LBD: The load balancing degree
BEGIN
01: Sort all the VNFs in descending order by CPU demand;
02: i = 0;
03: for each VNF vf ∈ F, do
04:    Select an SDN domain D_i by using the round-robin scheduling algorithm;
05:    s = SelectOptimalServer(D_i, vf);
06:    if s! = NULL then
07:       Deploy VNF vf on server s;
08:       Update the energy cost and load of server s by Eq. (1);
09:       Delete VNF vf from F;
10:    else
11:       goto step 4;
12:    end if
13: end for
14: LES = SelectLowEnergyServer(G);
15: LLS = SelectLowLoadServer(G);
16: for each server ls ∈ LLS, do
17:   for each VNF lvf deployed on server ls, do
18:       mes = SelectMinadditonalEnergyServer(LES);
19:      if E_mes ≤ E_max and L_mes ≤ L_max, then
20:         Deploy lvf on server mes;
21:         Update the energy cost and load of server mes by Eq. (1);
22:      end if
23:    end for
24: end for
25: Switch all the servers in idle mode into off mode;
26: Calculate the total energy consumption by Eq. (5);
27: Calculate the load balancing degree by Eq. (6);
END
```

In the initial stage, all the VNFs are sorted in descending order by their corresponding CPU demand rd_{cpu}^i, illustrated in line 1.

In the second stage, each non-deployed VNF is repeated to execute the following procedures until all the VNFs are successfully deployed in multi-domain networks, illustrated lines 2–13.

(1) As illustrated in line 4, we execute the round-robin scheduling algorithm to select a candidate SDN domain D_i for the un-deployed VNF vf.

(2) Next, we select the optimal server s from the selected SDN domain D_i for VNF vf by executing the optimal server selection algorithm (Please See Algorithm 2), described in line 5.

(3) If the optimal server cannot be found from the current SDN domain D_i due to lack of CPU resources, we continue to search from the other SDN domains by leveraging the round-robin scheduling algorithm until we find the optimal server s. Then, we deploy VNF vf on the selected server s, described in line 7.

(4) After deploying VNF vf on server s, we update server's (s) total energy consumption and workload by Eq. (1), described in line 8, and delete VNF vf from F, described in line 9.

In the third stage, we make the multi-domain SFC deployment adjustment after deploying all the SFCs in multi-domain networks G, described in lines 14–24. Specifically, we first determine the set (LES) of the servers whose energy consumption is less than E_{max}, and the set (LLS) of the servers whose workload is less than L_{min}, respectively.

And then, we iteratively execute the following operations for each VNF lvf, which is deployed on the underloaded server ls in sequence.

(1) We search for the optimal server mes from LES, which satisfies the resource demand of VNF lvf and generates the minimum additional energy consumption for VNF lvf deployment, described in line 18. If there exist appropriate servers, we will randomly select one server from the candidate servers.

(2) If server mes satisfies the maximum energy consumption constraint and the maximum load constraint, we deploy VNF lvf on server mes, and allocate the corresponding resource to server mes, described in lines 19–20.

(3) We update the total energy consumption and workload of server mes by Eq. (1), described in line 21.

Next, we turn off all the servers on which no VNF is deployed, to reduce the total energy consumption of multi-domain networks, described in line 25. Finally, as described in lines 26–27, we compute the total server energy consumption by Eq. (5) and load balancing degree of multi-domain networks by Eq. (6), respectively.

Algorithm 2 The Optimal Server Selection Algorithm

INPUT: D_i: The ith SDN domain
vf: Virtual network function
OUTPUT: s_{vf}: The optimal server
BEGIN
01: Determine the type k of VNF vf by constraint (11)(f);
02: Get the resource demand $(rd^k_{cpu}, rd^k_{stor}, rd^k_{mem})$ of VNF vf;
03: Select the servers SR which satisfy resource constraints (11)(b)–(11)(d) from V^i_{SN};
04: $SE = empty$;
05: $j = 1$;
06: **for** each server $rs \in SR$, **do**
07: Calculate the total energy cost TE_{rs} of server rs by Eq. (1);
08: Calculate the additional energy cost AE_{rs} of server rs;
09: Calculate the total load TL_{rs} of server rs;
10: **if** $TE_{rs} \leq E_{max}$ and $TL_{rs} \leq L_{max}$, **then**
11: $SE[j] \leftarrow (rs, AE_{rs})$;
12: $j = j + 1$;
13: **end if**
14: **end for**
15: Select optimal server s_{vf} with least additional energy cost from SE;
END

Algorithm 2 illustrates the optimal server selection procedure for a VNF vf. First of all, we determine the type k of VNF vf by Eq. (11)(f), and obtain CPU, storage and memory demand $(rd^k_{cpu}, rd^k_{stor}, rd^k_{mem})$ of VNF vf, described in lines 1–2. Next, we select the servers SR from all the servers V^i_{SN} in current SDN domain D_i, which meet the resource demand of VNF vf by constraints (11)(b)–(11)(d). Then, we execute the following operations for each selected server rs in SR iteratively.

In lines 4–15, we select the optimal server from SR according to energy consumption cost and the total server load. Here, we use SE (described in line 4) to store the candidate server nodes and their corresponding additional energy consumption. Then, the following operation is completed for each server in SR:

(1) For a specific server rs, by Eq. (1), we first calculate its total energy consumption TE_{rs} when VNF vf is deployed on server rs, described in line 7.

(2) We compute its additional energy consumption AE_{rs} caused by the VNF vf deploying on server rs, described in line 8.

(3) We compute its total workload TL_{rs} when VNF vf is deployed on server rs, described in line 9.

(4) If the total energy consumption and total workload of server rs are smaller than the maximum energy consumption E_{max} and the maximum load threshold L_{max} simultaneously, server rs and its additional energy consumption AE_{rs} caused by VNF vf are recorded in the candidate server set (SE), illustrated in lines 10–13.

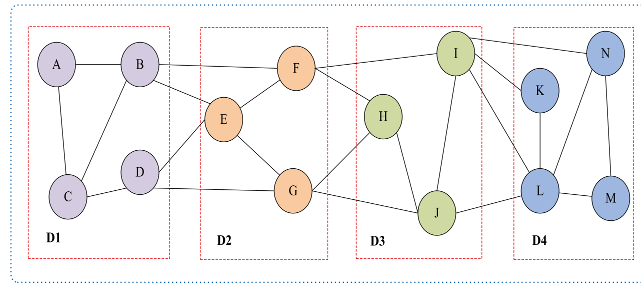


Fig. 2. A simple example.

Finally, we choose an optimal server s_{vnf} with the minimum additional energy consumption from the candidate server set SE, to deploy VNF vf, illustrated in line 15.

5.2. A simple example

In order to facilitate the understanding of the proposed heuristic service deployment algorithm, we give a simple example as follows.

As shown in Fig. 2, there are four SDN domains (i.e., $D1 \sim D4$) and fourteen servers (i.e., $A \sim N$) in multi-domain networks. With respect to the service requests, we assume that there are three different SFCs, denoted by $F1 = \{f_1, f_3, f_5\}$, $F2 = \{f_2, f_4, f_5\}$ and $F3 = \{f_1, f_2, f_4\}$, respectively. The specific deployment workflow of three SFCs is described as follows.

(1) We sort all the VNFs (i.e., $f_1 \sim f_5$) by the CPU demands of different types of VNFs in descending order. We assume that the got consequence is (f_1, f_2, f_3, f_4 and f_5) after sorting the VNFs.

(2) Based on CPU resource demand sorting, we determine the set of all the VNFs that needs to be deployed, denoted by $Deps = \{F_{11}, F_{31}, F_{21}, F_{32}, F_{12}, F_{22}, F_{33}, F_{13}, F_{23}\}$ where F_{ij} is jth VNF in SFC F_i.

(3) For VNF F_{11}, we select an SDN domain (e.g., $D2$) by leveraging the round-robin scheduling algorithm.

(4) For each server (i.e., $E \sim G$) in the selected SDN domain $D2$, we calculate its total energy consumption, the additional energy consumption caused by VNF F_{11} and the total workload.

(5) We select an optimal server (e.g., F) for VNF F_{11}, which satisfies the maximum energy consumption threshold E_{max} and maximum load threshold L_{max}, and has the minimum additional energy consumption.

(6) If there is no an appropriate server in the selected SDN domain $D2$, we re-select another appropriate SDN domain by executing the round-robin scheduling algorithm, and determine the optimal server candidate from the recent selected SDN domain by executing the same steps mentioned above again.

(7) For all the remaining undeployed VNFs, we select an SDN domain randomly and determine the optimal server for each VNF by following the same step mentioned above. Here, we assume that they are deployed on servers C, I, M, J, G, B, E and K, respectively.

(8) From the selected servers on which the VNFs are deployed, we select the servers with small workload and the servers with low energy consumption, and re-deploy the VNFs on the former into the latter. We assume that the server E is in low load and the server G has small energy consumption. Then, we re-deploy VNF F_{13} (deploying on server E) on server G.

(9) We switch all the servers (i.e., A, D, E, H, L and N) which are in idle mode into off mode.

5.3. Complexity analysis

The proposed heuristic service deployment algorithm mainly involves two important procedures, i.e., SFC deployment (Lines 1–13 in

Algorithm 1) and SFC deployment adjustment (Lines 14–24 in Algorithm 1). The time complexity and space complexity are analyzed as follows.

We assume that the total number of VNFs in all the SFCs is N_{vnf}. The total number of VNF types is N_f. The maximum number of server nodes in each SDN domain is N_s. Therefore, the time complexity of Algorithm 2 is $T_{a2} = O(N_f + N_s)$, and the space complexity of Algorithm 2 is $S_{a2} = O(N_f + N_s)$.

The time complexity of the first procedure in Algorithm 1 is $T_{a11} = O(N_{vnf}^2) + O(N_{vnf} \cdot (N_f + N_s)) \sim O(N_{vnf}^2)$. Similarly, the space complexity of the first procedure in Algorithm 1 is $S_{a11} = O(N_{vnf}^2)$.

The time complexity of the second procedure in Algorithm 1 is $T_{a12} = O(N_s) + O(N_s^2 \cdot N_{vnf}) \sim O(N_s^2 \cdot N_{vnf})$, and the space complexity of the second procedure in Algorithm 1 is $S_{a12} = O(N_s^2 \cdot N_{vnf})$.

Thus, according to the above analysis, the total time complexity of the proposed heuristic service deployment algorithm (i.e., Algorithm 1) is $T_{a1} = O(N_s^2 \cdot N_{vnf} + N_{vnf}^2)$. The total space complexity of the proposed heuristic service deployment algorithm (i.e., Algorithm 1) is $S_{a1} = O(N_s^2 \cdot N_{vnf} + N_{vnf}^2)$.

5.4. Discussion

We assume that the total number of network domains is N_G, the total number of server nodes in multi-domain networks is N_{ser}, the total number of nodes in multi-domain networks is N_v, the total number of service requests is N_{sfc}. The time complexity of comparison algorithms is analyzed as follows.

The time complexity of the random deployment (RANP) algorithm [37] is $O(N_G \cdot N_s \cdot N_{vnf})$; the time complexity of first-fit deployment (FFP) algorithm [38] is $O(N_{ser} \cdot N_{vnf})$; the time complexity of energy aware SFC placement (EASP) algorithm [18] is $O(N_{ser} \cdot N_{vnf})$; the time complexity of multi-domain SDN SFC deployment (MDSP) algorithm [25] is $O(N_v^4)$; the time complexity of energy efficient multi-domain SFC provisioning (MDEP) algorithm [26] is $O(N_v^2 \cdot N_{sfc})$.

Compared with the above five algorithms, although the proposed heuristic deployment algorithm can efficiently reduce server's energy consumption and keep load balancing of multi-domain networks, its time complexity is higher. In particular, as the total number of VNFs and servers increases, the time cost of the proposed heuristic deployment algorithm becomes large.

6. Performance evaluation

In this section, we introduce simulation setup and evaluation metrics. And then, we analyze the simulation results.

6.1. Simulation setup

The simulation experiments are implemented using MATLAB, running on Windows 7 personal computer with Inter (R) Core(TM), 2.93 GHz CPU, 4GRAM. In order to simulate multi-domain networks, we select CERNET2 topology [39] and Interoute topology [40] as the test topologies, and randomly divide CERNET2 (CR) network and Interoute (IR) network into 4 and 10 domains, respectively. CERNET2 network is composed of 20 nodes and 22 links, and Interoute network is composed of 110 nodes and 148 links.

The main parameters used in the simulations is summarized in Table 2. In order to make our simulation scenarios more generic, similar to [41], "units" is adopted to quantify resource consumption and energy cost. With respect to resource capacity, the CPU, storage and memory resource capabilities of each server in CERNET2 network scenarios are a real number uniformly distributed between 15 and 20 units, respectively. While, the CPU, storage and memory resource capabilities of each server in Interoute network scenarios are uniformly distributed within [3, 5] units, respectively.

Table 2
Summary of simulation parameters.

Parameter	Value
rp_{cpu}^{ij} (CR)	$U(15, 20)$
rp_{stor}^{ij} (CR)	$U(15, 20)$
rp_{mem}^{ij} (CR)	$U(15, 20)$
rp_{cpu}^{ij} (IR)	$U(3, 5)$
rp_{stor}^{ij} (IR)	$U(3, 5)$
rp_{mem}^{ij} (IR)	$U(3, 5)$
$minE$	$U(5, 10)$
$maxE$	$U(40, 60)$
E_{max}	85%
L_{max}	85%
L_{min}	15%
N_f	6
rd_{cpu}^{i}	$U(0, 1.2)$
rd_{stor}^{i}	$U(0, 1.2)$
rd_{mem}^{i}	$U(0, 1.2)$
N_{F_i}	$U(2, 5)$
N_F	50

In both network scenarios, the minimum energy consumption ($minE$) of each server is assumed to be uniformly distributed within [5, 10] units, and the maximum energy consumption of each server ($maxE$) is uniformly distributed within [40, 60] units. The maximum energy consumption threshold (i.e., E_{max}) of each server is assigned to 85%. The maximum and minimum load thresholds (i.e., L_{max} and L_{min}) of each server are assigned to 85% and 15%, respectively.

Similar to [36], we assume that both network scenarios can support six VNF types, i.e., NAT, FW, IDS, Load Balancer (LB), WAN Optimization and Flow Monitor (FM). The CPU, storage and memory demands in each type of VNF are a real number uniformly distributed with [0, 1.2] units, respectively. In particular, to analyze the impact of resource demand per VNF on network service deployment scheme, the resource demand per VNF is divided into three categories, i.e, $D(0, 0.4)$, $D(0.4, 0.8)$ and $D(0.8, 1.2)$. $D(a, b)$ represents that the resource demand per VNF obeys the uniform distribution between a and b units. For example, $D(0.8, 1.2)$ denotes that the CPU, storage and memory resource demands per VNF are uniformly distributed within [0.8, 1.2], respectively.

Similar to [40], all the SFCs are generated randomly, and the number of VNFs in each SFC is uniformly distributed with [2, 5]. For service requests, we assume that the total number of SFCs is 50.

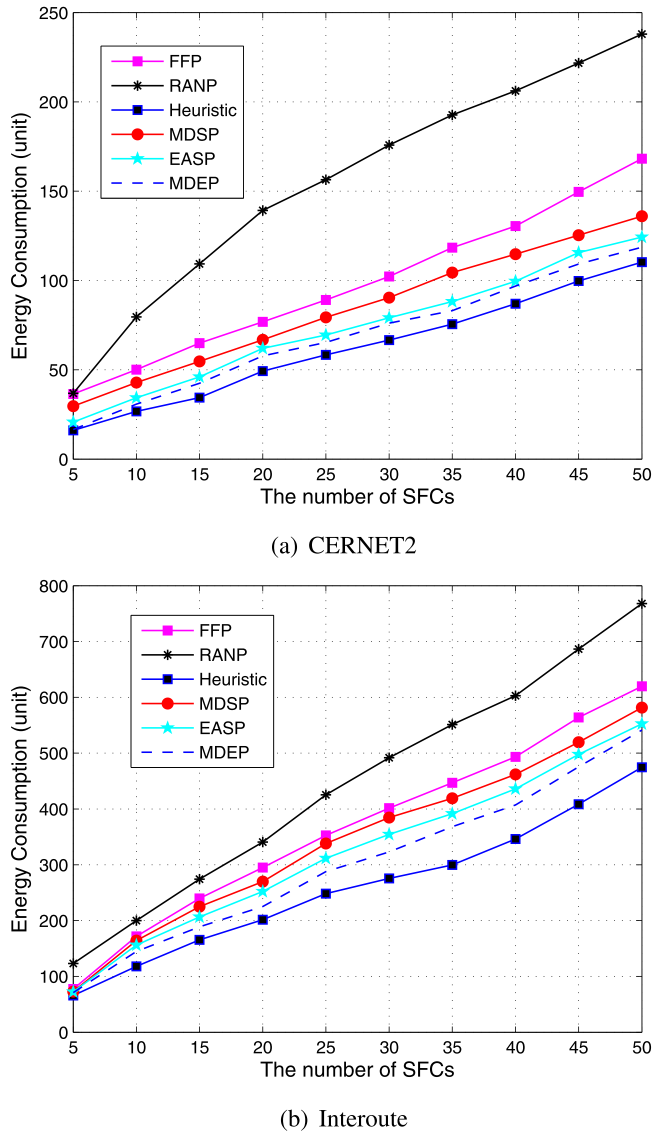
To evaluate the performance of the proposed heuristic deployment (Heuristic) algorithm, we select RANP algorithm, FFP algorithm, EASP algorithm, MDSP algorithm and MDEP algorithm as comparison algorithms. In particular, EASP algorithm is used to solve the problem of energy efficient network service deployment in single domain network scenarios. We conduct the comparisons between Heuristic algorithm and EASP algorithm to show that EASP algorithm is not suitable for solving our proposed energy efficient multi-domain network service deployment problem.

In order to obtain more exact simulation results, in our simulation experiments, each experiment is repeated 20 times, and the average is calculated as the final result. The confidence intervals for all the results are set to 95%.

6.2. Performance metrics

Three metrics including energy consumption, load balancing degree and time overhead are selected to comprehensively evaluate the performance of the proposed heuristic deployment algorithm.

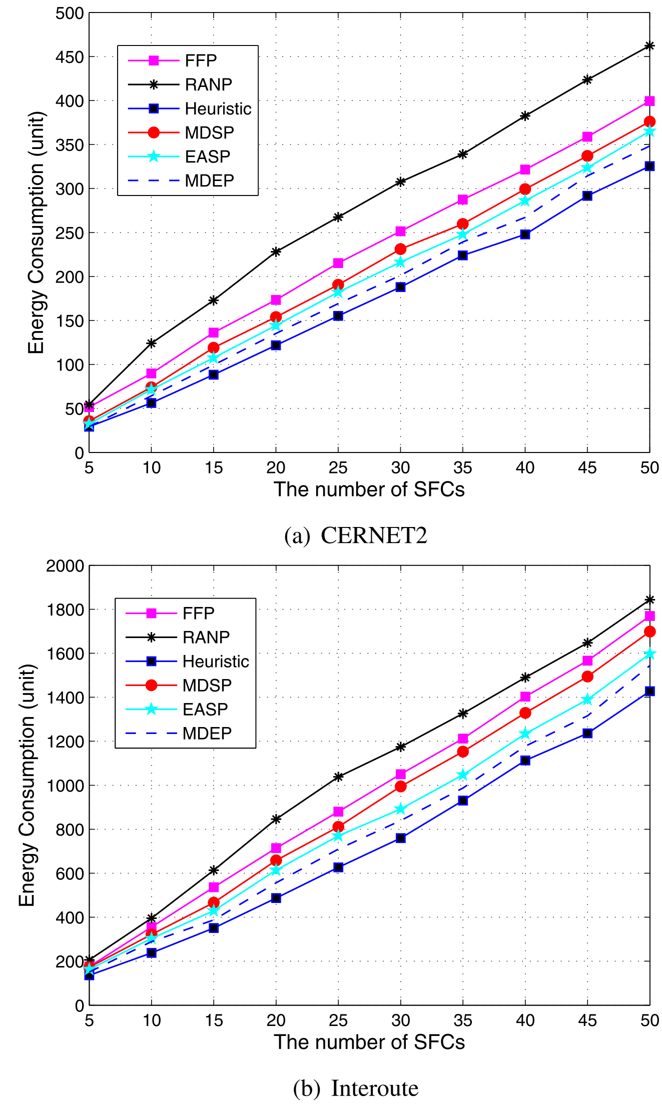
- **Energy consumption**: It refers to the total energy consumption of all the active server nodes, calculated by Eq. (5).
- **Load balancing degree**: It refers to the load balancing degree of multi-domain networks, calculated by Eq. (10).



(a) CERNET2

(b) Interoute

Fig. 3. Energy consumption [$D(0, 0.4)$].



(a) CERNET2

(b) Interoute

Fig. 4. Energy consumption [$D(0.4, 0.8)$].

- **Time overhead**: It refers to the time overhead spent by the service deployment algorithm deploying a set of SFCs in multi-domain networks.

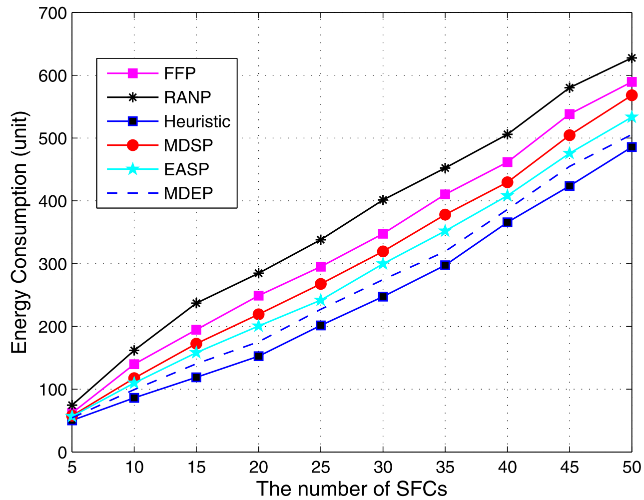
6.3. Simulation results

6.3.1. Energy consumption

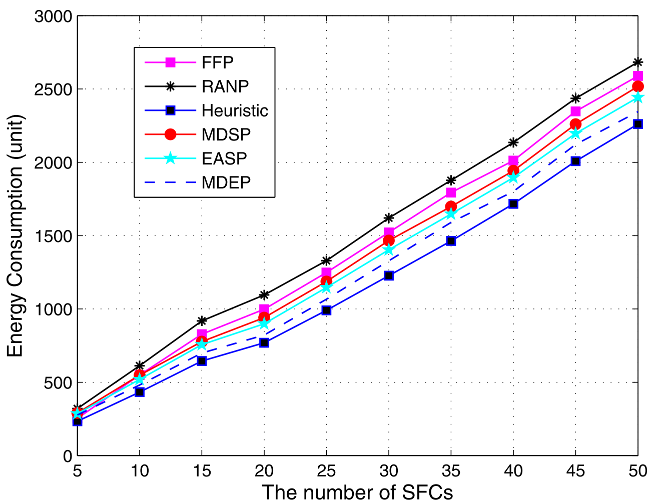
Figs. 3–5 show energy consumption comparison results of six deployment algorithms under different resource demands in both network scenarios, respectively. We can observe that with the increase of SFCs, the total energy consumption generated by six deployment algorithms becomes big. This is because as the number of SFCs increases, the total number of VNFs becomes bigger. Accordingly, more VNFs are deployed in multi-domain networks. To provide efficient service, the servers require more CPU resource, storage resource and memory resource to instantiate the deployed VNFs, further generating a lot of energy consumption. Similarly, it can be also observed from Figs. 3–5, with the growth of resource demand per VNF, the total energy consumption in both network scenarios shows a increasing tendency. This is because to instantiate the VNFs, the servers consume more resources, especially CPU resource.

Compared with five comparison algorithms, we can observe that the proposed heuristic deployment algorithm can produce the lowest energy consumption in both network scenarios. Meanwhile, compared with the MDSP algorithm, first-fit deployment algorithm and random deployment algorithm, the MDEP algorithm and EASP algorithm can produce smaller energy consumption. The detailed reasons are as follows.

In the first-fit deployment algorithm, each VNF is deployed on the server which is the first to satisfy its resource demand. However, in the random deployment algorithm, each VNF is randomly deployed on a server. Compared with the first-fit deployment algorithm, the random deployment algorithm can occupy more servers to deploy the VNFs, further resulting in higher energy consumption. The MDSP algorithm tries to deploy the VNFs in the same SFC in few SDN domains with small resource usage cost as possible. However, the first-fit deployment algorithm, random deployment algorithm and MDSP algorithm do not consider reducing energy consumption. Different from them, all of the other three algorithms regard energy saving as their optimization objectives. Although the EASP algorithm tries to save energy, it is limited to single domain network scenarios rather than multi-domain network scenarios. Its energy efficiency is lower than that of the heuristic deployment algorithm. The simulation results demonstrate that the

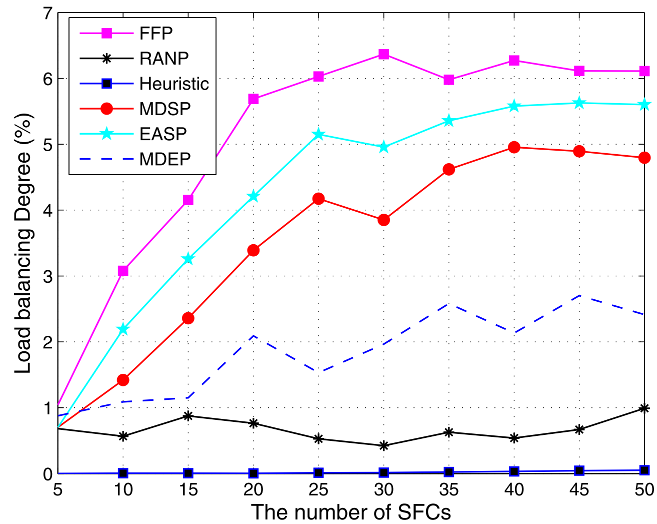


(a) CERNET2

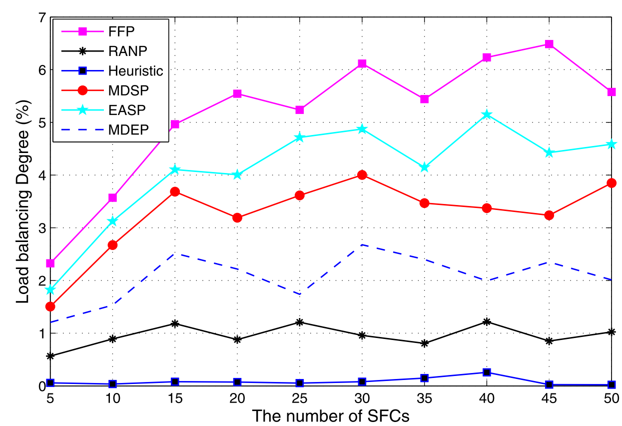


(b) Interoute

Fig. 5. Energy consumption [$D(0.8, 1.2)$].



(a) CERNET2



(b) Interoute

Fig. 6. Load balancing degree [$D(0, 0.4)$].

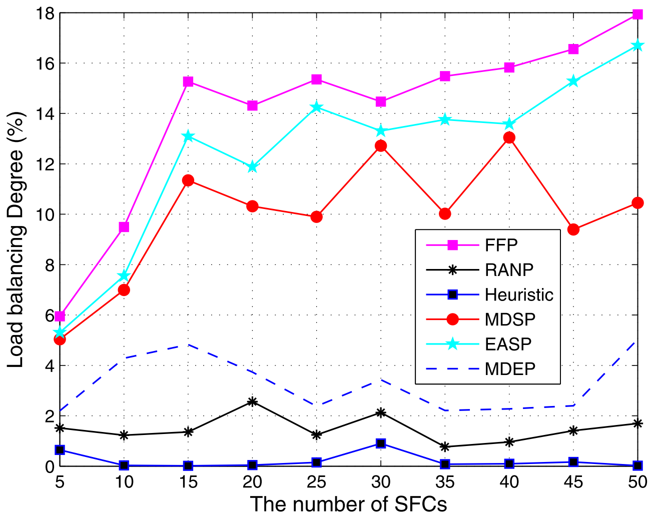
EASP algorithm is not suitable for solving the proposed problem in this paper. Different from the MDEP algorithms, the proposed heuristic deployment algorithm deploys each VNF on the server with the smallest additional energy consumption cost. Moreover, the proposed heuristic deployment algorithm further adjusts the SFC deployment and switches off all idle servers to reduce energy consumption.

In addition, by comparing CERNET2 and Interoute network scenarios, we can observe that the total energy consumption in CERNET2 network scenarios is less than that in Interoute network scenarios. This is because compared with CERNET2 network scenarios, Interoute network scenarios has a larger number of SDN domains and server nodes, and each server in it has smaller resource capacity. When we deploy the same number of VNFs with same resource demands in CERNET2 and Interoute network scenarios respectively, the total number of used servers in CERNET2 network scenarios is less than that in Interoute network scenarios.

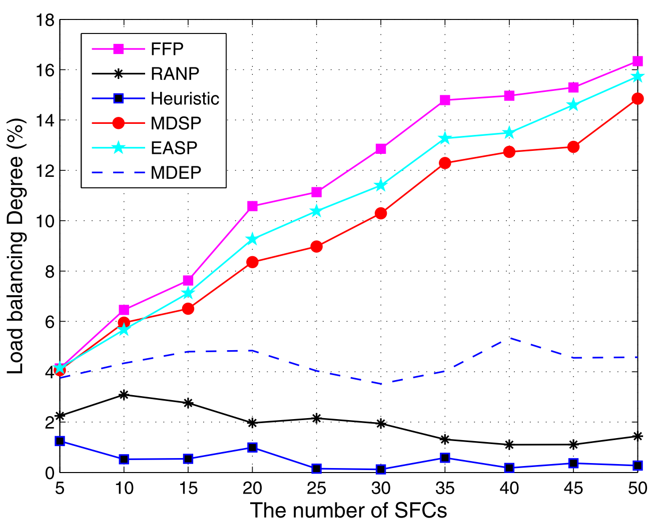
6.3.2. Load balancing degree

Figs. 6–8 show six different service deployment algorithms comparison results on load balancing degree under different resource demands, respectively. We can observe from Figs. 6–8 that, among six service deployment algorithms, the proposed heuristic deployment algorithm can generate lowest load balancing degree, and the first-fit deployment algorithm can generate the biggest. In our design, the smaller the load balancing degree is, the better the load balancing effect can be achieved. In other words, the proposed heuristic deployment algorithm can achieve better load balancing effect than the other five comparison algorithms. The detailed reasons are as follows.

Regarding with the first-fit deployment algorithm, it centrally deploy the VNFs on a limited number of server nodes. This easily results in serious server load imbalance. However, the random deployment algorithm randomly selects a server with satisfying each VNF's resource demand to deploy it. Compared with the first-fit deployment algorithm, the random deployment property can improve the load balancing effect of multi-domain networks. On the other hand, although the MDEP algorithm and EASP algorithm attempt to reduce energy consumption, they ignore the load balancing effect of multi-domain networks. Similar to the proposed heuristic deployment algorithm, the MDSP algorithm considers optimizing multi-domain SFC deployment. However, it does not consider optimizing load balancing of multi-domain networks. Different from five comparison algorithms, the proposed heuristic deployment algorithm deploys all the VNFs in all the SDN domains as uniformly as possible. More importantly, to achieve the load balancing

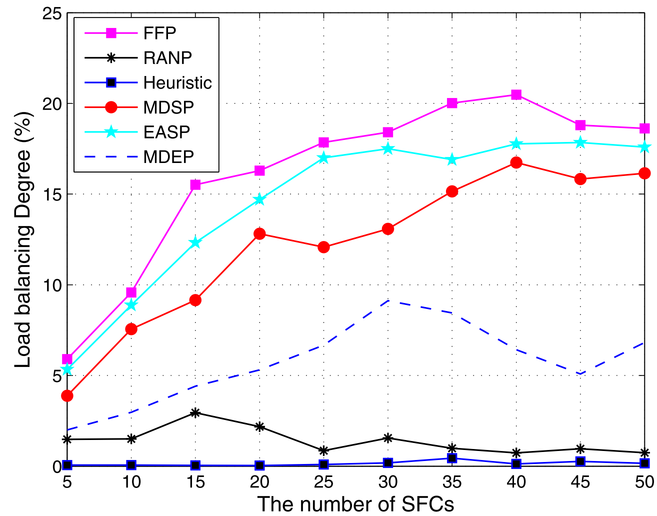


(a) CERNET2

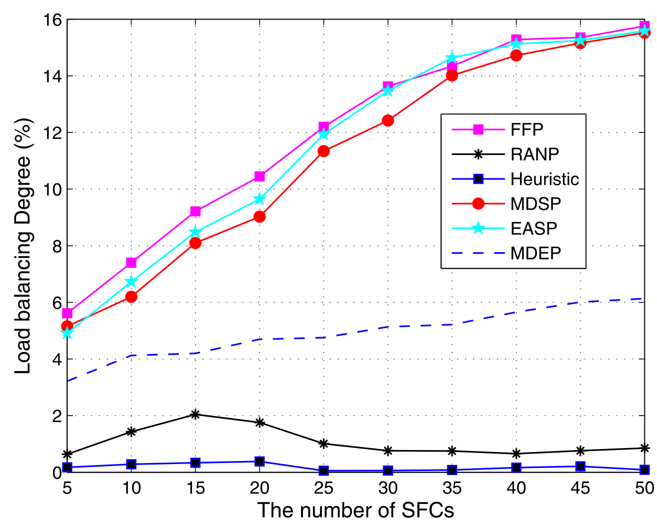


(b) Interoute

Fig. 7. Load balancing degree [$D(0.4, 0.8)$].



(a) CERNET2



(b) Interoute

Fig. 8. Load balancing degree [$D(0.8, 1.2)$].

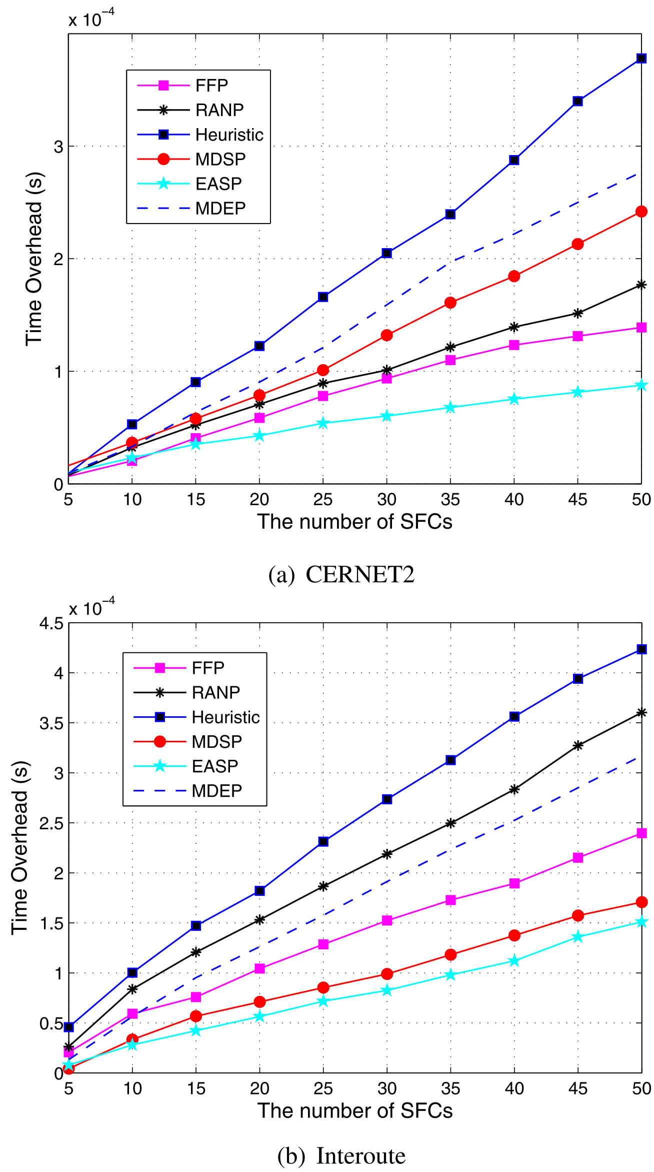
of multi-domain networks, it also sets the maximum and minimum load thresholds, respectively.

It can be also observed from Figs. 6–8 that six deployment algorithms show different load balancing degree under different resource demand. This is because each server node has different resource capacity (i.e., CPU, storage and memory), and each VNF is deployed on the server node which can satisfy the changing resource demand, rather than a fixed server node. With respect to different resource demands per VNF, the same deployment algorithm selects different server nodes in different SDN domains to deploy all the VNFs. Different deployment locations of the VNFs can make different effects on load balancing of multi-domain networks.

6.3.3. Time overhead

Figs. 9–11 show time overhead comparison results of six deployment algorithms, respectively. We can observe that as the total number of SFCs increase, the time overhead of each deployment algorithm shows an increasing tendency. This is because with the increase of SFCs, more VNFs are generated. To deploy the VNFs in multi-domain networks, each deployment algorithm requires more time to search the optimal servers for all the VNFs. Moreover, compared with five comparison algorithms, the proposed heuristic deployment algorithm generates higher time overhead. This is because compared with five comparison algorithms, the process procedure of the proposed heuristic deployment algorithm is more complex. Specifically, it first deploys all the VNFs into multi-domain networks by leveraging the optimal server selection algorithm. And then, it makes the SFC deployment adjustment wherein all the VNFs deployed on the servers with low load are re-deployed on another appropriate servers whose energy consumption is smaller than the set threshold. The complex process procedure takes a lot of time. In particular, the total number of VNFs has a great influence on the time complexity of the proposed heuristic deployment algorithm. As illustrated in Figs. 9–11, with the increase of VNFs, the time overhead generated by the proposed heuristic deployment algorithm increases rapidly.

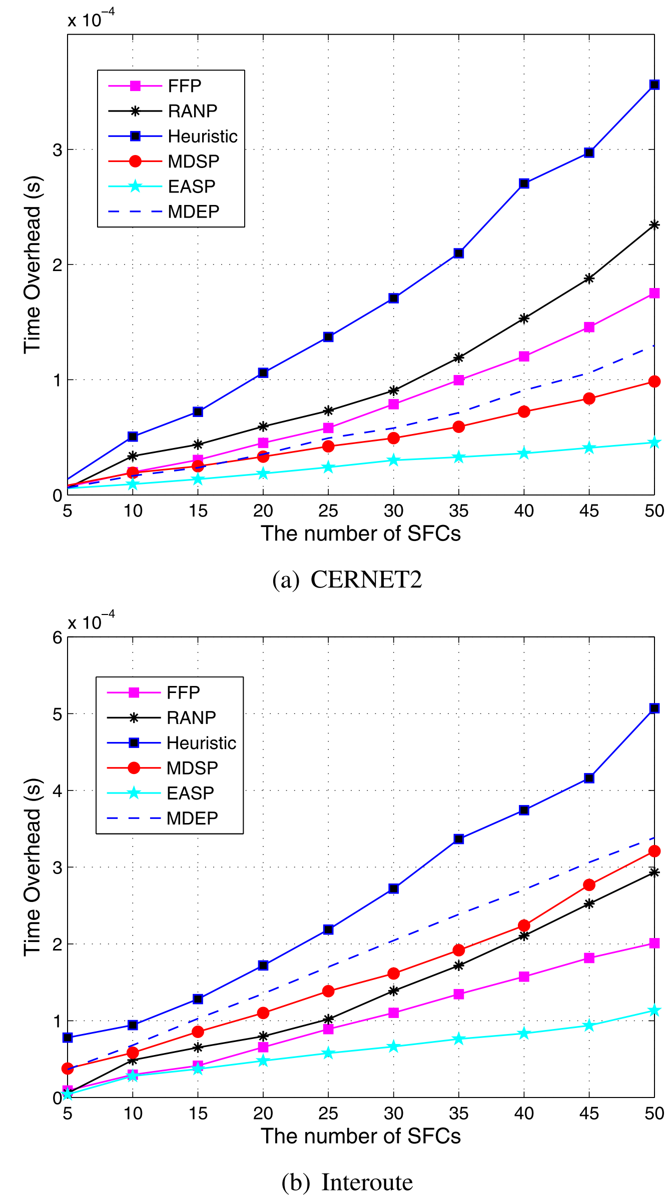
By comparing CERNET2 and Interoute network scenarios, we can observe that the time overhead of each service deployment algorithm in Interoute network scenarios is much higher. The explanations are as follows. Compared with CERNET2 network scenarios, Interoute network scenarios has more server nodes and more network domains, and the resource capacity of its each server is smaller. When we deploy a set of



(a) CERNET2

(b) Interoute

Fig. 9. Time overhead [$D(0, 0.4)$].



(a) CERNET2

(b) Interoute

Fig. 10. Time overhead [$D(0.4, 0.8)$].

VNFs, each service deployment algorithm need traversal more servers and takes more time to search the optimal servers in Interoute network scenarios. In particular, the total number of servers also has a great influence on the time complexity of the proposed heuristic deployment algorithm. As illustrated in Figs. 9–11, compared with CERNET2 network scenarios, the proposed heuristic deployment algorithm generates higher time overhead in Interoute network scenarios.

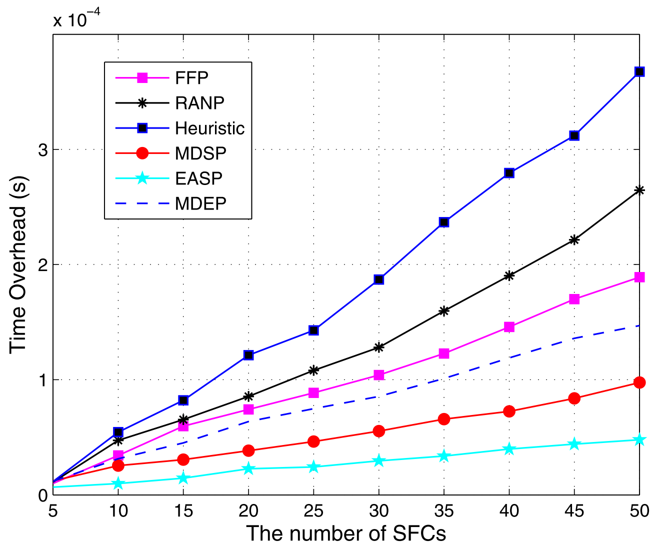
In addition, we can also observe that with the increase of resource demand per VNF, the time overheads of each deployment algorithm gradually becomes big. This is because each server has the limited resource capacity. When the resource demand per VNF increases, the total number of servers which can satisfy the resource demand of VNFs becomes less. In this case, each service deployment algorithm takes more time to search the suitable servers to deploy all the VNFs in the SFC.

In summary, we draw the conclusions from the above performance comparisons that the proposed heuristic deployment algorithm is efficient and outperforms comparison algorithms in terms of energy consumption and load balancing degree.

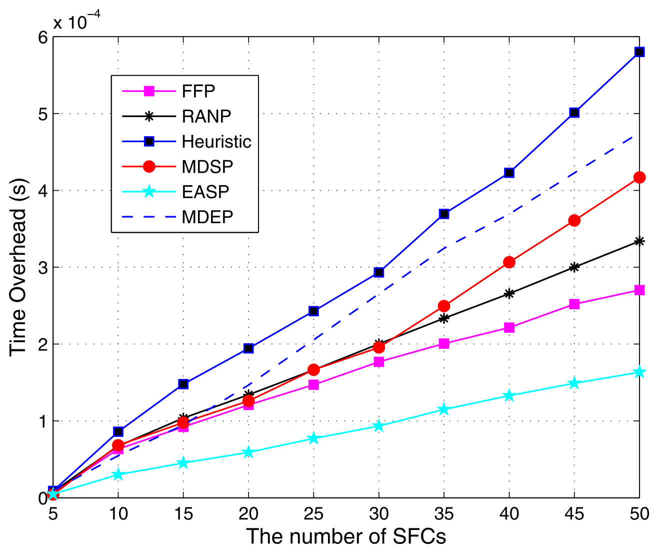
7. Conclusion

In this work, we present an energy efficient service deployment framework to realize flexible SFC deployment in multi-domain networks. A multi-objective optimization model is proposed to minimize energy consumption and load balancing degree. To solve it, we propose a heuristic service deployment algorithm. The extensive simulations demonstrate that the proposed algorithm is efficient and outperforms comparison algorithms in terms of energy consumption and load balancing degree.

There is no a mature and unified simulation platform for SFC deployment in SDNFV environments so far. Our work only verify the proposed heuristic algorithm. In the future, we try to build a real network environment using real machines to verify the proposed scheme. Moreover, in real scenarios, network topologies and user's service demands in different SDN domains usually change with time dynamically. However, this work is limited to static multi-domain SFC deployment. To improve users' QoE, our future work will also explore how to achieve dynamic multi-domain SFC deployment with the aim of



(a) CERNET2



(b) Interoute

Fig. 11. Time overhead [$D(0.8, 1.2)$].

optimizing server and link energy consumption and load balancing by considering additional constraints, such as bandwidth, delay, hop, loss and the agreements between different operators on pricing models.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Chuangchuang Zhang: Conceptualization, Methodology, Software, Writing - original draft. **Xingwei Wang:** Supervision. **Anwei Dong:** Writing - review & editing. **Yong Zhao:** Writing - review & editing. **Qiang He:** Writing - review & editing. **Min Huang:** Writing - review & editing.

Acknowledgments

This work is supported by the Major International (Regional) Joint Research Project of NSFC under Grant No. 71620107003, and the National Natural Science Foundation of China under Grant No. 61872073. We would like to thank all the referees for their invaluable insights and suggestions.

References

[1] X. Cheng, Y. Wu, G. Min, et al., Network function virtualization in dynamic networks: a stochastic perspective, IEEE J. Sel. Areas Commun. 36 (10) (2018) 2218–2232.
[2] C. Bu, X. Wang, H. Cheng, et al., Routing as a service (RaaS): an open framework for customizing routing services, J. Netw. Comput. Appl. 125 (2019) 130–145.
[3] M. Otokura, K. Leibnitz, Y. Koizumi, et al., Evolvable virtual network function placement method: mechanism and performance evaluation, IEEE Trans. Netw. Serv. Manag. 16 (1) (2019) 27–40.
[4] R. Mijumbi, J. Serrat, J.L. Gorricho, et al., Network function virtualization: state-of-the-art and research challenges, IEEE Commun. Surv. Tutor. 18 (1) (2016) 236–262.
[5] C. Zhang, X. Wang, F. Li, M. Huang, NNIRSS: neural network-based intelligent routing scheme for SDN, Neural Comput. Appl. 31 (10) (2019) 6189–6205.
[6] K.T. Foerster, S. Schmid, S. Vissicchio, Survey of consistent software-defined network updates, IEEE Commun. Surv. Tutor. 21 (2) (2018) 1435–1461.
[7] C. Zhang, X. Wang, F. Li, et al., Deep learning-based network application classification for SDN, Trans. Emerg. Telecommun. Technol. 29 (pp) (2018) e3302.
[8] B. Zhang, X. Wang, M Huang, Multi-objective optimization controller placement problem in internet-oriented software defined network, Comput. Commun. 123 (2018) 24–35.
[9] B. Zhang, X. Wang, M Huang, Adaptive consistency strategy of multiple controllers in SDN, IEEE Access 6 (2018) 78640–78649.
[10] Q. Zhang, X. Wang, M. Huang, et al., Software defined networking meets information centric networking: a survey, IEEE Access 6 (2018) 39547–39563.
[11] L. Linguaglossa, S. Lange, S. Pontarelli, et al., Survey of performance acceleration techniques for network function virtualization, Proc. IEEE 107 (4) (2019) 746–764.
[12] R. Bolla, C. Lombardo, R. Bruschi, et al., DROPv2: energy efficiency through network function virtualization, IEEE Netw. 28 (2) (2014) 26–32.
[13] M.F. Tuysuz, Z.K. Ankarali, D. G?zpek, A survey on energy efficiency in software defined networks, Comput. Netw. 113 (2017) 188–204.
[14] M.M. Tajiki, S. Salsano, M. Shojafar, et al., Energy-efficient path allocation heuristic for service function chaining, in: IEEE Conference on Innovation in Clouds, Internet and Networksand Workshops, 2018, pp. 1–8.
[15] B. Kar, E.H.K. Wu, Y.D. Lin, Energy cost optimization in dynamic placement of virtualized network function chains, IEEE Trans. Netw. Serv. Manag. 15 (1) (2018) 372–386.
[16] I. Jang, D. Suh, S. Pack, et al., Joint optimization of service function placement and flow distribution for service function chaining, IEEE J. Sel. Areas Commun. 35 (11) (2017) 2532–2541.
[17] N. Huin, A. Tomassilli, F. Giroire, et al., Jaumard, Energy-efficient service function chain provisioning, J. Opt. Commun. Netw. 10 (3) (2018) 114–124.
[18] K. Yang, H. Zhang, P. Hong, Energy-aware service function placement for service function chaining in data centers, in: IEEE Global Communications Conference, 2016, pp. 1–6.
[19] M.M. Tajiki, S. Salsano, L. Chiaraviglio, et al., Joint energy efficient and QoS-aware path allocation and VNF placement for service function chaining, IEEE Trans. Netw. Serv. Manag. 16 (1) (2018) 374–388.
[20] V. Eramo, M. Ammar, F.G. Lavacca, Migration energy aware reconfigurations of virtual network function instances in NFV architectures, IEEE Access 5 (2017) 4927–4938.
[21] C. Pham, N.H. Tran, S. Ren, et al., Traffic-aware and energy-efficient vnf placement for service chaining: joint sampling and matching approach, IEEE Trans. Serv. Comput. (2017).
[22] A. Marotta, F. D.A.ndreagiovanni, A. Kassler, et al., On the energy cost of robustness for green virtual network function placement in 5G virtualized infrastructures, Comput. Netw. 125 (2017) 64–75.
[23] J. Sun, Y. Chen, M. Dai, et al., Energy efficient deployment of a service function chain for sustainable cloud applications, Sustainability 10 (10) (2018) 3499–3523.
[24] O. Soualah, M. Mechtri, C. Ghribi, et al., Energy efficient algorithm for VNF placement and chaining, in: Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2017, pp. 579-588.
[25] C. Zhang, X. Wang, F. Li, et al., Network service chains deployment across multiple SDN domains, Int. J. Commun. Syst. 31 (18) (2018) e3826.
[26] G. Sun, Y. Li, G. Zhu, et al., Energy-efficient service function chain provisioning in multi-domain networks, in: International Conference on Internet of Things, Big Data and Security, 2018, pp. 144–152.

[27] Q. Xu, D. Gao, H. Zhou, et al., An energy-aware method for multi-domain service function chaining, J. Internet Technol. 19 (6) (2018) 1727–1739.
[28] K. Kaur, S. Garg, G. Kaddoum, et al., An energy-driven network function virtualization for multi-domain software defined networks, in: IEEE INFOCOM 2019 Workshop on Intelligent Cloud Computing and Networking (ICCN 2019), 2019, pp. 1–6.
[29] G. Li, H. Zhou, B. Feng, et al., Context-aware service function chaining and its cost-effective orchestration in multi-domain networks, IEEE Access 6 (2018) 34976–34991.
[30] G. Sun, Y. Li, H. Yu, et al., Energy-efficient and traffic-aware service function chaining orchestration in multi-domain networks, Future Gener. Comput. Syst. 91 (2019) 347–360.
[31] S. Kim, S. Park, Y. Kim, et al., VNF-EQ: dynamic placement of virtual network functions for energy efficiency and QoS guarantee in NFV, Cluster Comput. 20 (3) (2017) 2107–2117.
[32] R. Bruschi, A. Carrega, F. Davoli, A game for energy-aware allocation of virtualized network functions, J. Electr. Comput. Eng. 2016 (7) (2016) 1–10.
[33] T. Wang, H. Xu, F. Liu, Multi-resource load balancing for virtual network functions, in: IEEE International Conference on Distributed Computing Systems, 2017, pp. 1322–1332.
[34] M.T. Thai, Y.D. Lin, P.C. Lin, et al., Towards load-balanced service chaining by hash-based traffic steering on softswitches, J. Netw. Comput. Appl. 109 (2018) 1–10.
[35] X. Fei, F. Liu, H. Xu, et al., Towards load-balanced VNF assignment in geo-distributed NFV Infrastructure, in: IEEE/ACM 25th International Symposium on Quality of Service, 2017, pp. 1–10.
[36] F. Carpio, S. Dhahri, A. Jukan, VNF Placement with replication for load balancing in NFV networks, in: IEEE International Conference on Communications (ICC), 2017, pp. 1–6.
[37] C. Rose, M. Hluchyj, The performance of random and optimal scheduling in a time-multiplex switch, IEEE Trans. Commun. 35 (8) (1987) 813–817.
[38] X. Tang, Y. Li, R. Ren, et al., On first fit bin packing for online cloud server allocation, in: IEEE International on Parallel and Distributed Processing Symposium, 2016, pp. 323–332.
[39] C. Zhang, X. Wang, Y. Zhao, et al., Cost efficient and low-latency network service chain deployment across multiple domains for SDN, IEEE Access 7 (2019) 143454–143470.
[40] S. Sahhaf, W. Tavernier, M. Rost, et al., Network service chaining with optimized network function embedding supporting service decompositions, Comput. Netw. 93 (2015) 492–505.
[41] J. Liu, W. Lu, F. Zhou, et al., On dynamic service function chain deployment and readjustment, IEEE Trans. Netw. Serv. Manag. 14 (3) (2017) 543–553.