An Energy-aware Method for Multi-domain Service Function Chaining

Qi Xu, Deyun Gao, Huachun Zhou, Wei Quan, Wenfeng Shi

School of Electronic and Information Engineering, Beijing Jiaotong University, China
{15111046, gaody, hchzhou, wquan, 14111038}@bjtu.edu.cn

Abstract

As the fast increasing of multi-domain networks, energy efficiency has been widely focused on and become a critical issue in service function chaining (SFC). In this paper, we propose an energy-aware method for multiple SFC-enabled domain networks. Firstly, based on hierarchical SFC (hSFC), we propose a hierarchical control architecture to support the orchestration that allows service chains across multiple domains. Secondly, this paper proposes an energy-aware service function chain placement (EA-SFCP) algorithm that minimizing the power consumption of each service chain. Thirdly, we also propose an energy-aware service function chain migration (EA-SFCM) algorithm to improve the overall energy efficiency of the network. Finally, we implemented the proposed algorithms and conducted comparison simulations with existing algorithm to evaluate their performance. The results show that can reduce power cost at least 33.3% over the candidate algorithm.

Keywords: SFC, Energy-aware, Multi-domain networks, SFC placement, SFC migration

1 Introduction

The ever-increasing services and applications bring huge challenges for efficiently service delivery [1-2]. Service function chaining (SFC) provides a network capability by steering traffic through a chain-ordered set of service functions (SFs) to achieve the flexible network management and service provision [3]. SFC is not a new concept, which is originated from two emerging technologies called software defined networking (SDN) and network function virtualization (NFV). On the one hand, SDN decouples the control plane and the data one, and provides appropriate programming abstractions to control traffics to correctly traverse SFs in order. On the other hand, NFV enables the virtualization of software-implemented SFs which can be more efficiently and dynamically chained. Thanks to the intertwinement between SDN and NFV technologies, it makes the SFC implementation feasible getting rid of the limitation of the traditional network [4].

Energy consumption has currently become a critical issue for the datacenters. In 2009, the datacenters in China consumed over 36 TWh power consumption occupying 1.2% of the total power consumption in China. What's more, the Figure further increased to 70 TWh in 2011 [5]. The worldwide datacenters consumed 268 TWh power consumption in 2012 [6]. According to the study [7], even at the off-peak period, idle servers still consumes about 60% of power, resulting in energy waste. In such environment where SFs are implemented as software on virtual machines (VMs) which running on the physical servers, the locations of service chains have an important impact on the energy consumption in datacenter networks.

This paper provides a n energy-aware method for multiple SFC-enabled domain networks based on hSFC architecture. In this solution, we focus on two points to reduce energy consumption, including service chain placement and service chain migration. Both of them have significant impacts on resource utilization and energy efficiency. In particular, the service chain placement refers to that the SFC-enabled network needs to provision sufficient physical resources (eg. SFs or link bandwidth) when receiving a service request, and to release corresponding physical resources when the request expires. The service chain migration refers to re-orchestrate the service chains that has been placed according to a certain optimization goal. To the best of our knowledge, there are few efforts toward the energy-efficiency problem in multi-domain SFC-enabled networks. The main idea of our proposed approach is to occupy as few physical resources as possible to accommodate as many as service requests and to turn off the lightly loaded network equipment if possible. The main contributions of this paper can be summarized as the following three aspects. Firstly, a hierarchical control architecture is proposed for hierarchical SFC (hSFC) that compartmentalizes a large-scale network into multiple administration domains [8]. Secondly, EA-SFCP algorithm is designed to find an energy efficient placement that minimizing power consumption of each

*Corresponding Author: Qi Xu; E-mail: 15111046@bjtu.edu.cn
DOI: 10.3966/160792642018111906010

service chain. Thirdly, EA-SFCM algorithm is also proposed to reduce the total energy consumption of the entire network. We implement the energy-aware method by Microsoft Visual Studio and conduct comparison simulations with existing algorithm proposed in [9] to evaluate the performance. The results show that our proposed algorithm can reduce power cost at least 33.3% over the compared algorithm.

The rest of this paper is organized as follows. In Section 2, we discuss related works. Then, a hierarchical control architecture is presented in Section 3. In Section 4, we detail EA-SFCP algorithm and EA-SFCM algorithm. The simulation results are presented in Section 5. Section 6 concludes this paper.

2 Related Works

This section presents the background information about service function chaining and service chain placement approaches.

2.1 Service Function Chaining

SFC has been deployed by network operators for many years. Many service functions such as firewall are used by network operators in the delivery of services to end users. The common SFs deployment is to insert a specific dedicated device in the routing path between communicating peers. Therefore, complex modifications are involved into the network configuration. What's more, expensive devices make traditional SFC has low scalability and availability. The rise of SDN and NFV provides flexible traffic steering and efficient SFs deployment. And several research institutions have started to investigate service function chaining from different aspects. In 2013, IETF established the SFC Working Group to define the architecture of SFC, the necessary protocols, several use-cases, and the mechanisms for steering traffic and so on [3]. ETSI NFV group focused on the management and orchestration of virtualized network functions (VNFs), and its VNF forwarding graph concept is very relevant to SFC [10]. Moreover, the SFC project in OpenDaylight foundation aims to develop and apply service chains [11].

2.2 Service Chain Placement Approaches

There are many works with the discussion of the placement problem of SFs and SFC. Ma et al. [12] investigated the traffic changing effects of middleboxes, and proposed an optimal solution to deploy NFV middleboxes efficiently to achieve optimal network performance. Sahhat et al. [13] presented the idea of service decomposition which was similar with the hierarchy of SFC, and proposed two algorithms to embed network services to the shared network infrastructure to minimize the mapping cost by making appropriate decompositions of network functions. Besides, three greedy algorithms are presented in [14] to address the online virtual function mapping and scheduling problem. Hirwe and Kataoka [15] described an approach to address the VNF placement problem, called LightChain. It utilized a heuristic way to optimize the placement of VNFs across service chains in the network in a polynomial run time. However, only a few works discussed about the placement problem of SFC in multi-domain networks. Medhat et al. [9] proposed a near optimal SF selection algorithm in a multi-datacenter environment. Zhang et al. [16] described a distributed computing algorithm for mapping a service function chain request in multi-domain networks. Chien et al. [22] proposed a service-oriented SDN-SFC load balance mechanism. The above work is either to discuss how to optimize the placement of SFs in a single domain or discuss how to implement service chains in multi-domain networks. None of these efforts take energy efficiency into consideration. Distinguishing from them, this paper aims to resolve the energy efficiency problem in multiple SFC-enabled domain networks.

3 Hierarchical Control Architecture

In this section, we introduce a hierarchical control architecture based on the hSFC architecture

3.1 Hierarchical SFC

For implementing service chains that across multi-domain networks, Dolson et al. [8] initiated hierarchical SFC (hSFC) architecture in IETF SFC WG. As shown in Figure 1, hSFC divides a large-scale network into multiple level domains: top-domain is similar to a wide area network and connects multiple sub-domains. The data plane of each domain consists of four types of components, named service functions (SFs), classifiers (CFs), service function forwarders (SFFs) and internal boundary nodes (IBNs), respectively. In NFV, SFs are usually hosted in physical service nodes (SNs) as VMs and responsible for specific treatment of received packets. CFs are used to select which traffic enters an SFC-enabled domain by matching the classification rules. SFFs are responsible for forwarding traffic to one or more connected SFs, as well as SFFs. IBNs are located between top-domain and sub-domains, responsible for bridging packets. IBN has different meanings for different domains. It behaves as a SF to the higher level, but as a CF or SFF to the lower level.

However, the standardization is still in progress and research on the control plane of the service chain is just the beginning. Boucadair [17] described requirements for exchanging information between SFC control plane and SFC data plane components, but it does not consider inter-domain problem. And Dolson et al. [8] discussed that each control plane manages a single

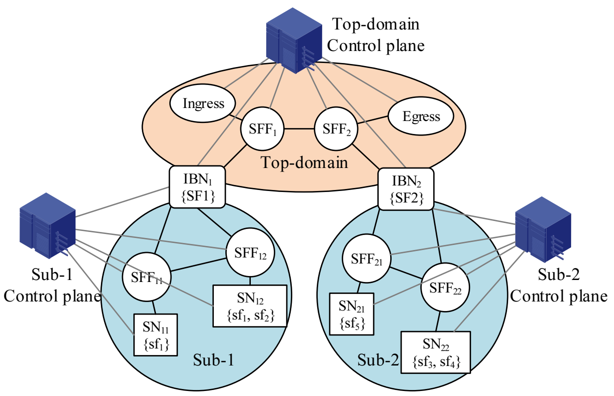


Figure 1. The hierarchical SFC architecture

level domain so that to reduce the complexity of management and orchestration. As far as we know, there is not a specific scheme for SFC control plane, especially for control hierarchy.

3.2 Hierarchical Control Architecture

We present the hierarchical control architecture for hSFC, displayed in Figure 1. For the sake of clarity, it only consists of two levels of hierarchy. If desired, the SFs in sub-domain also could be an independent sub-domain. All components in data plane are defined in IETF SFC architecture RFC [3], with the exception of Service Node (SN) that is able to host SF instances. The ingress and the egress are the gateways of the SFC-enable network, which play a role of the classifier. Moreover, there are four kinds of interface between control plane and data plane, namely C1, C2, C3, and C4 [17]. Due to IBN behaving as a SF to top-domain, it is controlled by interface C3 or C4. Besides, IBN acts as a classifier and a SFF of end-of-chains to sub-domain, it exchanges information with sub-domain control plane via interface C1 and C2. In this architecture, each level of hierarchy of domains is independent, which means each control plane is only responsible for managing SFC of the local domain. Further, the orchestration of control plane in top-domain is at a coarse level. Each SFC in top-domain consists of an order of complex SFs. Complex SF is a logic SF which is actually a SFC composed by more refined SFs within a certain sub-domain. Control plane in top-domain does not need to be responsible for orchestrating these SFCs within the sub-domains. When a complex SF is invoked by top-domain, it looks like that a sub-domain receives a service request from top-domain. Then, sub-domain executes fine-grained SFC orchestration to provide the requested service the complex SF should provide.

As we know, there are no standard protocol for the communication between the control plane and the components on the data plane. In this paper, we focus on the required information to be conveyed between IBN and Top/Sub-domain control plane via interface C3/C1, and design a control protocol for it. In general, the communication mechanism between IBN and control plane is shown in Figure 2. It is important that the control protocol works well for the hierarchical control architecture of hSFC. In the procedure of hierarchical orchestration, once a complex SF is invoked by top-domain control plane, it will be happened that the relevant sub-domain control plane should carry out the SFC instantiation to provide the invoked complex SF for top-domain. Besides, for dynamic, an IBN acts as a SN in top-domain, maintains its resource information which could be the sum of all SNs in a sub-domain and updated by the sub-domain control plane. Thus, the availability of the invoked complex SF must be guaranteed.

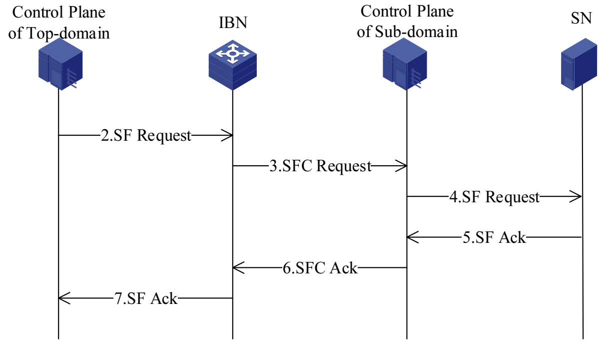


Figure 2. The delivery of signaling messages

As shown in Figure 2, there are four kinds of message type between IBN and control plane. To our knowledge about the required information by hierarchical orchestration, we design their message formats are show in Figure 3. All kinds of message format contain a message header which is constituted by two fields, named by *Request ID* and *Type* respectively. *Request ID* is the identifier of a SFC request from user/IBN or a SF request from control plane. *Type* represents the type of message which is applied to distinguish these messages as follows:

- 0: ACK message
- 1: SFC request message
- 2: SF request message

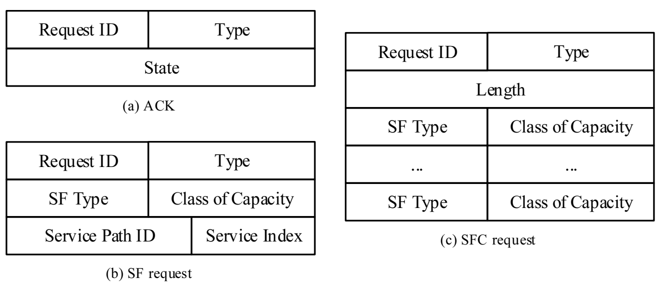


Figure 3. The proposed signaling message format

SFC request message consists of three part: (1) message header; (2) *Length* field that means the number of required SFs in the SFC; (3) *SF Type* fields and *Class of Capacity* fields. The length of the third part is variable, which is determined by the *Length* field. *SF Type* field represents the type of the required SF (e.g. firewall, NAT). And the order of *SF Type* fields once be decided, it is fixed during the procedure of orchestration. Each *SF Type* field follows a *Class of Capacity* field, which stands for the resource requirement of the required SF.

SF request message is a part of SFC request message, actually. It also consists of three part: (1) Message header; (2) *SF Type* field and *Class of Capacity* field; (3) SFP information which includes *Service Path ID* (SPI) field and *Service Index* (SI) field. The SFP information represents the SFC invokes the requested SF.

ACK message contains a message header and a *state* field. The *state* field represents the result of a SF/SFC request, and it has two values: it equals to 1 if the request represented by the *Request ID* field succeeds, otherwise it equals to 0.

4 Energy-aware Service Chain Method

In this section, we describe the problem, and present the power consumption formula. Then we propose two heuristic algorithms, EA-SFCP and EA-SFCM, to improve the energy efficiency of network together. Finally, we detail the SFC placement in multi-domain networks.

4.1 Problem Description

As detailed in Section 3, each domain in the hSFC architecture is independent with other ones. Consequently, the multi-domain energy efficiency problem can be simplified into several independent single-domain sub-problems. This article tries to improve energy efficiency at two aspects: energy-aware SFC placement and energy-aware SFC migration. The former one refers to that given a substrate network G with an amount of physical resource, finding a placement scheme for a given service request G_v that makes the additional power consumption minimum after deploying it. The latter one refers to migrating the workload on the light-load SN nodes to other SN nodes.

4.2 Power Consumption Model

We model the substrate network as an undirected weighted graph $G = (N, L)$, where N and L denote the set of physical nodes and physical links, respectively. The physical nodes are divided into three categories according to their own role, denoted as N_{SN}, N_{SFF}, N_{CF}, respectively. That is $N = (N_{SN}, N_{SFF}, N_{CF})$. An assumption is made that one SFF only attaches one SN node. Besides, we also assume the amount of CPU resource as the constraint of SN node, and the bandwidth as the resource constraint of link. Each SN $n_j \in N_{SN}$ has residual CPU capacity described by C_j. Each physical link $l_{j,k} \in L$ represents the connectivity between node j and node k, and its residual bandwidth capacity denoted as $B_{j,k}$. Similar to the substrate network, a service request is represented as a directed weighted graph $G_v = (N_v, L_v)$, where N_v and L_v denote the set of required SFs and related virtual links, respectively. Here, $N_v = (sf_1, ..., sf_i)$, where sf_i represents the ith SF in a given service request G_v. The resource requirements associated with relevant SFs and virtual links required by G_v, are denoted as c_i and $b_{i-1,i}$, respectively.

The following is the formulas for calculating the energy consumption of each service chain in this article.

Power cost of a service chain. The power cost of a service chain G_v is composed by the power cost of SNs and links, calculated as follows:

$$\Delta P(G_v) = \Delta P_{SNs} + \Delta P_{Links} \tag{1}$$

Service node power cost. The additional SN power cost for placing a requested SFC is calculated by:

$$\Delta P_{SNs} = \sum_{i \in N_v} \sum_{j \in N_{SN}} x_j^i \Delta PN_j^i \tag{2}$$

where ΔPN_j^i denotes the additional power cost for placing a SF $sf_i \in N_v$ on a SN $n_j \in N_{SN}$, x_j^i is a binary variable indicates if the SF is successfully placed.

Fan et al. [18] had studied the power model at the machine level. They found the workload in CPUs made a major contribution to the power consumption of a server and the relationship between them was consistent with an empirical non-linear model. Thus, the additional power cost for placing a SF can be calculated as follows:

$$\Delta PN_j^i = \theta_j \cdot P_{idle} + (P_{busy} - P_{idle}) \cdot (2u - u^r) \tag{3}$$

where P_{idle} represents the power cost of an idle server, P_{busy} is the power cost of a busy one, u represents CPU utilization, and r is a parameter used to minimize the square error [19]. θ is a binary variable. If θ=1, service node SN_j is inactive.

Link power cost. The total link power cost for placing a requested SFC can be calculated as follows:

$$\Delta P_{Links} = \sum_{l_{i-1,i} \in L_v} \sum_{l_{j,k} \in L} y_{j,k}^{i-1,i} \Delta PN_{j,k}^{i-1,i} \tag{4}$$

Where $\Delta PL_{j,k}^{i-1,i}$ denotes the additional power consumption for mapping a virtual link $l_{i-1,i} \in L_v$ on a physical link $l_{j,k} \in L$, $y_{j,k}^{i-1,i}$ is a binary variable indicates if the virtual link is successfully mapped.

In [20], the power cost of links is composed by a static contribution due to the line cards and by an additional term due to the repeaters. The additional power cost for mapping a virtual link is given by:

$$\Delta PL_{j,k}^{i-1,i} = (\frac{Dis(j,k)}{\rho} \cdot P_r + P_{lc}) \cdot \frac{b_{i-1,i}}{BW_{j,k}} \quad \textbf{(5)}$$

where P_r denotes the power consumption of a repeater, P_{lc} is the power consumption of a line card, ρ represents the distribution density of repeaters on the link, and $BW_{j,k}$ means the total bandwidth capacity of the physical link connects node j with node k.

4.3 Energy-aware SFC Placement Algorithm

In the real network, service requests arrive and expire over time. They are usually temporary and dynamic, and can only exist for some time. The EA-SFCP algorithm aims to resolve service chain placement problem in the real network. It is able to produce an energy-efficient placement scheme for each service request so that reducing the overall energy consumption of the network. The main idea of our proposed algorithm is to occupy as few physical resources as possible to accommodate as many SFCs as possible and to turn off the idle equipment of the network. Due to the dynamic nature of service requests, we defined three queues, namely Q, Q_{suc} and Q_{run} for purposes. Q is the queue of arrival service requests. The EA-SFC algorithm computes SFC placement scheme for service requests in the queue Q. Q_{suc} that is a subset of Q, denotes the queue of those service requests that succeeded to place on the network. At last, Q_{run} is a subset of subset Q_{suc}, and denotes the queue of those service requests that have been placed in the network and have not yet expired. During the procedure of SFC placement, we do not only consider SF resource constraint, but also satisfy link resource constraint. The pseudo-code of the energy-aware SFC placement algorithm is given as follows:

Algorithm 1. EA-SFCP Algorithm

Input: G, Q, η

Output: Energy-efficient SFC placement solution *sfp*

```
1  For G_v ∈ Q do
2    For sf_i ∈ N_v do
3      Find the set of all available SNs n_i ∈ N_SN.
4      Sort n_j according to u_j in nondecreasing order.
5      For n_i ∈ N_SN do
6        Compute ΔPN_j^i based on equation(1)
7        For l_{j,k} ∈ L do
8          If B_{j,k} ≤ (1−η)·BW_{j,k} then
9            Set weight as INF.
10         Else
11           Set weight as ΔPL_{j,k}^{i−1,i} based on equation (5)
12         Compute the shortest weighted path L_short.
13       Goto Step 5
14       Select n_j with lowest (ΔPN_j^i + ΔPL_{j,k}^{i−1,i}).
15       Return SFC_Placement_Faild
16       Push G_v into Q_suc and Q_run.
17       Update ΔP , sfp, and  residual network resources.
18       If G_v expired then
19         Release occupied resources.
20         Push G_v out from Q_run.
21     Goto Step 1
22 Return SFC_Placement_Success
```

When a service request arrives, its power cost ΔP can be calculated based on equation (1). EA-SFCP traverses all available SN nodes for placing a series of wanted SFs in the service request. Meanwhile, we use the *Dijkstra* algorithm to calculate the shortest weighted path for the virtual link between adjacent SFs. The weight of each physical link is different and represents its power consumption, calculated by equation (5). Obviously, the shortest path must connect the current SN node with the SN node hosts the previous SF. The sum of power cost by the SN node and power cost by the shortest path will be calculated. When all the SN nodes have been traversed, the SN node and path with the lowest additional power cost will be selected to place the requested SF and its neighbor link. Above process will be repeated until all SFs in the service request are placed. Note that, for the first SF in a SFC, the shortest weighted path is calculated between the ingress node and the current selected node. For the last SF in a SFC, the shortest path is calculated between the service node that hosts previous SF and the current selected node in addition to the path between the current selected node and the egress node. When the service request expires, the corresponding occupied network resources should be released and the service request is pushed out from the queue Q_{run}. Furthermore, we define a parameter for each physical link called load balance factor η, which takes the value between 0 and 1. It determines the upper limit of the residual bandwidth resources $B_{j,k}$ for physical link $l_{j,k}$ to reserve a certain amount of redundant bandwidth, so as not to exhaust total bandwidth and occur traffic congestion. When the residual bandwidth of a link is less than the redundant bandwidth, the weight of the link is set to infinity so that it is ignored when calculating the shortest weighted path. Thus, it provides a tradeoff between traffic load and energy efficiency.

Algorithm 2. EA-SFCM Algorithm

Input: G, Q_{run}

Output: new placement solution *sfp*

```
1 Find the set of all available SN nodes n_j ∈ N_SN.
2   Sort n_j according to the value of stress in
      nondecreasing order.
3   For n_j ∈ N_SN do
4     Find the already placed SF instances sf_i ∈ N_SF
```

```
5     Sort n_i according to resource capacity
        in decreasing order.
6      For sf_i ∈ N_SF do
7        Update the physical resource record.
8        Refind the suitable SN node n_best and links L_best
           with lowest ΔP based on Algorithm 1.
9        If Refind succeed
10         Migrate sf_i from n_j to n_best and its relevant links
           Update sfp and the physical resource record.
11         Else
12           Recover the physical resource record.
13           Goto Step 6
14       End for
15    Return SFC_Migration_Success
16  End for
```

4.4 Energy-aware SFC Migration Algorithm

The main idea of EA-SFCP algorithm is to minimize the additional power consumption after mapping one service chain, thereby reducing the total power consumption of the entire network. However, there is a detail problem that is not taken into account. With the arrival of new service requests and the expiration of the old ones, the distribution of residual resource of substrate network is constantly changing. The placements of some service requests may produce the lowest power consumption at the beginning by using EA-SFCP algorithm. But after a period of time, the distribution of the residual resources has changed so that there may be much greener placements making the current ways is not so energy-efficient. Therefore, a desirable approach would be to remapping the SFCs that has already been mapped at regular intervals to gain more energy saving. In this section, we propose a heuristic algorithm to remapping the current already mapped service chains, called Energy-aware SFC Migration algorithm. EA-SFCM algorithm tries to migrate virtual resource requirements from one place to another much more appropriate place in the substrate network by the way of remapping. The optimization goal of EA-SFCM algorithm is to reduce the total power consumption of the entire network while keeping current SFCs has been placed to work properly. In this paper, we refer to the definition of stress in [21], which means the number of virtual instances mapped on it.

The pseudo-code of EA-SFCM algorithm is detailed in *Algorithm2*. Once the EA-SFCM algorithm begins to execute, it will traverse all of the SN nodes in an increasing order according to their stress values. Specially, for each SN node, the group of SFs that have been placed on top of it are arranged in descending order according to the value of their computing resource requirements. This ensures that the service requests that occupy more virtual resources are processed first. EA-SFCM algorithm will them perform remapping operations on these SF requests and associated virtual links request in turn. The remapping operations are based on Algorithm 1. It will recalculate a new placement scheme for the virtual resource request. Then it compares the power consumption between the new placement and the old placement. If the new placement consumes less power, it migrates the SF request from the current SN node to the new SN node. If not, keep the current placement unchanged.

Figure 4 shows a simple example of service migration. As shown in Figure 4a, there are three SFCs that have been placed in the substrate network. After executing EA-SFCM algorithm, the two SFs in SFC 3 are remapped as shown in Figure 4a, and migrated to the new SN nodes as shown in Figure 4b. In such case, the idle SN nodes also can be shut down to save energy.

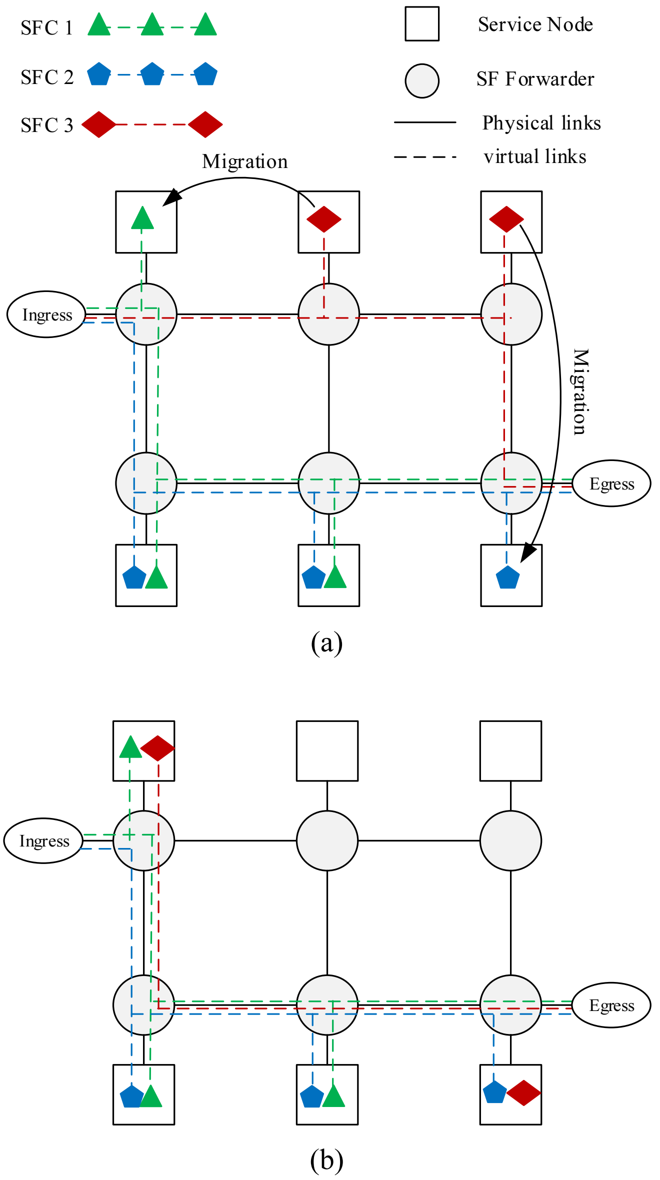


Figure 4. Migration of an already placed SFC

4.5 Multi-domain SFC Placement

As mentioned above, the orchestration, provisioning and migration of SFCs between different domains are independent in the hSFC architecture. Taking advantage of this feature, the problem of SFC placement in multi-domain can be simplifiedinto several independent sub-problems in a single domain.

A clear example of SFC placement in multi-domain is given in Figure 5. In general, service requests in queue Q enter the SFC-enabled network through the ingress gateway. First, the top-domain is responsible for achieving coarse-grained placement. The EA-SFCP algorithm is executed to allocate the required network resources for received service request. Therefore, after being orchestrated, IBN_1 and IBN_2 are responsible for providing SF1 and SF2, respectively. Note that SF1 and SF2 are logical service function, which actually are SFC in their own sub-domain. Second, the two IBNs split the service request received by the top-domain into two sub service requests, and deliver to the corresponding sub-domains. Third, sub-domains are responsible for achieving fine-grained placement for received sub service requests. The two sub-domains also execute the EA-SFCP algorithm on the requests. In this case, a full SFC placement in multi-domain networks is successfully completed, and the service request is pushed into the queue Q_{suc} and Q_{run}. Traffic flow will be routed along the programed path and service functions. When the service request expires, each network domain needs to release the corresponding occupied network resources and the service request is pushed out from the queue Q_{run}. Based on the SFC placement by performing EA-SFCP algorithm, EA-SFCM algorithm is executed periodically to re-orchestrate already placed SFCs in the queue Q_{run} for improving energy efficiency. For simplicity, in the following, we call it Green SFC (G-SFC) algorithm that integrates EA-SFCP algorithm and EA-SFCM algorithm.

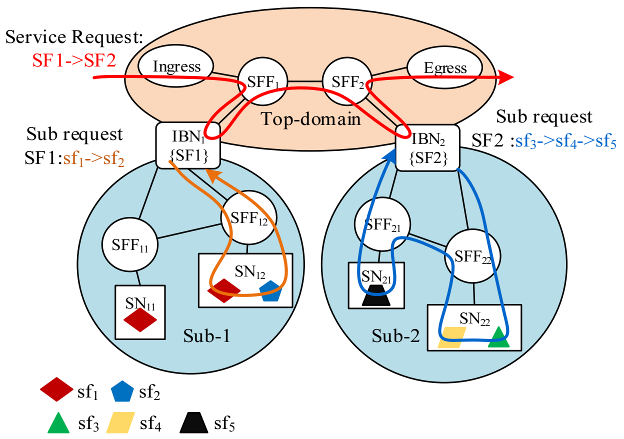


Figure 5. An example of SFC placement in multi-domain networks

5 Evaluations

In this section, we validate and evaluate our proposed architecture and algorithms by conducting simulations in realistic network, and perform comparisons with existing algorithm.

5.1 Simulation Setup

The topology of the infrastructure network in our simulation consists of one top-domain and 8 sub-domains, as show in Figure 6. The network topology of top-domain uses the topology of a real network, called the China Education and Research Network (CERNET), which includes 2 gateways as classifier and 8 SFFs from SFF1 to SFF8. Each SFF in top-domain is connected with an independent sub-domain which includes one IBN, 4 SFFs and 4 SNs. The number marked on the link indicates the length of the link in kilometers. As mentioned above, each sub-domain acts as a SN to top-domain and each SFC in sub-domain acts as a complex SF to top-domain. The traffic flow starts at the ingress, and leaves the network from the egress after passing through several SFs.

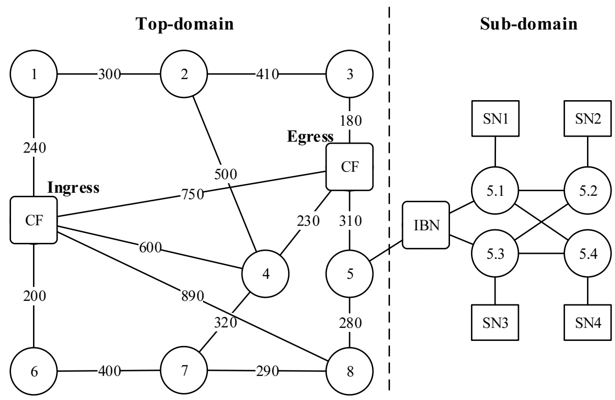


Figure 6. The infrastructure network topology

With regard to the physical resource capacity of the infrastructure network, we made the following assumptions. We assume the CPU capacity as the computing resource and the bandwidth as the link resource. For the top-domain, the total capacity of CPU of each sub-domain is equal to 400 units. The bandwidth capacity of each link is equal to 400G in the top-domain. The propagation delay is up to the total distance of links through the traffic flow passes. For example, the shortest distance between SFF1 and SFF3 is 710km. For each sub-domain, the CPU capacity of each SN is equal to 100 units. And we assume that each link of the sub-domains has enough bandwidth for mapping a service request and the propagation delay in sub-domain can be neglected. We predefined 4 types of SFCs for every sub-domain to provide 4 types of complex SFs for the top-domain. The value of parameters used in our power consumption model is summarized in Table 1.

Table 1. Parameter used in the model

Parameter	Description	Value
P_{idle}	The power cost of an idle server.	165W
P_{busy}	The power cost of an busy server.	166.5W
P_r	The power cost of a repeater	1kW
P_{lc}	The power cost of a line card	100W
ρ	The distribution density of repeaters on the link	70km/per

Besides, in our simulation, 5,000 service requests will be generated randomly, and consist of a given number of SFs. We assume that the CPU resource required by each SF is randomly among [0, 20] memory units, and the bandwidth resource required by each service request is randomly among [0, 20] Gbps. Similar to most prior studies, the arrival and expiration of the service request follow the Poisson process with parameters α and β, respectively. The α determines the time interval of service request arrival, and the β determines the time interval of service request expiration.

5.2 Compared Algorithm

We implemented our algorithm by Microsoft Visual Studio and performed comparisons to the algorithms that are summarized in Table 2. The work in [9] proposed a near optimal SF selection algorithm in a multi-datacenter environment, referred to herein as NeO-SFC algorithm. NeO-SFC algorithm can make a flexible tradeoff between the end-to-end delay performance and the loads of SFs by adjusting a parameter during the service chain placement phase. EA-SFCP algorithm is our proposed solution without the consideration of service migration. And the G-SFC algorithm integrates the migration algorithm on the basis of the EA-SFCP algorithm. These three algorithms will compute the corresponding placement for each service request. Furthermore, the G-SFC algorithm also periodically performs EA-SFCM Algorithm on the basis of EA-SFCP algorithm calculation. In our simulation, every 100 service requests, we let the G-SFC algorithm perform service migration once.

Table 2. Compared algorithms

Algorithm	Description
EA-SFCP	The algorithm 1 that proposed in this work
G-SFC	The energy-aware method that integrated algorithm 1 and 2.
NeO-SFC	Approach proposed in [9].

The performance metrics used in our simulation are summarized below. Firstly, we use average propagation delay to evaluate the hierarchical control architecture. Secondly, referring to [19], we use Blocking Ratio (BR), Real-time Power Consumption (RtPC) and Average Power Consumption (APC) as evaluation metrics. Finally, we analyze the effect of service migration by the statistical data of occupied CPU capacity and saved power.

5.3 Average Propagation Evaluation

As we know, there is certain delay in the process of SFC placement, especially in multi-domain. It is usually composed of calculate delay and propagation delay. The calculation delay is generated by the process of calculating SFC placement scheme for service requests by the controller. The propagation delay is the delay that signaling messages are routed between different domains. Generally, the long distance between different domains in multi-domain networks makes the propagation delay much larger than the calculation delay. Therefore, we evaluate the propagation delay in hSFC architecture using EA-SFCP algorithm to compute placement schemes for 5000 service requests. In this simulation, the controller of top-domain is placed at the ingress node and the controller of each sub-domain is placed as the IBN node.

Here, we average the propagation delay for every 100 service requests. As shown in Figure 7, the average propagation delay is gradually stabilized as more requests arrives. This shows that the proposed hierarchical control architecture is stable. In addition, the average propagation delay increases as the number of SFs increases. Due to one sub-domain only maps one complex SF and each SN node of sub-domain only maps one SF, adding a SF means the number of sub-domains that the control plane of top-domain needs to request will increase. Besides, the average propagation delay is relevant to the placement of controllers. Different locations of controllers, the average distance between controller and the components of data plane associated with the controllers are different.

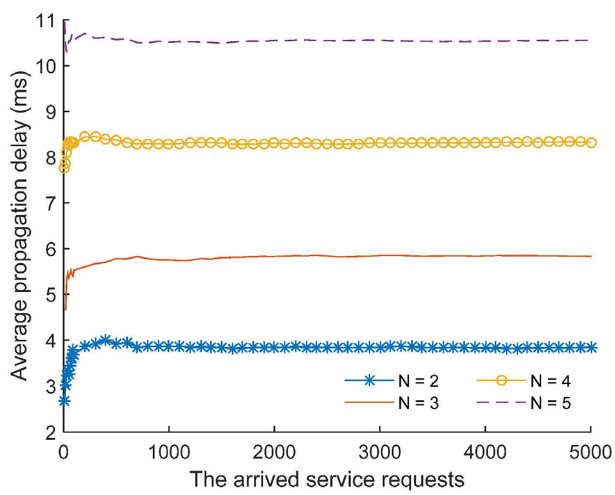


Figure 7. Propagation delay of signaling messages

5.4 Blocking Ratio Evaluation

Due to the limited network resource, only a certain number of service requests can be satisfied at the same time. The blocking rate is an important indicator that represents the efficiency of the SFC placement algorithms, calculated as follows:

$$BR = (1 - \frac{|Q_{suc}|}{|Q|}) \times 100\% \quad \textbf{(6)}$$

where $|Q|$ and $|Q_{suc}|$ respectively indicate the number of service requests in its own queue. In our simulations, these two parameters, α and β, are very important. On the one hand, if the value of α is too large, a large number of service requests cause the length of the queue Q to grow rapidly in a short time. However, the network cannot provide sufficient resources to so many service requests at the same time, resulting in a decrease in the number of service requests in the queue Q_{suc}. The consequence of this case is an increase of BR. On the other hand, when the value of α is too small, it cannot make full use of network resources, resulting in waste of resources. And the effect of expiration frequency β on BR is similar to that of arrival frequency. Therefore, it is very important for SFC placement algorithm that satisfying as many service requests as possible under the network resource constraints. Moreover, for the next comparative simulations, it is very critical to determine appropriately the values of these two parameters. We tried many different values of α and β, carried out a lot of tests and finally selected some meaningful values to compare. Here we give the results of three groups of simulations (Set1: α=50, β=2; Set2: α=60, β=2; Set2: α=60, β=1.4), shown in Figure 8, Figure 9 and Figure 10, respectively.

In each group of contrast simulation, we also make load balance factor η as a variable to observe if the change of η has an effect on the BR. The Simulation results prove that the BR of these three algorithm is deeply affected by parameter α, β, and η. From the three simulation results, it can be clearly known that the BR is negatively correlated with η. Furthermore, let us look at the effect of the other parameters on the BR. Comparing the results of Figure 8 and Figure 9, it is concluded that the BR is positively correlated with the arrival frequency α. And comparing the results of Figure 9 and Figure 10, it is concluded that the BR is also positively correlated with the expiration frequency β. These results verifies our previous analysis. Then we analyze the difference between these three algorithms in BR. Firstly, regardless of whether the value of η is 0.6 or 0.8, the BR of NeO-SFC is significantly higher than the others. Secondly, when the network is idle, such as the case of Figure 8, the BR of G-SFC is lower than that of EA-SFCP. However, when it is busy, such as in Figure 10, EA-SFCP is better than G-SFC. Figure 9 shows exactly what the BR performance of the two algorithms is similar, and their trend curves cross each other. In general, in terms of blocking ratio, both of EA-SFCP and G-SFC are superior than NeO-SFC algorithm. Hereinafter, in order to compare the power consumption, we make α equal to 60, and β to 2.

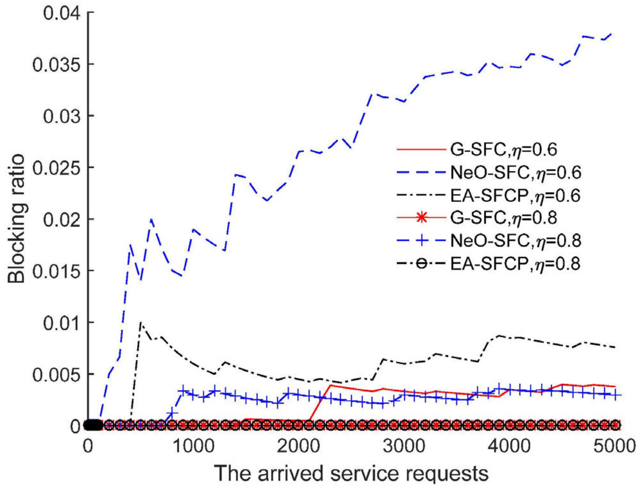


Figure 8. Blocking ratio evaluation (α =50, β =2)

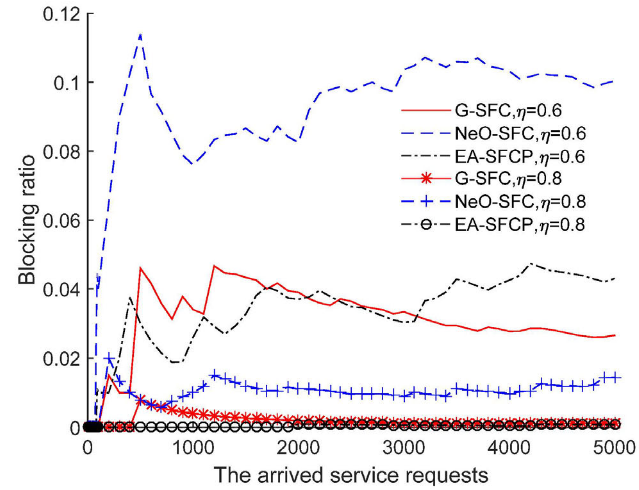


Figure 9. Blocking ratio evaluation (α =60, β =2)

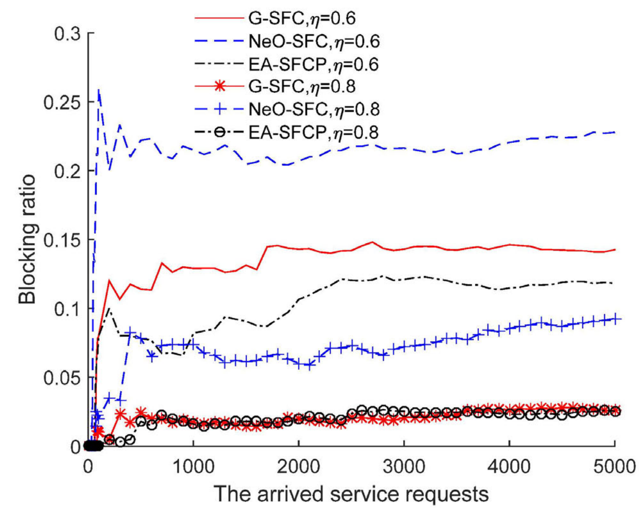


Figure 10. Blocking ratio evaluation (α =60, β =1.4)

5.5 Power Consumption Evaluation

In this section, we study and analyze the energy efficiency of three different algorithms with different load balance factor η. Here, we refer to the definition of APC and RtPC in [19]. The RtPC means the sum of the power consumption of the requested services in the queue Q_{run}. The APC is the average of the power consumption of in the queue Q_{suc}. Their calculations are as follows:

$$APC = \frac{\sum_{G_v \in Q_{suc}} \Delta P(G_v)}{|Q_{suc}|} \quad \mathbf{(7)}$$

$$RtPC = \sum_{G_v \in Q_{run}} \Delta P(G_v) \quad \mathbf{(8)}$$

where $\Delta P(G_v)$ is the power consumption that produced by the requested service G_v, which is calculated through equation (1).

Figure 11 shows the RtPC results of the three approaches, and Figure 12 presents the results of APC. The simulation results show that the RtPC and APC of NeO-SFC algorithm are much larger than the results of EA-SFC and G-SFC algorithm. Also, the EA-SFC and G-SFC algorithm has almost the same result of APC. However, the RtPC of G-SFC algorithm is slightly lower than the result of EA-SFC. In addition, the power consumption of these three algorithms is negatively correlated with the load balancing factor. In Figure 12, we can clearly find that the APC of each algorithm decreases as the load balance factor η increases. Smaller the load balance factor is, the less available bandwidth resource each link has. Thus, with the reduction of load balance factor, the network has to produce more power consumption to satisfy same number of service requests. Besides, the G-SFC produce lower RtPC than EA-SFC in Figure 11, Although the APC results of EA-SFC and G-SFC are almost same in Figure 12. Furthermore, compared to the APC of NeO-SFC, the results of EA-SFCP and G-SFC are far lower than it. Specifically, when η equals to 1, the APC of NeO-SFC is stable at around 600W, and the APC of other two approaches are about 400. As a result, it saves about 33.3% of the average power consumption. However, when η reduces to 0.8, the APC of NeO-SFC reduces to about 550W and the APC of other algorithms also reduce to about 355W. As a result, it saves the APC nearly by 35.5%. Therefore, load balancing factor also has an impact on the energy saving capacity of our proposed approaches: the smaller η, the stronger the energy saving capacity. Similar conclusions can be drawn from the results analysis of RtPC.

5.6 Service Migration Analysis

As servic requests arrive and expire over time, the resource sage of each domain is in a state of constant dynamic change. Using different SFC placement algorithm can also result in changes in resource usage for each domain. For analyzing the variation of different algorithms, we sampled the occupied CPU capacity of each subdomain randomly during the simulation as shown in Figure 13. It is obviously proved that the resource usages of each subdomain by using three algorithms are different. By using NeO-SFC algorithm, the resource usages are evenly distributed among the subdomains of which the occupied computing capacity fluctuates from 60 to 90. On the contrary, it is not evenly by using EA-SFCP and G-SFC algorithm. With EA-SFCP algorithm, a number of computing resource of sub-domain 1/3/4 are occupied. Only a few of computing resource of the other subdomains, especially subdomain 5 and 8, are occupied, which means the SN nodes in those subdomains are in a light-load state. Furthermore, with G-SFC algorithm, we can see the subdomain 5 and 8 are in a zero load state. Comparing the results of subdomain 5 and 8, we can see that the virtual instances are migrated from lightly-loaded subdomain to the other appropriate ones. Thus, this proves that G-

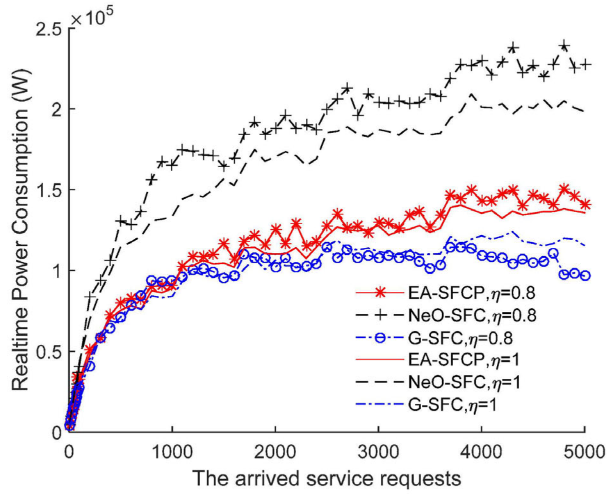


Figure 11. Real-time power consumption evaluation

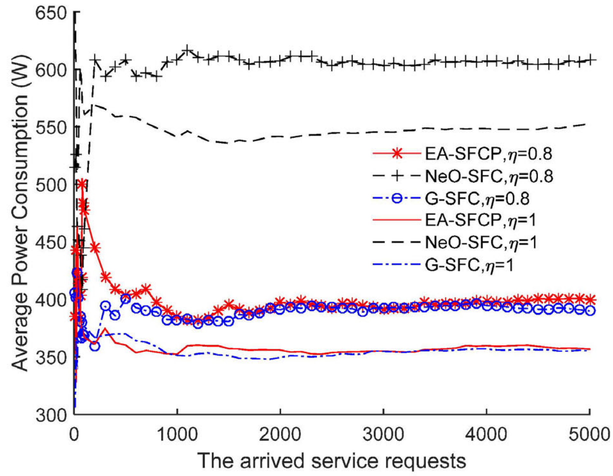


Figure 12. Average power consumption evaluation

SFC algorithm can consolidate network resource by service migration.

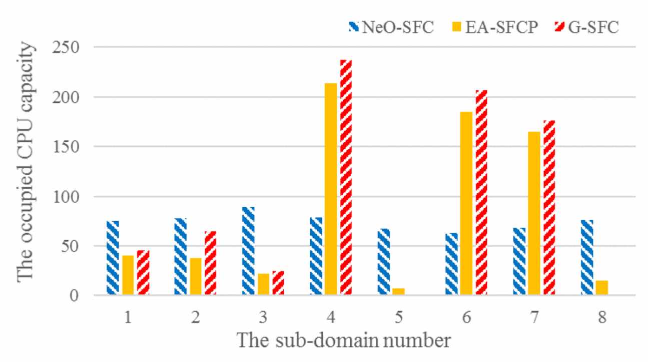


Figure 13. The occupied CPU capacity of each domain

The following is an analysis of the energy efficiency of our proposed service migration algorithm. Figure 14 shows the accumulated trend of power saving by performing service migration. During the simulation process, the service migration is performed every 100 service requests, and 50 times totally. Every time the service migration is performed, the total power consumption of entire network will be reduced. Moreover, we also studied the effects of load balance factor η on the energy efficiency of service migration. When η equals to 0.6, every time resource consolidation can reduce power consumption about 1721.514W. When the factor equals to 0.8, this number drops to 994.922W. In general, the power consumption can be reduced through service migration. Besides, the effects of service migration are different for different time, which depends on the current distribution of residual resource. It is also affected by load balance factor which determines the redundant link resources.

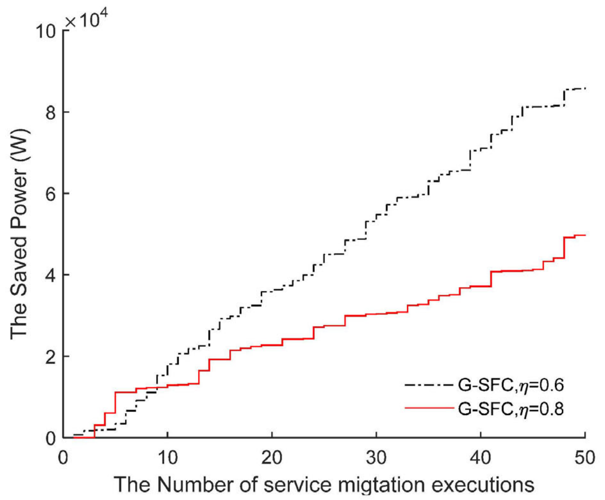


Figure 14. The saved power by service migration

6 Conclusion

In this paper, we studied the energy efficiency problem in multi-domain service function chaining. Specifically, we designed a hierarchical control architecture based on the hSFC architecture to support service chains across multi-domain networks. We also proposed an energy-aware SFC placement algorithm and an energy-aware SFC migration algorithm to improve energy efficiency of network together. Simulation results show that EA-SFCP algorithm can provide an energy-efficient placement for each service chain. Based on EA-SFCP, G-SFC, which integrates EA-SFCM, can consolidate network resource and further reduce the energy consumption of the network. Compared to existing algorithms, our proposed method not only have a lower blocking ratio, but also reduce at least 33.3% power consumption for each service request. In the future work, we will further study the implementation of hierarchical control plane, and verify our proposed control protocol.

Acknowledgments

This work was supported by NSFC under Grant No. 61232017, NSAF under Grant No. U1530118, National High Technology of China (“863 program”) under Grant No. 2015AA015702, and National Basic Research Program of China (“973 program”) under Grant No. 2013CB32910.

References

[1] F. A. Moghaddam, P. Lago, P. Grosso, Energy-efficient Networking Solutions in Cloud-based Environments: A Systematic Literature Review, *ACM Computing Surveys (CSUR)*, Vol. 47, No. 4, Article No. 64, July, 2015.

[2] T. Kaur, I. Chana, Energy Efficiency Techniques in Cloud Computing: A Survey and Taxonomy, *ACM Computing Surveys (CSUR)*, Vol. 48, No. 2, Article No. 22, November, 2015.

[3] J. Halpern, C. Pignataro, *Service Function Chaining (SFC) Architecture*, IETF RFC 7665, October, 2015.

[4] W. John, K. Pentikousis, G. Agapiou, E. Jacob, M. Kind, A. Manzalini, F. Risso, D. Staessens, R. Steinert, C. Meirosu, Research Directions in Network Service Chaining, *IEEE SDN for Future Networks and Services (SDN4FNS)*, Trento, Italy, 2013, pp. 1-7.

[5] China Electronics Standardization Institute, *Research Report on Energy Efficiency of China Datacenters*, June, 2015.

[6] W. Van Heddeghem, S. Lambert, B. Lannoo, D. Colle, M. Pickavet, P. Demeester, Trends in Worldwide ICT Electricity Consumption from 2007 to 2012, *Computer Communications*, Vol. 50, pp. 64-76, September, 2014.

[7] D. Meisner, B. T. Gold, T. F. Wenisch, PowerNap: Eliminating Server Idle Power, *ACM Sigplan Notices*, Vol. 44, No. 3, pp. 205-216, March, 2009.

[8] D. Dolson, S. Homma, D. Lopez, M. Boucadair, D. Liu, T. Ao, V. Vu, *Hierarchical Service Function Chaining*, IETF draft-ietf-sfc-hierarchical-01, September, 2016.

[9] A. M. Medhat, G. Carella, C. Lück, M.-I. Corici, T. Magedanz, Near Optimal Service Function Path Instantiation in a Multi-datacenter Environment, *11th International Conference on Network and Service Management (CNSM)*, Barcelona, Spain, 2015, pp. 336-341.

[10] ETSI, *Network Functions Virtualization*, http://www.etsi.org/technologies-clusters/technologies/nfv

[11] OpenDaylight, *Service Function Chaining: Main*, Available: https://wiki.opendaylight.org/view/Service_Function_Chaining:Main

[12] W. Ma, C. Medina, D. Pan, Traffic-Aware Placement of NFV Middleboxes, *IEEE Global Communications Conference (GLOBECOM)*, San Diego, CA, 2015, pp. 1-6.

[13] S. Sahhaf, W. Tavernier, M. Rost, S. Schmid, D. Colle, M. Pickavet, P. Demeester, Network Service Chaining with Optimized Network Function Embedding Supporting Service Decompositions, *Computer Networks*, Vol. 93, pp. 492-505, December, 2015.

[14] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. D. Turck, S. Davy, Design and Evaluation of Algorithms for Mapping and Scheduling of Virtual Network Functions, *1st IEEE Conference on Network Softwarization (NetSoft)*, London, UK, 2015, pp. 1-9.

[15] A. Hirwe, K. Kataoka, LightChain: A Lightweight Optimisation of VNF Placement for Service Chaining in NFV, *IEEE NetSoft Conference and Workshops (NetSoft)*, Seoul, South, Korea, 2016, pp. 33-37.

[16] Q. Zhang, X. Wang, I. Kim, P. Palacharla, T. Ikeuchi, Vertex-centric Computation of Service Function Chains in Multi-domain Networks, *IEEE NetSoft Conference and Workshops (NetSoft)*, Seoul, South Korea, 2016, pp. 211-218.

[17] M. Boucadair, *Service Function Chaining (SFC) Control Plane Components & Requirements*, IETF draft-ietf-sfc-control-plane-07, August, 2016.

[18] X. Fan, W. D. Weber, L. A. Barroso, Power Provisioning for a Warehouse-sized Computer, *ACM SIGARCH Computer Architecture News*, Vol. 35, No. 2, pp. 13-23, May, 2007.

[19] G. Sun, V. Anand, D. Liao, C. Lu, X. Zhang, N.-H. Bao, Power-efficient Provisioning for Online Virtual Network Requests in Cloud-based Data Centers, *IEEE Systems Journal*, Vol. 9, No. 2, pp. 427-441, June, 2015.

[20] L. Chiaraviglio, M. Mellia, F. Neri, Minimizing ISP Network Energy Cost: Formulation and Solutions, *IEEE/ACM Transactions on Networking (TON)*, Vol. 20, No. 2, pp. 463-476, April, 2012.

[21] Y. Zhu, M. Ammar, Algorithms for Assigning Substrate Network Resources to Virtual Network Components, *25th IEEE International Conference on Computer Communications (INFOCOM)*, Barcelona, Spain, 2006, pp. 1-12.

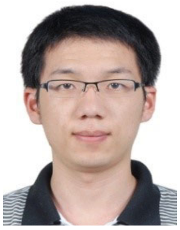
[22] W.-C. Chien, C.-F. Lai, H.-H. Cho, H.-C. Chao, A SDN-SFC-based Service-oriented Load Balancing for the IoT Applications, *Journal of Network and Computer Applications*, Vol. 114, pp. 88-97, July, 2018.

Biographies

Qi Xu received the B.S. degree in telecommunications engineering from Beijing Jiaotong University in 2014, and then entered National Engineering Lab for Next Generation Internet Interconnection Devices at BJTU. Currently, he is pursuing the Ph.D. degree in telecommunications and information system at BJTU, China. His research interests include Software Defined Network, Network Function Virtualization, Service Function Chaining and 5G.

Deyun Gao received BEng and MEng degrees in electrical engineering and a PhD degree in computer science from Tianjin University, China, in 1994, 1999, and 2002, respectively. He spent one year as a research associate with the Department of Electrical and Electronic Engineering, Hong Kong University of Science and Technology, Kowloon. He then spent three years as a research fellow in the School of Computer Engineering, Nanyang Technological University, Singapore. In 2007, he joined the faculty of Beijing Jiaotong University as an associate professor of School of Electronics and Information Engineering and was promoted to a full professor in 2012. In 2014, he was a visiting scholar in University of California at Berkeley, USA. His research interests are in the area of Internet of Things, vehicular networks and next-generation Internet.

Huachun Zhou received the B.S. degree from the People's Police Officer University of China in 1986. He received the M.S. in telecommunication automation and Ph.D. degrees in telecommunications and information system from Beijing Jiaotong University in 1989 and 2008, respectively. In October 1994, he joined Institute of Automation Systems, BJTU, where he is a lecturer. From Apr. 1999 - Sep. 2009, he was a senior engineer at School of Electronic and Information Engineering, BJTU, and at Network Management Research Center, BJTU. From Oct. 2009 to now, he is a professor in National Engineering Lab for Next Generation Internet Interconnection Devices at BJTU. He has authored more than 40 peer-reviewed papers and he is the holder of 17 patents. His main research interests are in the area of mobility management, mobile and secure computing, routing protocols, network management and satellite network.



Wei Quan received his Ph.D. degree from Beijing University of Posts and Telecommunications in 2014 and currently works as a Lecturer at Beijing Jiaotong University, Beijing, China. His research interests include future Internet, vehicular networks and Internet of energy. He has published more than 20 research papers in international journals and conferences, and also serves as the Associate Editor of Journal of Internet Technology (JIT) and technical reviewers in several journals including IEEE Transactions on Parallel and Distributed Systems, IEEE Communications Letters.

Wenfeng Shi received his Bachelor's degree from Taishan University, Taian, China, in 2012. He is pursuing his Ph.D. degree at School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China. His research interests include next-generation networks, delay/disruption tolerant networks and satellite networks.