A Novel Resource Optimization Algorithm for Dynamic Networks Combined with NFV and SDN

Qian Zhang[1], Xiaohua Qiu[2], and Xiaorong Zhu[1](✉)

[1] Nanjing University of Posts and Telecommunications,
Nanjing 210003, China
xrzhu@njupt.edu.cn
[2] Nanjing Institute of Technology, Nanjing 210003, China

Abstract. Various services of Internet of Things (IoT) require flexible network deployment to guarantee different quality of services (QoS). Aiming at the problem of service function chain deployment, in this paper, we propose the combination of NFV and SDN to optimize resources. Considering forwarding cost and traffic load balance, a joint optimization model of virtual network function (VNF) placement and service function chain routing is given and is proved to be NP-Hard. In order to solve this model, we propose two heuristic algorithms. One is the service function chain deployment algorithm of First Routing Then Placing (FRTP) and the other is the Placing Followed by Routing (PFBR) based on node priority. Simulation results show that the former can reduce forwarding times and bandwidth consumption than the latter. And PFBR algorithm outperforms in balancing network traffic load and improving the acceptance ratio of the chain requests compared with other algorithms.

Keywords: IoT · VNF · Service function chain deployment · Node priority · Load balance

1 Introduction

With the development of the Internet of Things, the demands of various services are increasingly diversified [1]. For example, the ideal latency for Internet of Vehicles is almost zero; it is necessary to control the lowest packet loss and ultra-low latency in industrial manufacturing; the mobile video surveillance networks need higher bandwidth.

To satisfy those distinct QoS requirements simultaneously, Software Defined Networking (SDN) could offer smart routing and scheduling solutions. Network Function Virtualization (NFV) uses virtualization to eliminate dependencies on

---

Foundation Items: National Natural Science Foundation of China (61871237), National Science and Technology Major Project (2017ZX03001008), Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX17_0767), Natural Science Foundation of the Higher Education Institutions of Jiangsu Province (16KJA510005) and YKJ201422.

M. Jia et al. (Eds.): WiSATS 2019, LNICST 281, pp. 283–296, 2019.
https://doi.org/10.1007/978-3-030-19156-6_27

dedicated hardware and consolidates different types of network devices onto industry-standard and high-capacity servers, switches or storage [2], which reduces costs of network equipment meanwhile enhancing the sharing capacity of network resources and quality of service. And virtualization in the form of network slicing could be used to isolate IoT use-cases with conflicting requirements.

Although the combination of SDN and NFV is expected to meet the diversified demands of services of the Internet of Things, NFV abstracts and finely decomposes network services, making its components more complex. The service function chain (Service Function Chain, SFC) deployment is the key challenge. Due to its large size of services and high dynamic network load, the complex service-driven dynamic network construction mechanism of Internet of Things is a challenge for both the current and future network. The existing dedicated network faces severe defects such as stiff network architecture, unreasonable function plane partitioning and difficulty in network upgrades and maintenance, which makes it hard to cope with the ever-changing requirements of the Internet of Things.

For the resource allocation of NFV, the authors of [3] propose a comprehensive cost model to meet the needs of users and achieve the lowest cost. Moens et al. consider the scenario where physical network functions and virtual network functions coexist [4], then decide what kind of the functions to be placed to make full use of the physical and virtual resources. Reference [5] uses graph theory to solve the problem of resource allocation in NFV, where network functions and servers (with multiple virtual machines) are regarded as two disjoint subsets in a bipartite graph. The resource utilization and the bandwidth utilization are weighed with the coefficients so as to achieve load balancing in the network [6]. Nam et al. [7] introduce the NFV to the radio access network. It is possible to allocate nearby base stations to obtain a lower latency experience. NFV is extended to the radio access section [8], and the main drawback is that each function can only be deployed once, and be placed on different nodes, which, to some extent, introduces link latency. Reference [9] locates the scene in the enterprise network, and it is committed to reducing the total cost of deploying virtual network functions. In order to implement service chain deployment, [10] considers sharing virtual network functions among service chains (SCs) and avoiding bandwidth consumption by deploying adjacent VNFs in one node. Reference [11] aims at minimizing NFV nodes as an optimization target for the VNFs sharing of multiple SFCs.

Although an increasing number of the researches on NFV resource allocation and SDN routing strategies, there are still many areas need to be improved. For example, in most researches, VNFs required by service function chains merely occupy a single kind of resource, and almost do not consider the case where multi-dimensional resources such as computation and storage are needed at the same time. In addition, some papers focus on one aspect of NFV resource allocation or SDN routing strategies, resulting in local optimal solution. How to coordinate the relationship between them deserves further study.

In this context, this paper studies the resource allocation method of NFV and SDN in dynamic service network. According to the dynamic changes of network status and service characteristics, a joint optimization model is proposed to determine the resource allocation and routing strategies of service-driven optimal transmission performance. In order to facilitate the optimization model, we propose two heuristic algorithms according to the order of VNFs placement and service flow routing. One is the service function chain deployment algorithm of First Routing Then Placing (FRTP) and the other is the Placing Followed by Routing (PFBR) based on node priority. The former is more focused on reducing forwarding times and bandwidth consumption than the latter. And PFBR algorithm outperforms in balancing network traffic load and improving the acceptance ratio of the chain requests compared with other algorithms.

The rest of this paper is organized as follows. Section 2 describes the problem of deployment of service function chain. In Sect. 3, a joint optimization model of virtual network function placement and service function chain routing is given. We present our solutions in Sect. 4. In Sect. 5, we validate the algorithms and discuss the results. Section 6 concludes the paper.

2 System Description

In this paper we assume that the service function chain deployment involves two aspects. In one aspect, we place a series of virtual network functions on the NFV enabled nodes to meet user-customized network services. And the other is that, the paths are selected according to the demands of service function chain until a complete end-to-end service path is formed.

In order to perform the resource allocation of the service function chains, we model the physical network as a undirected graph denoted by $G = (V, E)$, where $V = \{v_n | n = 1, 2, \ldots, N\}$ represents the set of nodes and E is the set of the links. Resource capacity of each node is C_n^t and is associated with its type $t \in T$ (e.g., computing, storage and so on). To be noted that if $C_n^t > 0, t \in T$, the node v_n can host different VNFs unless its capacity is less than the demand of VNFs. Otherwise, the node v_n only be able to forward packets. Each link $e \in E$ is associated with a bandwidth capacity B_e.

There are many kinds of VNFs, f_x represents the x_th kind in the VNF set $F = \{f_x | x = 1, 2, \ldots, X\}$. The resource demands of f_x is $C_x^t, t \in T$. Service function chain is composed by a series of VNFs and need to be traversed in particular order. Suppose there are a total of K service function chains in a period of time, the set of which can be expressed as $S = \{S^k | k = 1, 2, \ldots, K\}$. For S^k, it contains source node, destination node and a series of VNFs and can be expressed as $N^k = \{s^k, f_1^k, f_2^k, \ldots, f_M^k | f_m^k \in F, d^k\}$, where $m = 0$ and $m = M + 1$ represents the source and destination node respectively. Then, the set of neighbor nodes of f_m^k are represented as $\eta^k(m)$, which indicates the adjacent virtual functions of f_m^k.

And $E^k = \{(s^k, f_1^k), (f_1^k, f_2^k), \ldots, (f_{M-1}^k, f_M^k), (f_M^k, d^k)\}$ describes the logical link set. Suppose that the QoS of S^k can be represented by r^k, the demand of bandwidth. In addition, $q_{mx}^k \in \{0, 1\}$ is introduced in our model. It implies whether the kind of f_m^k in S^k is f_x and hence its resource requirement is indicated as $\sum_{x \in X} q_{mx}^k \cdot C_x^t, t \in T,\ T = \{\text{computing, storage,} \ldots\}$.

As shown in Fig. 1, the service function chain requires five VNFs. On the one hand, five virtual network functions are assigned at physical node 1, 2, 3, 4, 3. On the other hand, the computed path by the controller is (s,1)→(1,2)→(2,3)→(3,5)→(5,4)→(4,3)→(3,t). Both of the two aspects represent resource allocation and link mapping in one service function chain.

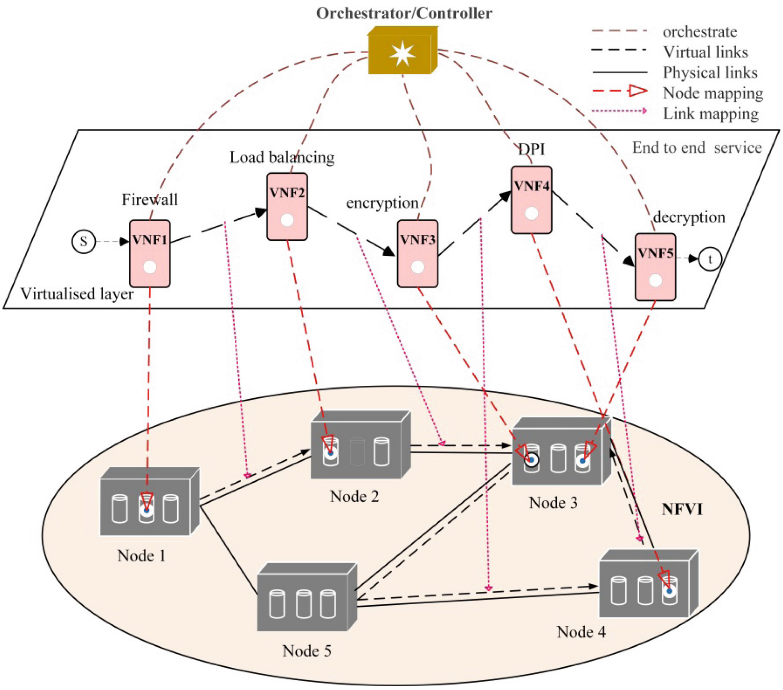

Fig. 1. Resource allocation and link mapping in one service function chain

As a consequence, we define several sets of variables to indicate above process. The first is $a_m^k(n)$, and it equals 1 if f_m^k in S^k is placed on node v_n, and 0 otherwise. The next is $l_{m_1 m_2}^{k,p}(n_1, n_2)$, and it equals 1 when the logical link $(m_1, m_2) \in E^k$ of S^k is routed by the path p from node n_1 to n_2, where $p \in P(n_1, n_2)$ and $P(n_1, n_2)$ is the link set between node n_1 and n_2. The last is $\varphi^k \in \{0, 1\}$, it equals 1 if the S^k is accepted, and 0 otherwise. Furthermore, if $\varphi^k = 0$, then $\sum_{n \in N} a_m^k(n) = 0$ and $\sum_{p \in P(n_1,n_2)} l_{m_1 m_2}^{k,p}(n_1, n_2) = 0$.

3 Service Function Chain Deployment

The service function chain is regard as a form of service. From the user's point of view, the key is real-time response. But from long-term of network operation, it is essential to improve resource utilization. Accordingly, the optimization goal of this paper consists of two aspects. One is to minimize the total forwarding costs of service function chains as (1). It is related to forwarding times and bandwidth requirements. And the other is to balance the traffic load as much as possible. We use the degree of load balancing to measure the maximum link utilization in the network and they are represented as (2) and (3) respectively. The lower degree of load balancing is, the more balanceable the traffic loads are.

It should be noted that if the request is not deployed, the total forwarding cost and the link utilization are lower. However, which cannot satisfy their needs. Penalty functions are introduced when SFCs are rejected, and they are expressed as (4) and (5). Therefore, the overall objective is formulated as (6), where coefficients and indicate the weight of the two different aspects.

$$C_{forward} = \sum_{S^k \in S} \sum_{(m_1,m_2) \in E^k} \sum_{n_1,n_2 \in N} \sum_{p \in P(n_1,n_2)} l_{m_1 m_2}^{k,p}(n_1, n_2) \cdot r^k \cdot |p| \tag{1}$$

$$d_{load} = \max_{e \in E} R_e / B_e \times 100\% \tag{2}$$

$$R_e = \sum_{S^k \in S} \sum_{(m_1,m_2) \in E^k} \sum_{n_1,n_2 \in N} \sum_{p \in P(n_1,n_2)} l_{m_1 m_2}^{k,p}(n_1, n_2) \cdot r^k \tag{3}$$

$$C_{penalty} = \sum_{S^k \in S} (1 - \varphi^k) r^k \cdot \max_{p \in P(s^k, d^k)} |p| \tag{4}$$

$$d_{penalty} = \left(\sum_{S^k \in S} (1 - \varphi^k) r^k \right) \Big/ \min_{e \in E} B_e \tag{5}$$

$$\min \alpha \cdot (C_{forward} + C_{penalty}) + \beta \cdot (d_{load} + d_{penalty}) \tag{6}$$

$$\sum_{n \in N} a_m^k(n) \le 1, \quad \forall m \in N^k, S^k \in S \tag{7}$$

$$\sum_{f_n^k \in s^k} \sum_{n \in N} a_m^k(n) = \varphi^k \cdot \left(\left| N^k \right| - 2 \right), \quad \forall S^k \in S \tag{8}$$

$$\sum_{S^k \in s} \sum_{f_m^k \in S^k} \sum_{x \in X} a_m^k(n) \cdot q_{mx}^k \cdot C_x^t \le C_n^t, \forall t \in T, \forall n \in N \tag{9}$$

$$\begin{gathered} \sum_{n_1,n_2 \in N} \sum_{p \in P(n_1,n_2)} l_{m_1 m_2}^{k,p}(n_1, n_2) = a_{m_1}^k(n_1) \times a_{m_2}^k(n_2) \\ \forall (m_1, m_2) \in E^k, \forall S^k \in S \end{gathered} \tag{10}$$

$$\begin{gathered} l_{m_1 m_2}^{k,p_1}(n_1, n_2) + l_{m_1 m_2}^{k,p_2}(n_2, n_1) \le 1 \\ \forall n_1, n_2 \in N, \forall S^k \in S, \forall (m_1, m_2) \in E^k \\ \forall p_1 \in P(n_1, n_2), p_2 \in P(n_2, n_1) \end{gathered} \tag{11}$$

$$\begin{gathered} \sum_{S^k \in S} \sum_{(m_1,m_2) \in E^k} \sum_{n_1,n_2 \in N} \sum_{p \in P(n_1,n_2)} l_{m_1 m_2}^{k,p}(n_1, n_2) \cdot r^k \le B_e \\ \forall e \in P(n_1, n_2) \end{gathered} \tag{12}$$

$$\begin{gathered} \sum_{n_2 \in N} l_{m_1 m_2}^{k,p_1}(n_1, n_2) - \sum_{n_2 \in N} l_{m_1 m_2}^{k,p_2}(n_2, n_1) = a_{m_1}^k(n_1) - a_{m_2}^k(n_1) \\ \forall (m_1, m_2) \in E^k, \forall S^k \in S \\ \forall p_1 \in P(n_1, n_2), \forall p_2 \in P(n_2, n_1). \end{gathered} \tag{13}$$

$$\begin{aligned} & r^k \leq \min(B_e - R_e),\ e \in P(n_1, n_2), l^{k,e}_{m_1 m_2}(n_1, n_2) = 1 \\ & \forall n_1, n_2 \in N, \forall S^k \in S \end{aligned} \tag{14}$$

Constraint (7) indicates that any VNFs other than the source and destination of the SFCs, at most selects one physical node to be placed; the constraint (8) indicates whether S^k is accepted, if it is, the required VNFs must be allocated one physical node to host them. Constraints (9) indicate that the resources required by the VNFs cannot exceed the resources available on the nodes. $a^k_{m_1}(n_1) \times a^k_{m_2}(n_2)$ of constraints (10) represents the adjacent VNFs $f^k_{m_1}$ and $f^k_{m_2}$ in S^k are placed on n_1 and n_2 respectively, and one path between them should be routed. Constraint (11) represents each logical link in one SFC cannot be mapped to a bi-directional link; the constraint (12) indicates that the aggregation bandwidth on the physical link cannot exceed the link bandwidth; Constraints (13) ensures that the logical links of S^k map to a continuous path of the physical topology, and constraints (14) guarantee quality of service.

The resource optimization problem of combination of NFV and SDN studied in this paper is a NP-hard problem [10]. If all the nodes are guaranteed to have the same available resources, the node can be regarded as a fixed-capacity packet and the requested VNFs are objects in different sizes. It is converted to a normal problem called binary-knapsack. Suppose that B_e is infinite, which means that the routing is not important, the problem becomes a pure VNF placement, which has been proved to be NP-hard. Therefore, the resource optimization problem of combination of NFV and SDN studied in this paper is proved to be an NP-hard problem.

4 Heuristic Algorithms

In order to solve above NP-hard problem in reasonable time, we propose two heuristic algorithms according to the order of VNFs placement and traffic flow routing. First Routing Then Placing (FRTP) algorithm is to choose paths for SFCs firstly, and then place the required VNFs orderly along the chosen paths. Another Placing Followed by Routing (PFBR) algorithm is just the opposite. Firstly, the VNFs required by SFCs are assigned to some nodes whose priorities are relatively high. Then, the source node, the intermediate node and the destination node are connected together into a complete path. During the placement phase of PFBR algorithm, the priority of nodes is given as (15).

$$w(v_n) = \left(1 - \frac{hop_n}{\max_{j \in N}(hop_j)}\right) \cdot \left(\frac{rec_n}{\max_{j \in N}(rec_j)}\right) \tag{15}$$

The priority of a node is determined by the available resources of the node and the number of hops introduced by the node, where, indicates the number of forwarding hops introduced by the node, represents the total amount of available resources of the node, indicates the maximum number of forwarding hops and implies the maximum amount of available resources.

Algorithm 1. First Routing Then Placing algorithm

```
1: function FRTP ($S$, $F$, $G=(V,E)$)
2: order SFC ← order SFCs by $r^k$ in decreasing order
3: cost=0; Accept=0; $p_k$, paths={0}; $\pi_k, \pi = 0$;
4: $\forall S^k \in S$:
5: if $r^k > \lambda$ then
6:     $p \leftarrow$ Shortest_path($G$, $s^k, t^k$)
7:     if $D_k^t \le \sum_{v_n \in p} C_n^t, \forall t \in T$ && $r_k \le \min_{e \in p}(B_e - R_e)$ then
8:         $\forall f_m^k \in N^k$, $v_n \in p$:
9:         place $f_m^k$ to $v_n$;
10:        $\pi_k \leftarrow \pi_k$ U $v_n$;  $p_k \leftarrow p_k$ U $p$;
11:        cost=cost+ $r^k \cdot |p_k|$;  Accept++;
12:        update $C_n^t$, $B_e$;
13:    else
14:        Remove $p_k$ in $G$; continue;
15:    end if
16: else $p \leftarrow$ K_Shortest_path($G$, $s^k, t^k$);
17:    repeat 9-20;
18: end if
19: Add $\pi_k$ to $\pi$; $p_k$ to paths;
20: end function
```

Both of the FRTP and PFBR algorithm proposed in this paper decide different paths for requests with different bandwidth requirements. Once the request arrives, we judge whether the demand of bandwidth exceeds the threshold λ. If it exceeds, the request is regarded as a high bandwidth service request, we try to guide traffic to the shortest path so as to reduce forward times and bandwidth consumption. Otherwise, we try to search a path which has more available link resources, which can balance the network traffic load. Algorithms 1 and 2 give the pseudocode of the heuristic solutions respectively. Both of them take as input a set of SFC requests S, and a set of VNFs annotated with the network topology graph. In initial step, we order SFCs by r^k in decreasing order.

Besides, the outputs include π, a set of nodes which host the required VNFs, routing paths and etc.

Algorithm 2. Placing Followed by Routing algorithm

```
1: function PFBR ($S, F, G=(V,E)$)
2: $order$ SFC←order SFCs by $r^k$ in decreasing order
3: cost=0; Accept=0; $p_m^k, p_k, paths=\{0\}; \pi_k, \pi = 0$;
4: $\forall S^k \in S$ :
5: calculate $w(v_i) = (1 - \frac{hop_i}{\max_{j \in N}(hop_j)}) \cdot (\frac{rec_i}{\max_{j \in N}(rec_j)})$;
6: $order\, v_n$ ← order $v_i$ by $w(v_i)$ in decreasing order
7: $\forall f_m^k \in S^k$ , $v_n \leftarrow order\, v_i$ :
8: if $D_k^t \le C_n^t, \forall t \in T$ then
9:   place $f_m^k$ to $v_n$ ;
10:   $\pi_k \leftarrow \pi_k$ U $v_n$ ;
11:   if $r^k > \lambda$ then
12:     $\forall (s^k, \pi_k), (\pi_k, t^k)$ :
13:     $p_1^k \leftarrow$ Shortest_path($G$, $s^k, \pi_k$);
14:     if $r_k \le \min_{e \in p_1^k}(B_e - R_e)$ then
15:       Add Add $p_1^k$ $to$ $p_k$ ;
16:       cost=cost+ $r^k \cdot |p_k|$ ;Accept++;
17:       update $C_n^t$, $B_e$ ;
18:     else Remove $p_k$ in $G$; continue;
19:     end if
20:   else
21:     $p_1^k \leftarrow K$_Shortest_path($G$, $s^k$, $\pi_k$);
22:     repeat 17-19;
23:   end if
24: else
25:   Remove $v_n$ in $G$; continue;
26: end if
27: Add $\pi_k$ to $\pi$; $p_k$ to $paths$
28: end function
```

5 Experiment Result

In order to evaluate the algorithms proposed in this paper, we verify the feasibility of the proposed model and the effectiveness of the proposed algorithm. In terms of total forwarding costs and degree of load balancing coupled with the acceptance ratio of requests, the proposed heuristic algorithms are compared with Routing before Placement (RBP) algorithm proposed in [12]. During the routing phase, RBP algorithm is always searching shortest path for each demand unless resource constraints are violated. Then, VNFs are placed on the path greedily. If no such path satisfies all resource constraints, the demand is rejected.

5.1 Experimental Environment and Settings

In simulation experiment, we use Matlab2015a for program simulation and run them on the Windows 10 with Intel Core i7-6900, 3.40 GHz CPU and 8 GB RAM.

Our experiment adopts random network topology with 20 to 100 nodes. Each node has three resource types. The total amount of resources follows a uniform distribution with an average of 100 and a variance of 30, and the average link capability is about 200. There are 10 types of VNFs in the VNFs set. And 3 to 6 kinds of which are required of a SFC. Each type of resource is required by one VNF is 0.4 to 1 unit. The bandwidth demand r^k of each SFC varies from 1 to 5 units, and λ is the average of overall bandwidth requirements. In our experiment, the number of SFCs is increased from 200 to 1000.

5.2 Performance Analysis

Figure 2 demonstrates that total forwarding cost increases with the number of SFCs. When the nodes in the network is fixed as 50, the total forwarding cost of FRTP algorithm is lower than that of PFBR based on node priority, and the results of them are higher than RBP algorithm. However, only consider the cost of forwarding is not comprehensive, because it is related with the forward times and bandwidth demands of SFCs. Figure 3 shows the degree of load balancing as the number of SFCs increases. In the aspect of load balancing, FRTP algorithm is inferior to that of PFBR algorithm based on node priority, and it shows stronger advantages than the RBP algorithm without load balancing.

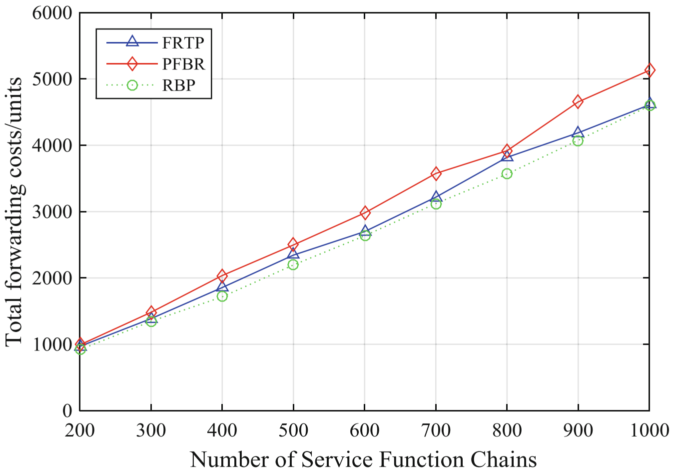

Fig. 2. Total bandwidth occupation of the number of SFCs

When the network size gradually increases, the total forwarding costs do not vary greatly. This is due to the fact that the average distances generated by random networks are about the same. RBP algorithm holds a slight advantage in forwarding cost, while FRTP and PFBR algorithm show little difference. Figure 4 shows the degree of load balancing when the network size gradually increases. Generally, PFBR algorithm outperforms than FRTP algorithm in load balancing, and RBP algorithm is inferior to the algorithms proposed in this paper. The average running time of the algorithms increases with the increasing of the number of SFCs and nodes. As shown in Figs. 5 and 6, the average running time of FRTP algorithm is lower than that of PFBR algorithm, because the latter takes more time to calculate the node priority. RBP algorithm takes least time in the traffic routing phase without load balancing.

Furthermore, a set of experiments are carried out with 20 nodes to verify the requests acceptance ratio. Figure 7 shows the acceptance ratio of various algorithms under different number of SFCs. Obviously, PFBR algorithm can balance the traffic load better than other algorithms, and can make greater use of resources. As a result, the acceptance ratio of requests is obviously improved compared with other algorithms.

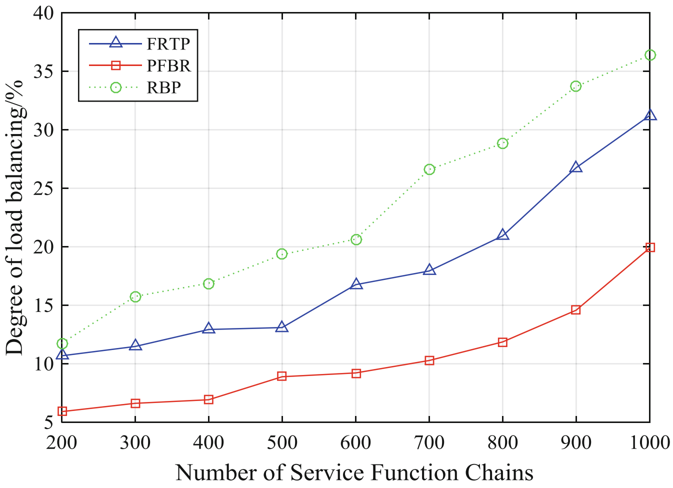

Fig. 3. Degree of load balancing versus the number of SFCs

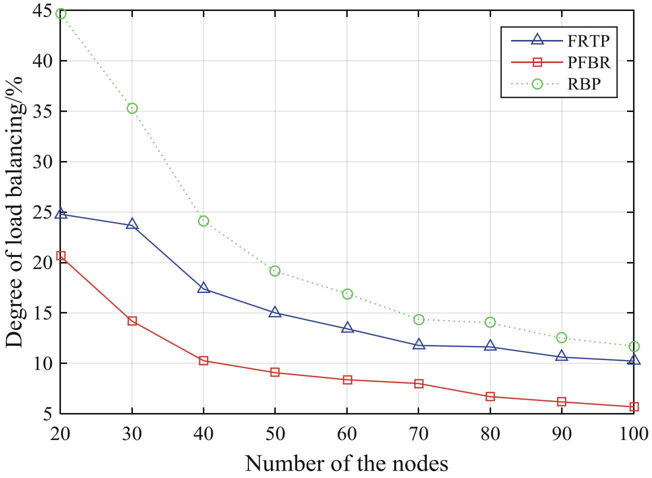

Fig. 4. Degree of load balancing versus nodes in the network

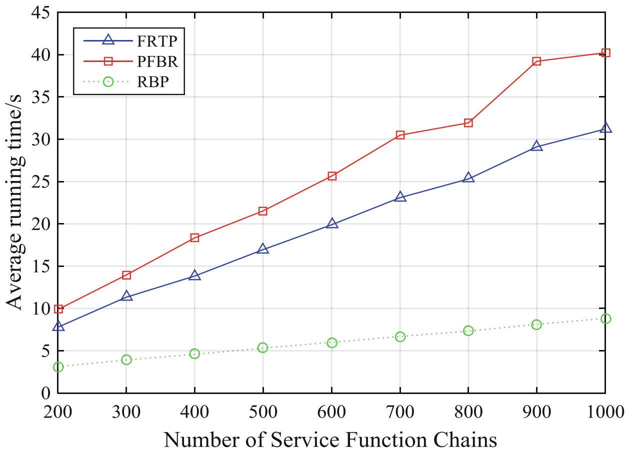

Fig. 5. Average running time of the number of SFCs

To sum up, compared with RBP algorithm, the two algorithms proposed in this paper can more remarkably balance the traffic load and improve the requests acceptance ratio at the expense of slightly high forwarding cost. However, compared with PFBR algorithm, FRTP algorithm focuses more on reducing the total forwarding cost and enables a large number of traffic to be rapidly forwarded in the network. It can also be used as an online algorithm to deployment real-time service function chains. On the other hand, PFBR algorithm can effectively balance the network traffic load and improve the overall link utilization in the network even the number of SFCs or nodes are large.

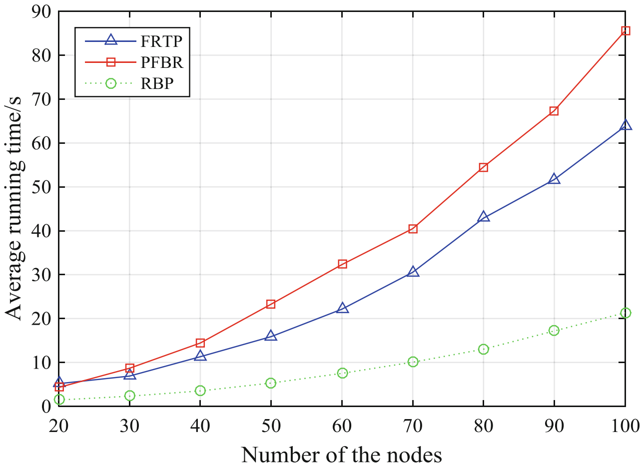

Fig. 6. Average running time versus the number of nodes in the network

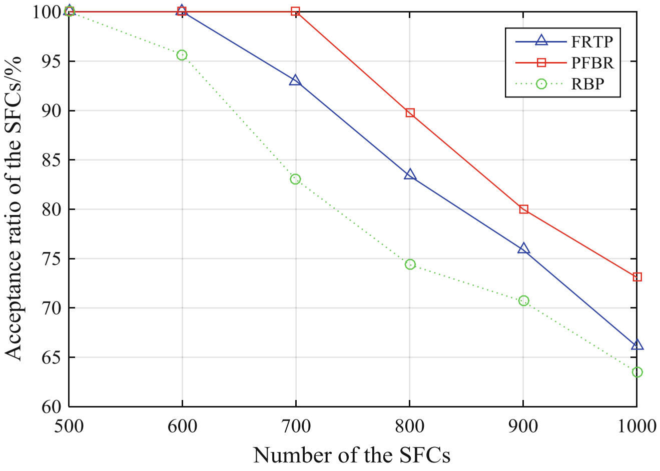

Fig. 7. Acceptance ratio under different intensity of SFCs

6 Conclusion

In this paper, we propose a joint optimization model of service function chain placement and routing algorithm. Then two heuristic algorithms are proposed according to the order of VNFs placement and traffic flow routing. The simulation results show that FRTP can make a large number of data packets be rapidly forwarded, which can reduce service function chaining time and improve user experience to a certain extent. Although PFBR algorithm is slightly inferior in reducing the total forwarding cost, the advantage in load balancing is prominent, which improves the request acceptance ratio of requests. The beneficial combination of the two algorithms enables more flexible deployment of the service function chains. However, given the traffic load balancing in the network, the time overhead of proposed two algorithms increases rapidly when the scale of network and the number of service function chains get larger. Therefore, in the future work, we expect to study more optimized methods such machine learning to balance the load and improve the ratio of acceptance of requests while reducing the time overhead as much as possible.

References

1. Bizanis, N., Kuipers, F.A.: SDN and virtualization solutions for the Internet of Things: a survey. IEEE Access **4**, 5591–5606 (2016)
2. Mechtri, M., Ghribi, C., Soualah, O., Zeghlache, D.: NFV orchestration framework addressing SFC challenges. IEEE Commun. Mag. **55**(6), 16–23 (2017)
3. Wang, L., Lu, Z.M., Wen, X.M., Knopp, R., Gupta, R.: Joint optimization of service function chaining and resource allocation in network function virtualization. IEEE Access **4**, 8084–8094 (2016)

4. Moens, H., Turck, F.D.: Customizable function chains: managing service function chain variability in hybrid NFV networks. IEEE Trans. Netw. Serv. Manag. **13**(4), 711–724 (2016)
5. Kanizo, Y., Rottenstreich, O., Segall, I., Yallouz, J.: Optimizing virtual backup allocation for middleboxes. IEEE/ACM Trans. Networking **25**(5), 2759–2772 (2017)
6. Leivadeas, A., Falkner, M., Lambadaris, I., Kesidis, G.: Resource management and orchestration for a dynamic service function chain steering model. In: IEEE Global Communications Conference (GLOBECOM), Kiev, Ukraine, pp. 1–6 (2017)
7. Nam, Y., Song, S., Chung, J.M.: Clustered NFV service function chaining optimization in mobile edge clouds. IEEE Commun. Lett. **21**(2), 350–353 (2017)
8. Riggio, R., Bradai, A., Harutyunyan, D., Rasheed, T., Ahmed, T.: Scheduling wireless virtual networks functions. IEEE Trans. Netw. Serv. Manag. **13**(2), 240–252 (2016)
9. Bari, M.F., Chowdhury, S.R., Ahmed, R., Boutaba, R., Duarte, O.C.M.B.: Orchestrating virtualized network functions. IEEE Trans. Netw. Serv. Manage. **13**(4), 725–739 (2016)
10. Sun, Q.Y., Lu, P., Lu, W., Zhu, Z.Q.: Forecast-assisted NFV service function chain deployment based on affiliation-aware vNF placement. In: IEEE Global Communications Conference (GLOBECOM), pp. 1–6 (2017)
11. Savi, M., Tornatore, M., Verticale, G.: Impact of processing costs on service function chain placement in network functions virtualization. In: IEEE Conference on Network Function Virtualization and Software Defined Network, pp. 191–197 (2015)
12. Nguyen, T.M., Fdida, S., Pham, T.M.: A comprehensive resource management and placement for network function virtualization. In: 2017 IEEE Conference on Network Softwarization (NetSoft), Bologna, Italy, pp. 1–9 (2017)