A Quick Adaptive Migration Algorithm for Virtual Network Function

Yizhong Wang[1], Xiaorong Zhu[1](✉), and Xiaohua Qiu[2]

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

Abstract. The combination of software defined network (SDN) and network function virtualization (NFV) solves some problems in traditional networks, such as service deployment and configuration and management of network resources. However, it also introduces new problems such as network load imbalance. Virtual network function (VNF) migration is an effective way to solve these problems. In this paper, we propose a quick adaptive migration algorithm for VNF, which combines pre-calculation and real-time calculation to reduce the cost of migration. When the node triggers the light-overload-threshold, we perform a pre-calculation of migration for the node and set the result-set. When the node is overloaded, we perform the migration if the result-set is unexpired, otherwise we perform the real-time migration solution. Simulation results show that this algorithm can effectively reduce the number of migration, improve the stability of the system and reduce the overall network migration overhead of the system.

Keywords: Virtual network function · Migration algorithm · Multi-objective decision-making

1 Introduction

With the development of network and the large-scale application of Internet of things (IoT) technology, massive terminal access network. Traditional network architecture is confronted with many problems, such as bloated protocols, difficult business deployment and inflexible resource scheduling [1], so it is urgent to propose new technology to change this situation. Network function virtualization (NFV) and software defined network (SDN) came into being. NFV [2] technology can run the virtual network function on the general hardware, SDN can control the traffic forwarding path, and make the network programmable.

Currently, there have been cases with NFV and SDN technology in the network [3]. In SDN-NFV networks, programmable switches forward targeted traffic to the

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

M. Jia et al. (Eds.): WiSATS 2019, LNICST 281, pp. 333–347, 2019.
https://doi.org/10.1007/978-3-030-19156-6_31

appropriate network functional unit for processing, so dynamic deployment of network functions may result in unbalanced load between network control domains.

To address the issue of load imbalance in such NFV and SDN deployments, researchers introduce virtual network function (VNF) migration. VNF migration means removing VNF from the original physical machine and redeploying at a better location. VNF migration and virtual machine migration have similarities in some respects. Both of them occupy various types of network resources of the nodes, and the destination nodes of migration need to meet their performance requirements. However, the biggest difference between VNF migration and virtual machine migration [4] lies in that the VNF instances have a specific type and belong to one or several service function chains, and the service function chains usually have end-to-end performance requirements.

The researchers proposed different VNF migration strategies and migration methods from different perspectives. VNF has a variety of migration strategies, such as the horizontal scaling technology in the literature [5], the resources occupied by the VNF are allocated when being instantiated. When the demand changes, VNF instance should be directly removed without being changed. The vertical scaling technique is adopted in the literature [6–8], and the resources allocated to VNF are distributed dynamically with the change of business requirements. In this paper, the vertical scaling technology is used. When a VNF allocates a certain amount of resources during instantiation, when the requirements change, the resources occupied by the VNF change correspondingly until the VNF is triggered to migrate.

In the literature [9], a traffic cycle model and a system cost function are proposed, and the problem of VNF migration is transformed into a periodic migration problem of mapping strategy. The scenario proposed in this solution is a data center with a low traffic period, which cannot effectively migrate the traffic non-periodic scene. A distributed elastic control algorithm based on switch migration is proposed in [10]. This migration scheme can achieve load balancing among nodes, but the end-to-end delay constraint of the SFC is not considered in the migration process.

In view of the deficiencies of the above methods, this paper proposes a fast adaptive migration algorithm for network functions based on multidimensional environment perception, and adds the coordination of migration pre-calculation and real-time computation to reduce the real-time computation cost of network migration. The structure of the article is as follows: The second part gives the model of VNF migration problem in NFV network. The third part gives the optimization problem of reducing the network migration cost and reducing the number of VNF migration, and proposes a method of migration pre-calculation and real-time calculation. In the fourth part, we present a heuristic algorithm to solve the problem, which is used to solve the selection problem of the VNFs to be migrated and the migration destination nodes. The simulation results and analysis will be given in the fifth part, and the last part will give the conclusion.

2 System Model

This chapter will present the problem of VNF migration in NFV network. A network model is designed to describe the dynamic change of demand in NFV network, the occupation of various resources by VNF, and the conditions of triggering VNF migration.

2.1 Problem Statement

In SDN and NFV networks, there are many nodes, and each node has computing, bandwidth and storage resources. According to service requirements, we place the corresponding service function chain in the network. Each SFC consists of a series of ordered virtual network functions. According to the end-to-end delay performance requirement of the SFC, the VNF is placed on an appropriate node and occupies various resources of the node.

With the change of service requirements, due to the use of vertical scaling technology, the resources occupied by the corresponding VNFs also change. The scenario studied in this paper is to achieve fast network load balancing by reasonably selecting VNFs to be migrated and migrated destination nodes.

In the migration process, the performance requirements of the VNF to be migrated and the end-to-end performance needs of the service function chain to be migrated need to be considered. By reasonably selecting the destination node, the overall system migration cost and the migration times are reduced.

2.2 Network Model

We use one directed graph, $G = (V^{Node}, E)$, to represent the physical network, where V^{Node} is the set of the nodes, and E is the set of its edges. The characteristics of server nodes and links are described as follows:

- N_w^{core}: processing capacity of the node w in term of the number of cores that it has;
- N_w^{mem}: memory capacity of the node w in term of the number of gigabytes that it has;
- C_{ij}: bandwidth of the physical link between node i and node j;

V^{VNF} represents the set of virtual network function (VNF), The characteristics of VNF are described as follows:

- n_y^{core}: processing capacity of the VNF y;
- n_y^{mem}: memory capacity of the VNF y;
- c_y: bandwidth capacity of the VNF y;
- a_{ij}: binary variable assuming the value 1 if the VNF v_i is embedded in node n_j, otherwise its value is 0;

The usage of a certain resource of a node is defined as the ratio of the used resource in the current node to the total resource. The condition for triggering the migration calculation is that any resource usage is greater than its preset threshold $T_k, T_k \in (0,1), k = 1,2,3$, and each type of resource has its own threshold.

The set of service function chains (SFC) is $F = \{f_1, f_2, \ldots, f_n\}$, the maximum time delay constraint for SFC f_n is τ_n, therefore, when the VNF belongs to SFC f_n (only consider that each VNF instance belongs to a SFC), the sum of the additional delay τ_+

and the original delay τ_0 should be less than τ_n, i.e. $\tau_+ + \tau_0 < \tau_n$. We define a boolean variable b_{ik} to denote whether virtual network function v_i migrates or not, i.e., $b_{ik} = 1$ if the virtual network function v_i is moved to network node n_k, otherwise $b_{ik} = 0$.

The destination node should not overburden after the VNF moves in. It can be expressed by

$$\sum_{l \in V^{VNF}} n_l^{core} \cdot a_{lk} + n_i^{core} \cdot b_{ik} < T_1 \cdot N_{w_k}^{core} \tag{1}$$

$$\sum_{l \in V^{VNF}} n_l^{mem} \cdot a_{lk} + n_i^{mem} \cdot b_{ik} < T_2 \cdot N_{w_k}^{mem} \tag{2}$$

$$\sum_{l \in V^{VNF}} c_l \cdot a_{lk} + c_i \cdot b_{ik} < T_3 \cdot \sum_{k \neq j, l \in V^{VNF}} C_{jk} \tag{3}$$

The cost of migration $F(cost, times)$ is defined as a function of migration overhead $f(cost)$ and the number of migrations $f(times)$.

$$F(cost, times) = \lambda \cdot f(cost) + \mu \cdot f(times) \tag{4}$$

where λ and μ are weight coefficients.

Migration overhead $f(cost)$ is the sum of VNF migration calculation time and implementation migration time. The number of migrations $f(times)$ is the number of VNF migrations that occur during system operation. Implementation migration time [5] defines the out-of-service time due to the VNF implementing the migration. The downtime of VNF v_i is defined as the time for VNF v_i to migrate from node n_j to node n_k.

3 The Optimization Problem

In this chapter, we propose a multi-objective optimization problem to reduce the VNF migration overhead and the number of VNF migrations in the NFV network and improve the system stability. In order to reduce the real-time computing overhead of VNF migration in the system, a method of collaborative migration pre-calculation and real-time calculation is proposed to reduce the system migration cost.

3.1 System Modeling

This paper proposes a migration algorithm that migrates virtual network functions from overload node to idle node. Through the perception of computing resources, storage resources and communication resources, we can solve the problem of the migration of virtual network function (VNFMP) and realize Network Load balancing.

$$
\begin{aligned}
\min \quad & \{\lambda \cdot f(cost) + \mu \cdot f(times)\} \\
subject\ to: & \sum a_{ij} = 1 \quad a_{ij} \in \{0,1\} \\
& \frac{\sum_{i \in V^{VNF}} n_i^{core} \cdot a_{ij}}{N_{w_j}^{core}} < T_1, \quad \forall w \in V^{Node} \\
& \frac{\sum_{i \in V^{VNF}} n_i^{mem} \cdot a_{ij}}{N_{w_j}^{mem}} < T_2, \quad \forall w \in V^{Node} \\
& \frac{\sum_{i \in V^{VNF}} c_i \cdot a_{ij}}{\sum_{k \neq j, k \in V^{Node}} C_{jk}} < T_3, \quad \forall w \in V^{Node} \\
& \sum_{l \in V^{VNF}} n_l^{core} \cdot a_{lk} + n_i^{core} \cdot b_{ik} < T_1 \cdot N_{w_k}^{core} \\
& \sum_{l \in V^{VNF}} n_l^{mem} \cdot a_{lk} + n_i^{mem} \cdot b_{ik} < T_2 \cdot N_{w_k}^{mem} \\
& \sum_{l \in V^{VNF}} c_l \cdot a_{lk} + c_i \cdot b_{ik} < T_3 \cdot \sum_{k \neq j, l \in V^{VNF}} C_{jk} \\
& a_{ij} \cdot b_{ik} \cdot \Delta\tau_{jk+} + \tau_0 < \tau_{max}, v_i \in f_n
\end{aligned}
\tag{5}
$$

where T_1, T_2, T_3 respectively represent the thresholds of occupancy rate of computing, storage and communication resources in network nodes. $\tau_{\max}$ represents the maximum tolerable delay of the service function chain to which VNF v_i belongs.

The objective function represents that the cost of migration is defined as a function of migration overhead and the number of migrations. The first constraint ensures that each virtual network feature must be deployed on only one network node. The second to the fourth constraints ensure that the occupancy of computing, storage and communication resources of each network node does not exceed its preset threshold. The fifth to the seventh constraints ensure that the destination network node is not overloaded after the virtual network function is moved in. The last constraint ensures that the sum of the original delay and the additional delay caused by the virtual function migration is smaller than the upper limit of the delay of the service function chain to which it belongs.

3.2 Cooperation of Migration Pre-calculation and Real-Time Calculation

The main migration overhead for VNF migration is the time overhead for solving VNF migration scenarios and the time overhead for implementing VNF migration. When the number of nodes increases, the time and cost for the migration plan calculation increases sharply. In order to reduce the time overhead of solving the migration scheme, this paper presents a migration algorithm that combines migration pre-calculation and real-time calculation. We set the appropriate light overload threshold and overload threshold for the system to reduce the VNF real-time computing overhead. In order to ensure the validity of the results of the migration estimate, we add the time stamp to the result of the migration calculation, set a reasonable timeout time, and then carry out the migration calculation again after the result set migration scheme timeout expires.

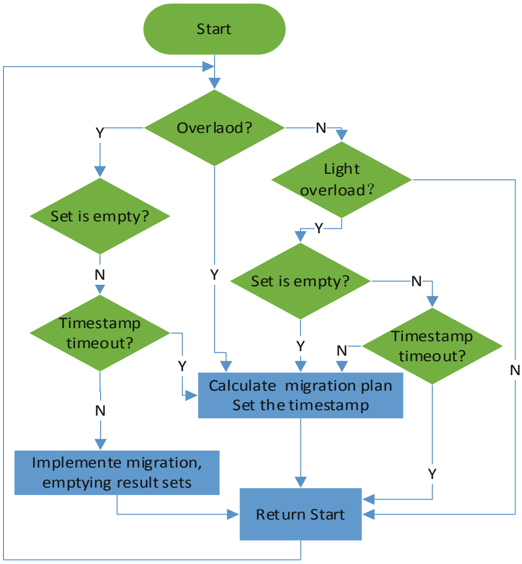

Fig. 1. Cooperation of migration pre-calculation and real-time calculation

As shown in Fig. 1, the implementation procedure of the migration algorithm for migration precomputation and real-time computation collaboration is as follows: Monitoring node resource usage, when the node is overloaded, we check the node's migration scheme result set. If the result set is not empty and the timestamp does not expire, we immediately implement the migration, and empty the node calculation result set, and continue to determine the status of the node. If the node is overloaded, the result set of the node migration scheme is empty, or the timestamp expires, the migration is performed in real time. When the node triggers the light overload threshold but does not trigger the overload threshold, we check the result set of the node migration scheme. If it is null or the timeout expires, we need calculate the node migration scheme again to ensure the validity of the migration scheme.

The pre-calculation and real-time calculation in the system adopt the same migration algorithm. The parameters received by the algorithm include the state of the node, the occupation of each resource by the VNF deployed on the node, and the related state information of other nodes in the system.

4 Heuristic Algorithm for Solving the Problem

The VNFMP problem is an NP-hard problem that can be proven through multiple commodity flows. This section divides VNFMP into two parts: the choice of the VNF to be migrated and the choice of the migrating destination. And two heuristic algorithms are proposed to solve these two problems respectively.

4.1 VNF Selection Algorithm for Migration

Based on experience, the earliest to reach the upper limit of resource types often become the bottleneck to improve performance. Therefore, we need to consider the situation of node overload caused by insufficient resources. We introduce a resource-aware RAIL [11] algorithm to make the decision to migrate the VNF.

When an overload occurs on a node and VNF migration is required, we need to select the VNF to be migrated according to the overload resource type, so as to reduce the resource occupation rate of the overloaded node to a reasonable extent as soon as possible. We define a ternary variable (α, β, γ) to represent the overloaded state of three resources of the node. When computing resources overload, α assignment is 1, otherwise 0, as same as storage resources and communication resources.

We define the migration index θ_i, based on the multi-dimensional environment of perception using RIAL dynamic weight settings. According to the usage of the corresponding resource, the weight of the overloaded resource is greater than 1, the weight of the lightly loaded resource is less than 1, and the greater the migration index of the virtual function used by the overloaded resource is.

$$\theta_i = \sum_K \left\{ \alpha_k \cdot \frac{1}{1 - x_k} + (1 - \alpha_k) \cdot (1 - x_k) \right\} \cdot \chi_{ik} \quad K = (Core, Mem, Com) \tag{6}$$

where α_k indicates whether resource type k is overloaded, x_k represents the occupancy of the resource type k of the node, χ_{ik} indicates the occupancy rate of VNF v_i to node K-type resources.

We sort all the VNF on the overload node according to the migration index, and select the VNF with the highest migration index into the VNF migration sequence.

Algorithm 1: VNF Selection Algorithm For Migration
1. Set VNF f which to be migrated is empty
2. VNF set on the node w_j is F_j
3. Set temporary variable temp=0;
4. for each VNF f_i in F_j
5. calculate θ_i of f_i according to RAIL
6. if θ_i >temp
7. temp= θ_i
8. $f = f_i$
9. end if
10.end for

4.2 Migration Destination Node Selection Algorithm

The selection of migration destination nodes needs to first consider the constraints of VNF migration, including the resource requirements of the VNF to be migrated and the end-to-end performance constraint of the service chain of the function where it is located. In the case of satisfying the constraints, it is necessary to select a node with more types of residual resources and less delay, as a destination node. The choice of the optimal node needs to make the multi-objective decision-making, so the TOPSIS algorithm is used to select the destination node.

The first step is to select the nodes that satisfy the delay constraint according to the end-to-end performance requirement of the service function chain to which the VNF belongs.

$$a_{ij} \cdot b_{ik} \cdot \Delta\tau_{jk+} + \tau_0 < \tau_{\max}, v_i \in f_n \tag{7}$$

In the second step, we select the nodes that meet the performance requirements of the VNF; and the nodes will not overload after the VNF moves in.

In the third step, we use TOPSIS algorithm [12] to calculate the node's immigration index. We need to calculate the positive and negative ideal solutions, and calculate the distance between the positive and negative ideal solutions and get the immigration index of the nodes.

$$T_{i+} = \min\{T_{ki}|k \in N_3\}, i \in \{1, 2, 3\} \tag{8}$$

$$\tau_{+} = \min\{\tau_k|k \in N_3\} \tag{9}$$

$$T_{i-} = \max\{T_{ki}|k \in N_3\}, i \in \{1, 2, 3\} \tag{10}$$

$$\tau_{-} = \max\{\tau_k|k \in N_3\} \tag{11}$$

We find the ideal and negative ideal solutions, and then compute the euclidean distance θ_{ok+} between the node and the positive ideal solution and the euclidean distance θ_{ok+} between the node and the negative ideal solution.

$$\theta_{ok+} = \sqrt{\sum_{i=1}^{3} [\gamma_i(T_{ki} - T_{i+})]^2 + [\gamma_4(\tau_k - \tau_{+})]^2} \tag{12}$$

$$\theta_{ok-} = \sqrt{\sum_{i=1}^{3} [\gamma_i(T_{ki} - T_{i-})]^2 + [\gamma_4(\tau_k - \tau_{-})]^2} \tag{13}$$

where γ_i represents the weight of each indicator, which is a predefined value. So that we can get the immigration index θ_{ok}. The larger the immigration index is, the closer the node is to the ideal solution. We choose to move in the node with the largest index as the destination node.

$$\theta_{ok} = \frac{\theta_{ok-}}{\theta_{ok-} + \theta_{ok+}} \tag{14}$$

Algorithm 2 Migration Destination Node Selection Algorithm

initial
  Destination node v =null
  VNF f_0 to be migrated
  Set of not overloaded nodes N_1
  Set of nodes N_2 =null, N_3 =null
  Set temporary variables temp=0
  Initialize bestNode,worstNode
end initial
for each v_i in N_1
  if $\Delta\tau_i < \tau_0$
    add v_i into N_2
  end if
end for
for each v_i in N_2
  if v_i satisfied the resource requirement of f_0
    add v_i into N_3
  end if
end for
calculate bestNode, worstNode in N_3 according to TOPSIS
for each v_k in N_3
  calculate θ_{ok} of v_k according to TOPSIS
  if θ_{ok} >temp
    temp= θ_{ok}, $v = v_k$
  end if
end for

5 Simulation and Performance Analysis

In this section, we evaluate the performance of the proposed heuristic at different node sizes through simulation experiments and compare it with other algorithms.

5.1 Simulation Settings

The initial size of the network is set to 100 nodes and randomly generates 500 links, which is equivalent to a medium-sized ISP network [13]. Each node is assigned a storage resource, a calculation resource, and a communication resource, each of which has 10 normalized units and a link delay of 1 to 10 random units. The traffic in the network is random, so the corresponding virtual network function is changing dynamically. We assume that each network service function chain consists of 2 to 5 virtual network functions, with 1 to 5 virtual network functions placed on each node. In order to improve the stability of the system, some idle nodes are placed in the network, that is, nodes that do not have the function of virtual network.

5.2 Performance Analysis

All the simulation algorithms are showed in Table 1. The first is RAIL & TOPSIS collaborative resource awareness (RT) algorithm, which can sense multi-domain resources through RAIL algorithm and make multi-objective decision through TOPSIS to find the best migration scheme. However, this algorithm will increase the computational complexity and increase the time spent in migration calculation. The second one is a fixed weight algorithm, which uses a fixed weight to find the VNF to be migrated. The third algorithm is a simple instant migration algorithm. When selecting a migration destination node, the algorithm finds any node which satisfies the performance and delay constraints as the migration destination node. The fourth algorithm is Pre-Calculation RAIL & TOPSIS (PRT) algorithm, which adds pre-calculation and real-time calculation synergy to the RAIL & TOPSIS algorithm to reduce the solution time and reduce the migration overhead.

Table 1. Comparison of algorithm differences

Algorithm	Similarity	Differences
RT algorithm	——	——
Fixed weight algorithm	Using TOPSIS to find dest-node	Using a fixed weight to find the VNF to be migrated
Simple instant algorithm	Using RAIL to find which VNF to be migrated	Choosing any node that satisfies constraints can be the dest-node
PRT algorithm	Using RAIL and TOPSIS algorithm	Adding cooperation of pre-calculation and real-time calculation

Figure 2 shows the total system migration costs under different migration algorithms. It can be seen from the figure that as the node size increases, the total system migration cost also increases. The total migration cost of RT algorithm is higher than other algorithms due to its high computational complexity. With the increase of node size, the migration overhead increases sharply. When the node size exceeds 400, the total migration of Simple Instant Migration algorithm begins to increase dramatically, becoming the algorithm with the highest total cost of migration.

Figure 3 shows the comparison of the number of transitions per unit time under different migration algorithms. It can be seen from the figure that the number of system migration times per unit time increases with the number of nodes. The number of RT algorithm migration is obviously less than the other two algorithms. Compared with RT

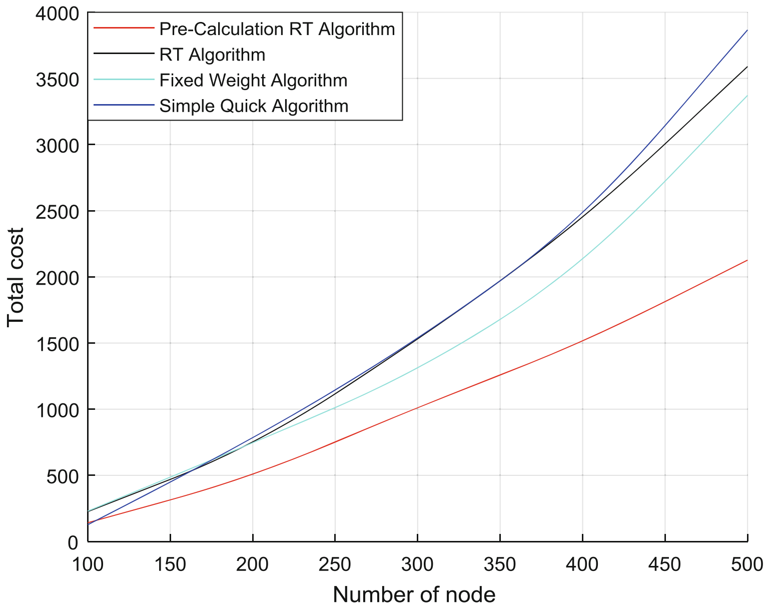

Fig. 2. Comparison of total migration costs

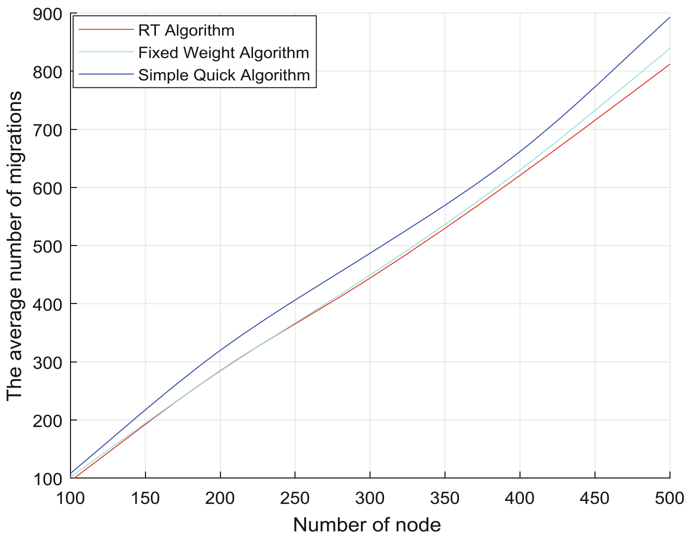

Fig. 3. Comparison of average migration times

algorithm and fixed weight algorithm, it can be concluded that resource awareness through RAIL algorithm can effectively reduce the number of migration. Compared with RT algorithm and simple real-time algorithm, it can be concluded that multi-objective decision making through TOPSIS can effectively reduce the number of migration.

Figure 4 shows the average resource utilization of the destination node when selecting the destination node for the RT algorithm and the simple real-time algorithm. It can be found that the TOPSIS algorithm can achieve load balancing in the network more effectively and make the resource utilization of each node in the network more balanced. Figure 5 shows the single migration cost of VNF under different algorithms. It can be seen from the figure that the cost of single migration is obviously higher than other algorithms due to the large migration cost and fewer migration times of RT algorithm.

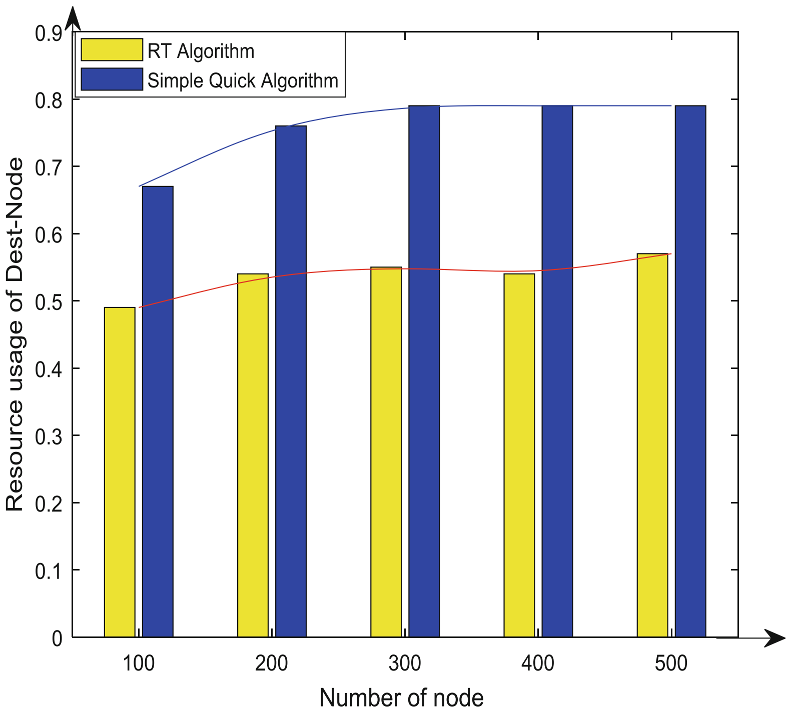

Fig. 4. Resource utilization ratio of destination node

In summary, RT algorithm can effectively reduce the number of migration, and achieve a more balanced network, but it also bring sa higher computational complexity what increases the migration calculation time, resulting in a higher single migration costs. In order to keep the advantages of RT algorithm and reduce the computation time of migration, a PRT algorithm is proposed, which is a combination of pre-calculation and real-time calculation. After adding pre-calculation, it can effectively reduce the migration calculation time, so as to reduce the migration cost. At the same time, the migration cost is almost the same as RT algorithm, which makes the cost of single migration significantly lower than other algorithms.

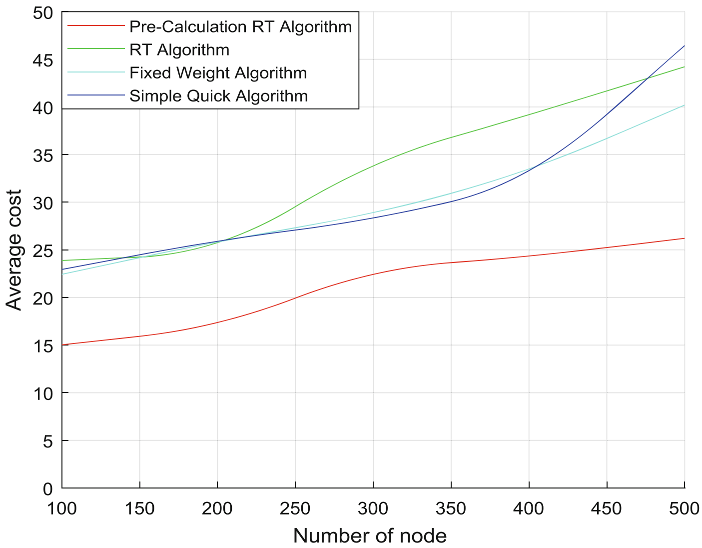

Fig. 5. Comparison of average migration cost per migration

6 Summary

Aiming at the problem of traffic unbalance in NFV network, this paper proposes a VNF-based multi-domain resource-aware migration algorithm based on pre-computation and real-time computation. We choose the VNF to move out by sensing the occupancy of overloaded nodes by the RAIL algorithm. Based on the resource requirements of the VNF and the end-to-end performance constraints of the SFC, TOPSIS is used for multi-objective decision making for better network balance.

The simulation results show that the VNF multi-domain resource sensing migration algorithm, which is based on migration pre calculation and real-time computation, can achieve the best choice of migrating destination nodes, reduce the total migration times and improve the stability of the system. However, with the addition of Multi-objective decision algorithm, the computational complexity is greatly increased, although the migration estimation mechanism can reduce the overall migration calculation cost, but the cost of the single real-time migration computation still cannot be reduced, which needs further research.

References

1. Xiong, G., Hu, Y.X., Tian, L., et al.: A virtual service placement approach based on improved quantum genetic algorithm. Front. Inf. Technol. Electron. Eng. **17**(7), 661–671 (2016)
2. Wen, T., Yu, H., Sun, G., et al.: Network function consolidation in service function chaining orchestration. In: 2016 IEEE International Conference on Communications (ICC), pp. 1–6. IEEE (2016)

3. Faraci, G., Schembra, G.: An analytical model to design and manage a green SDN/NFV CPE node. IEEE Trans. Netw. Serv. Manag. **12**(3), 435–450 (2015)
4. Moens, H., De Turck, F.: VNF-P: a model for efficient placement of virtualized network functions. In: 2014 10th International Conference on Network and Service Management (CNSM), pp. 418–423. IEEE (2014)
5. Ghaznavi, M., Khan, A., Shahriar, N., et al.: Elastic virtual network function placement. In: 2015 IEEE 4th International Conference on Cloud Networking (CloudNet), pp. 255–260. IEEE (2015)
6. Gember-Jacobson, A., Viswanathan, R., Prakash, C., et al.: OpenNF: enabling innovation in network function control. In: ACM Conference on SIGCOMM, pp. 163–174. ACM (2015)
7. Rajagopalan, S., Dan, W., Jamjoom, H., et al.: Split/merge: system support for elastic execution in virtual middleboxes. In: USENIX Conference on Networked Systems Design and Implementation, pp. 227–240 (2013)
8. Sherry, J., Chang, L., Popa, R.A., et al.: BlindBox: deep packet inspection over encrypted traffic. In: ACM Conference on Special Interest Group on Data Communication, pp. 213–226. ACM (2015)
9. Eramo, V., Ammar, M., Lavacca, F.G.: Migration energy aware reconfigurations of virtual network function instances in NFV architectures. IEEE Access **5**, 4927–4938 (2017)
10. Cheng, G.: Based on the meta-capabilities of the network functional combination of key technologies. PLA Information Engineering University (2015)
11. Chen, L., Shen, H., Sapra, K.: RIAL: resource intensity aware load balancing in clouds. In: Proceedings of IEEE, INFOCOM, 2014, pp. 1294–1302. IEEE (2014)
12. Wang, Z.H., Zhan, W., Qiu, W.H.: Application of an optimized entropy-TOPSIS multicriteria decision making model to facilities management. Int. J. Plant Eng. Manag. **11**, 129–136 (2006)
13. Xia, J., Cai, Z., Xu, M.: Optimized virtual network functions migration for NFV. In: 2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS), pp. 340–346. IEEE (2016)