A Reliability-and-Energy-Balanced Service Function Chain Mapping and Migration Method for Internet of Things

SIYA XU1, (Member, IEEE), BOXIAN LIAO1, BO HU2, CONG HAN1, CHAO YANG2, ZHILI WANG1, AND AO XIONG1, (Member, IEEE)

1State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100086, China
2State Grid Liaoning Electric Power Supply Company, Ltd., Liaoning 110000, China

Corresponding author: Ao Xiong (xiongao@bupt.edu.cn)

This work was supported by the Science and Technology Project of State Grid Corporation of China: Research and Application of Key Technologies in Virtual Operation of Information and Communication Resources.

ABSTRACT With the rapid development of Internet of Things (IoT) technology, Network Function Virtualization (NFV) is introduced in the edge network to provide flexible and personalized service. However, there still exist some problems to be solved, such as high cost, unbalanced load, and low availability. Therefore, a reliability-and-energy-balanced Service Function Chain (SFC) mapping and migration method is presented for IoT applications. First, aiming at improving network performance and reducing expenditure, an SFC mapping algorithm based on cost optimization, load balancing, and reliability is proposed to map SFC requests onto the network and provide backup. Second, aiming at optimizing resource configuration, an SFC migration method based on energy consumption and quality of service is proposed to integrate network resources. Simulation results show that the proposed method outperforms the compared algorithms by 15.5% and 24.55% in the acceptance ratio of SFC requests and the overall costs, respectively.

INDEX TERMS Edge networks, IoT, mapping, migration, NFV, SFC.

I. INTRODUCTION

With the fast development of smart city and 5G, the Internet of Things (IoT) has become a hot technology that can connect a wide variety of devices into the network, such as portable devices, cameras, and sensors. The amount of mobile data generated by IoT devices is growing exponentially, and the processing requirement of these services is becoming more differentiated. However, the tight coupling of hardware and software in the traditional communication network may cause management difficulties, high operation costs, and poor flexibility [1], making it difficult to meet the service requirements of IoT applications. Hence, Internet Service Providers (ISPs) usually adopt Network Function Virtualization (NFV) technology to decouple function and hardware so that Virtual Network Function (VNF) can be treated like software and be instantiated and removed rapidly. In a virtualized network, the service is regarded as a Service Function Chain (SFC) that steers traffic through an ordered sequence of VNFs.

NFV provides service by mapping the SFC to the physical network. In this way, the communicational and computational capabilities of the network can be shared among IoT services.

Besides, to shorten the distance of data transmission and relieve pressure of the core network, edge computing is applied in the IoT network. By integrating NFV and edge computing technology, ISPs can provide IoT applications with various services such as data cleaning, data processing, data aggregation, and protocol dialogue in the edge network. In addition, it is necessary to provide abnormal traffic and attack detection on the gateway to avoid leakage of private data or system attack from outside. Note that, the resource of edge network is limited. Thus, we need to reasonably embed SFCs into the abstract network, so as to better utilize the edge resource and guarantee service quality.

The SFC mapping mechanism has become a hot topic in recent years. When designing the mapping algorithm of SFC, many researchers aim at optimizing the cost of deployment, the reliability and end-to-end delay of service chain, the acceptance rate of service request, the rationality and effectiveness of resource allocation and other factors. Particularly,
an IoT device and a Docker-based light-weight virtualization functional architecture has been proposed [2], and several SFC mapping algorithms are designed for IoT in [3]–[6]. Fu [3] proposed an SFC mapping scheme based on deep reinforcement learning. Nguyen [4] studied the resource consumption optimization problem of the service chain deployment in the cloud-side collaborative network under the IoT scenario. He [5] studied the optimal placement of VNFs with multiple instances. Sarrigiannis [6] proposed an IoT architecture based on edge and cloud computing collaboration. To sum up, some useful solutions are provided in SFC embedding and edge-cloud computing collaboration, but they cannot concurrently reduce costs and balance the load. Moreover, none of them considers the efficiency of backup.

To address the above problems, this paper proposes a reliability-and-energy-balanced service function chain mapping and migration method (REB-SFCM) to provide service economically and reliably. It divides the SFC mapping process into mapping stage and migration stage. In mapping stage, it uses the Measure of Importance (MOI) to balance the cost and load, and provides multiple backup modes for IoT applications to improve the availability of service and the efficiency of backup; In migration stage, we take energy and Quality of Service (QoS) into consideration, and adjust migration solution in accordance with the changing traffic to further reduce the cost. The main contributions of this paper are listed as follows:

1) Design the Measure of Importance (MOI) as a critical factor in mapping scheme to balance the cost and load, and propose a multi-mode backup mechanism to ensure service availability, including link-backup and VNF-backup mode. In this way, when a node fails, it can choose an appropriate backup mode to avoid service interruption.

2) Design an energy-and-QoS-balanced migration solution that is adjusted with changing traffic to reduce energy consumption and avoid revenue loss of service. Considering energy is related to the utilization, we propose a comprehensive evaluation method for a multi-resource server to quantify the utilization of the device.

3) Propose a decision tree generating process and improve the Upper Confidence Bound Apply to Tree (UCT) to make a compromise between expansion and exploration of sub-branch. Meanwhile, we introduce the delay judgment and design a feedback mechanism to select the valuable child-node.

The rest of this paper is organized as follows. Section 2 discusses the related works. Section 3 constructs the system model. Section 4 describes the design of algorithms. Section 5 gives the simulation and evaluation of the algorithm. Finally, section 6 concludes the paper.

II. RELATED WORKS

In this paper, we focus on three important stages in resource allocation, which are the SFC mapping stage, reliability improvement stage, and migration stage.

A. SFC MAPPING

In the mapping stage, the cost, end-to-end delay, and the acceptance rate of service request are usually the key decision factors. Fu [3] proposed an SFC mapping scheme for IoT, but it neither specifies the specific application of the IoT nor describes the dynamic characteristics of the network resources. Beck [7] focused on improving the utilization of nodes and links. However, the utterly reliable device that they assumed is unrealistic in the physical world, and this could cause low reliability of SFC. Some researchers [8], [9] adopted the heuristic algorithms to deal with the Non-deterministic Polynomial (NP) model, while their model only considered part of the cost, availability, and load. Aiming at reducing cost in dynamic scenarios, Liu [10] and Zhu [11] proposed the Integer Linear Programming (ILP) model, yet it is more applicable to small-scale networks. Yu [12] designed a dynamic traffic model of the multicast service and focused on maximizing energy efficiency, but they ignored the load. In summary, the existing works cannot concurrently reduce costs and balance the load. Meanwhile, some papers ignored reliability.

B. RELIABILITY IMPROVING

While, in the reliability improvement stage, the trade-off between reliability and resource utilization is always taken into consideration but hard to realize. A backup for the VNF with the lowest reliability in SFC was proposed by Kong [13]. However, they did not consider the sharing between backup VNFs and caused the redundant occupation of the same resources. Aimed at improving the availability of SFC, Qu [14] designed a coordinated protection mechanism that did not consider the cost of backup. Besides, Nguyen [4] considered the delay and bandwidth consumption of the underlying IoT network, as well as the dynamic characteristics of network traffic. He also explained the specific types of IoT devices and how the IoT devices and the SFC collaborate to complete services. He [5] designed a distributed multi-instance VNF placement algorithm to minimize cost and delay, as well as balance the network load. Barrère [15] selected the backup with the most significant improvement in reliability and only considered preparing backups for a single VNF. As a result, this may lead to service interruption when the VNF having no backup fails. In addition, some researchers [16], [17] adopted heuristic algorithms to maintain reliability. But, same as previous researches, neither of them mentioned backup efficiency.

C. SFC MIGRATION

In the migration stage, the changes in underlying resources are difficult to analyze and always be ignored. Ibn-Khedher [18] proposed an ILP formulation and evaluated it in a small-scale scenario. Xia [19] proposed a heuristic algorithm to get the feasible scheme in polynomial-time and improve the performance in terms of time, cost, and load, but they did not take degradation of service into account.
Sarrigiannis [6] proposed a 5G IoT architecture based on the collaboration of edge computing and cloud computing, and designed an online VNF full life cycle algorithm. Eramo [20] considered revenue loss caused by QoS degradation in the migration process, but they neglected the change of weight. Zhang [21] aimed at building a more agile and flexible network and proposed related migration strategies. Carpio [22] proposed a scheme to deal with underutilization, but they ignored that multiple resources could exist in one node. Zhou [23] focused on replication and reduced migration frequency. Besides, the current researches in migration have seldom adjusted migration mechanism with the changing traffic.

In conclusion, the existing methods still have the following shortages: cannot optimize costs and load simultaneously while SFC mapping; lack efficient reliability-assure mechanism; cannot dynamically generate migration solution to fit traffic trend. Therefore, this paper proposes a two-step SFC placement process to dismiss the above disadvantages. First, we propose an SFC mapping method based on cost, load, and reliability to provide resources for service requests. Second, we present an energy-QoS-based migration method combined with the traffic model to change the migration strategy dynamically.

III. MODEL CONSTRUCTION

A. NETWORK MODEL

Internet of things (IoT) is a kind of transmission network that can intelligently identify information. It can connect any Internet item according to the contract through radio frequency identification device, infrared sensor, two-dimensional code reading device, laser scanner, global positioning system and other sensing equipment. The IoT devices send raw data or preprocessed data to the edge server and receive command from the edge layer. The virtualization mechanism realizes on-demand service by allocating the required resources to IoT services, making the network scalable. For latency-sensitive IoT applications (such as fast face authentication) or regional autonomy service (such as microgrid), the amount of data is large, and the service needs to be processed timely. So, we should map these SFCs to the edge platform; While, for latency-insensitive IoT applications (such as user behavior analysis), we can map these SFCs to the cloud platform.

Fig. 1 shows the three-layer network architecture and is expanded from left to right. IoT sensors are on the far left, like thermometers and pressure gauges. The middle is the edge network layer includes the gateway and edge data centre (DC). The far-right is the core network layer. The integrated network orchestrator is on the top of Fig. 1, and it can control the resource allocation process through the cloud/SDN controller (Ctl). NFV provides end-to-end network services in the form of a service chain. The service chain defines a specific sequence of VNF sets, and users’ traffic needs to pass through these network function nodes in order. Each VNF in the service chain is mapped to different server nodes in the underlying network, and the virtual links connecting each VNF are mapped to the physical links connecting each edge server node.

B. VARIABLES

The variables used are listed in Tab. 1 and Tab. 2.

TABLE 1. Physical network variables.

| Symbol | Definition |
|---|---|
| $V_\text{n}$ | Network node |
| $G(V,L)$ | Network topology, where $V$ is the set of nodes and $L$ is the set of links |
| $P_{\text{id}}$ | Idle power of node $V_\text{n}$ |
| $P_\text{w}$ | Working power of node $V_\text{n}$ |
| $\text{Cap}_\text{x}$ | Processing capacity of node $V_\text{n}$ |
| $\text{Cap}_s$ | Storage capacity of node $V_\text{n}$ |
| $F_\text{b}$ | Binary variable. 1 represents node $V_\text{n}$ is in use, otherwise 0 |
| $\mu_\text{r}$ | Resource utilization of node $V_\text{n}$ |
| $B_{\text{xy}}$ | Bandwidth of link $(x,y) \in L$ |

C. MAPPING MODEL

The mapping model includes the SFC request mapping model and backup model. The detailed description of the two models is given as follows.

1) SFC MAPPING AND BACKUP MODEL

a: SFC MAPPING MODEL

Aiming at balancing the load, reducing cost, and improving the reliability of service, we design the Measure of Importance (MOI) as a critical factor to select the mapping scheme:

$$MOI = \chi \cdot \text{Cost}_{s_i(l_i)} + \frac{\varphi}{b_{l_i}} \quad (1)$$

where $\text{Cost}_{s_i(l_i)}$ and $b_{l_i}$ are the cost and load factors of the mapping scheme $l_i$ of $s_i$, $\chi$ and $\varphi$ are weighting factors.
TABLE 2. Virtual network variables.

| Symbol | Definition |
|--------|------------|
| $V$   | $(f_1, f_2, \ldots, f_n)$ Set of all types of VNF. All VNFs are unique |
| $G_r(V_r, L_r)$ | Virtual network topology, where $V_r$ and $L_r$ are the set of virtual vertices and links respectively |
| $S = (s_1, s_2, \ldots, s_n)$ | Set of SFC requests |
| $b_i$ | Bandwidth requirement of $s_i$ |
| $\text{cap}^{p}_{f_m}$ | Processing capacity requirement of $f_m$ |
| $\text{cap}^{m}_{f_m}$ | Storage capacity requirement of $f_m$ |
| $\text{Scheme}_{s_f}$ | VNF $f$’s migration scheme in SFC $s_i$ |
| $F_{s_f} = (f_1, f_2, \ldots, f_n)$ | VNF type and order requested by SFC |
| $x^{v_f_{s_f}/v_i}_{v_i}$ | Binary variable 1 represents $s_i$ from VNF $f_m$ on node $v_i$, to VNF $f_m$ on node $v_{v_f}$, otherwise 0 |
| $x^{v_f_{s_f}}_{v_i}$ | Binary variable 1 represents VNF $f_m$ of $s_i$ on node $v_i$, otherwise 0 |
| $x^{v_f_{s_f}/s_f}_{v_i}$ | Binary variable 1 represents in $s_f$, the next function of VNF $f_m$ is $f_m$ |

**FIGURE 2.** Backup switching model.

$b$: SFC BACKUP MODEL

This model uses SFC backup and VNF backup to improve the reliability of SFC whose requirement is not met. Note that link backup has a higher priority. The link with the highest COST-EFFICIENCY ($CE_b$) will be selected as the link backup, and the expression is:

$$CE_b = \alpha \cdot (1 - r_{s_f(l)}) \cdot (1 - r_{s_f(l_m)}) - r_{s_f(l_m)} + \beta \cdot \eta \cdot \text{Cost}_{s_f(l)}$$  (2)

where $l_m$ is SFC mapping link; $\eta$ is the utilization of link; $\Delta r_{s_f(l)}$ is the increment of reliability of SFC improved by the backup link $l$, $\beta \approx \gamma$ and $\gamma$ are weighting factors. $r_{s_f(l)}$ is the reliability of the backup link.

If the selected link backup cannot meet the SFC’s requirement, this model would choose the VNF with highest $CE_{v_f}$ as backup VNF:

$$CE_{v_f} = \frac{\lambda \cdot (r_{s_f(\text{backup})} - r_{s_f(l)})}{\mu \cdot (p_{s_f(v_m)} + c_{s_f(v_m)})}$$  (3)

where $p_{s_f(v_m)}$ and $c_{s_f(v_m)}$ represent the processing and storage capacity of the node occupied by $s_f$. $\lambda$, $\mu$, and $\eta$ are weighting factors.

Fig.2 shows the backup switching model. If one VNF fails, we will switch to its backup by order of VNF backup, and then, link backup. Fig.3 shows the sharing mechanism of them.

**FIGURE 3.** Backup sharing mechanism.

2) FACTOR DESIGN

**a: RELIABILITY**

The reliability value $r_i$ of node $v_i$ determines whether backup is required. If the product of node $v_r$ on SFC is greater than the reliability requirement of service chain request, backup is not selected; If it is less than requirement, backup is selected. The reliability $r_i$ of VNF $f_i$ is calculated as:

$$r_i = \frac{MTBF}{MTBF + MTTR}$$  (4)

MTBF represents the mean time between failure of a device; MTTR represents the mean time of repairing the device.

**b: COST**

The cost of the mapping scheme includes node and link costs:

$$Cost = \sum_{s_i} \left( \sum_{m=1}^{v_f} (p_{s_f(v_m)} + c_{s_f(v_m)}) + B_i \cdot (i - 1) \right)$$  (5)

The link load $l_m$ is designed for mapping scheme and represents the load status of the mapping link. It is calculated as follow:

$$b_{l_m} = \sigma \cdot \sum_{m=1}^{v_f} (p_{s_f(v_m)} + c_{s_f(v_m)}) + \frac{1}{\zeta \cdot \text{total}_{s_f}}$$
a low utilization rate, which can be indicated by the green

In Fig. 4,a, the two servers (3, 4) where FW is hosted have consumption in Fig. 4,a and Fig. 4,b. Proper migration schemes for the two SFCs to reduce energy consumption, we migrate some VNFs of SFC1 consists of LB, VFW, IPS, and SFC2 that energy is the quadratic function of utilization (detailed in migrating model). The other is that the server with high utilization may fail to provide enough resources for new service demands and reduce energy consumption, we migrate some VNFs of SFC to other suitable edge nodes (ENs). VNF migration will be triggered in one of the typical migration scenarios shown in Fig. 4.a and Fig. 4.b.

Migration triggered with low utilization of ENs. a. Migration triggered with high utilization of ENs. c. Two types of SFC.

FIGURE 4. a.

where $\rho_{\text{resi}}(\nu_m)$ and $c_{\text{resi}}(\nu_m)$ are remaining processing and storage capacity of node $\nu_m$, and $l_{\text{ox}}(\text{resi}_{\nu_m})$ are the remaining ingress and egress bandwidth of $\nu_m$. $\nu_n$ is the total number of network nodes. $\zeta$, $\delta$, $\sigma$ are weighting factors.

The network load $b$ represents the load status of the overall network:

$$
\begin{align*}
b &= \omega \cdot \sum_{m=1}^{\nu_n} \left( \min \left\{ \frac{l_{\text{in}}(\text{resi}_{\nu_m})}{l_{\text{in}}(\text{resi}_{\nu_m})}, \frac{l_{\text{in}}(\text{resi}_{\nu_m})}{l_{\text{in}}(\text{resi}_{\nu_m})} \right\} \right) \cdot \left( l_{\text{ox}}(\text{resi}_{\nu_m}) + l_{\text{ox}}(\text{resi}_{\nu_m}) \right)
\end{align*}
$$

$d$: DELAY

The end-to-end delay of SFC $s_i$ can be expressed as:

$$
\begin{align*}
d_{\text{end-end}}(s_i) &= \sum_{m=1}^{i} d_{\text{proc}}(\nu_m) + \sum_{h=1, j=h+1}^{i-1} d_{\text{trans}}(h,j)
\end{align*}
$$

where $i$ is the total number of nodes in the SFC mapping link. $d_{\text{proc}}(\nu_m)$ an $d_{\text{trans}}(h,j)$ are processing and transmitting delay, respectively.

D. MIGRATION MODEL

1) MIGRATION SCHEME

We assume that the server can host various types of VNFs such as Load Balancing (LB), Virtual Firewall (VFW), Intrusion Prevention System (IPS), and Deep Packet Inspection (DPI). The VNFs on one server can be shared by multiple SFCs. Usually, the services in the network are always changing dynamically. To adapt to time-varying service demands and reduce energy consumption, we migrate some VNFs of SFC to other suitable edge nodes (ENs). VNF migration will be triggered in one of the typical migration scenarios shown in Fig. 4.a and Fig. 4.b.

Fig. 4.c shows that two SFCs are providing services in the network. SFC1 consists of LB, VFW, IPS, and SFC2 includes VFW and IPS. In the next step, we will choose proper migration schemes for the two SFCs to reduce energy consumption in Fig. 4.a and Fig. 4.b.

In Fig. 4.a, the two servers (3, 4) where FW is hosted have a low utilization rate, which can be indicated by the green circle. However, even if the utilization of two working servers are both low, their energy consumption is still more than a moderately utilized server. So we transfer the traffic flow from one server to the other, then shut down the idle one. In Fig. 4.a, if the traffic through FW of the two servers is merged into one, such as 3, the utilization of the working server will be increased. In Fig. 4.b, the orange circle means that the utilization is too large (for example, server 1). There are two reasons for reducing the utilization of server: one is that energy is the quadratic function of utilization (detailed in migrating model). The other is that the server with high utilization may fail to provide enough resources for new service, causing a low acceptance ratio of service in the overall network. To solve the above problems, this paper proposes two strategies: one is to migrate all the LB related SFCs to another server with a smaller load, as shown in 2. The other is to migrate part of the SFCs to the target server, as shown in 3 because it is also important to guarantee QoS. Furthermore, when and whether to execute the migration strategy is subject to the migrating model (detailed in migrating mode).
2) TRAFFIC MODEL

Although the behavior and preference of users are different, the traffic still shows a distinct periodic pattern [19]. Based on traffic periodicity, this model uses \( N \) and \( \Delta n \) to represent the length of the period and the time interval. The traffic phase and the traffic status are defined as \( n \) and \( T_n \). Additionally, the value of phase \( n \) is between \([0, N)\), and the difference between each phase’s value is \( \Delta n \). Typically, the period length \( N \) is defined as 24 hours, and the interval \( \Delta n \) is set to 1 hour. Based on the above settings, the traffic is divided into 24 phases, namely \( n = (0, 1, 2, \ldots, 23) \).

3) MIGRATING MODEL

SFC migration can reduce energy consumption at the cost of QoS degradation, but the degradation will lose revenue of service. To reduce energy consumption and maintain revenue, the objective function should be:

\[
\min \{ \sum_{n \mod N \in [0, N]} C_{\text{EQ}, n \mod N} \} \tag{9}
\]

where \( C_{\text{EQ}, n \mod N} \) is the energy-and-QoS (EQ) cost in the traffic phase \( n \). As traffic phase \( n \) changes, it will dynamically adjust the weights of energy cost and revenue loss. \( C_{\text{EQ}, n} \) is expressed as:

\[
C_{\text{EQ}, n} = w_{\text{ene}, n} \cdot C_{\text{energy}} + w_{\text{lo}, n} \cdot P_{\text{loss}} \tag{10}
\]

where \( C_{\text{energy}} \) and \( P_{\text{loss}} \) are energy cost and the revenue loss, \( w_{\text{ene}, n} \) and \( w_{\text{lo}, n} \) are their weights, respectively. Moreover, \( w_{\text{ene}} + w_{\text{lo}} = 1 \) means the two weights are inversely related. In the peak of traffic \( (n = 0) \), this model sets a higher weight of revenue loss to lessen the impact of energy consumption on the migration strategy, which can avoid frequent migration. Similarly, in the trough of traffic \( (n = (1 + N)/2) \), for the sake of effectively reducing energy cost, this model sets a higher weight \( w_{\text{ene}} \) for energy consumption to adjust resource configuration and shut down the idle server quickly:

\[
w_{\text{ene}} = w_{\text{cri}} - |n - (1 + N)/2| \cdot w_{\text{INT}} \tag{11}
\]

\( w_{\text{cri}} \) is the basis weight and \( w_{\text{INT}} \) is the correction weight. These two weights are both constant, and the values of them can be customized according to the traffic phase. This paper sets \( w_{\text{cri}} \) and \( w_{\text{INT}} \) to 0.5 and 0.05.

Assume that power is linear with the utilization of a server. At time \( t \), the online power of the server \( v \) is:

\[
P_v(t) = P_{\text{idle}}(t) + (P_{\text{max}}(t) - P_{\text{idle}}(t)) \cdot u_v(t) \tag{12}
\]

where \( P_{\text{max}}(t) \) and \( P_{\text{idle}}(t) \) are the maximum power and idle power of a server, respectively. The utilization \( u_v(t) \) can be expressed as:

\[
u_v(t) = \frac{\sum_{s \in S} x_{v_{s}} \cdot \text{cap}_{v_{s}, u_{s}}}{\text{Cap}_{v_{s}}^p} + \frac{\sum_{s \in S} x_{v_{s}} \cdot \text{cap}_{v_{s}}^m}{\text{Cap}_{v_{s}}^m} \tag{13}
\]

where \( w_p \) and \( w_m \) represent weights of the utilization of processing and storage capacity, and \( w_p + w_m = 1 \). \( \text{Cap}_{v_{s}}^p \) is the processing capacity, and \( \text{Cap}_{v_{s}}^m \) is the storage capacity.

Equation (13) indicates that the utilization of server \( v \) is affected by processing and storage capacity. The measure of utilization must take into account all the usage status of multiple resources within the server. Therefore, the weights of different resources need to be set separately, which is especially critical when the utilization of resources differs widely. Therefore, \( w_p \) is defined as follow:

\[
w_p = \frac{w_{p-bas}}{w_{p-bas} + w_{m-bas}} \cdot 100\% \tag{14}
\]

\[
w_{p-bas} = \begin{cases} w_{p0} + |w_{p0} - w_{p1}| \cdot w_{p1}, & w_{p0} \geq w_{p1} \\ w_{p0} - |w_{p0} - w_{p1}| \cdot w_{p1}, & w_{p0} < w_{p1} \end{cases} \tag{15}
\]

where \( w_{p0} \) is defined as follow:

\[
w_{p0} = \frac{1}{n} \sum_{1}^{n} \frac{1}{u} \tag{16}
\]

where \( u \) is the utilization rate, \( u_p \) is the processing utilization rate, and \( u_m \) is the storage utilization rate. And \( \sum_{1}^{n} \frac{1}{u} = \frac{1}{u_p} + \frac{1}{u_m} \).

The correction factor \( w_{p1} \) is:

\[
w_{p1} = \frac{\text{Cap}_{v_{s}}^p}{\text{Cap}_{v_{s}}^p + \text{Cap}_{v_{s}}^m} \tag{17}
\]

\( w_{p-bas} \) can be got in the same way.

The high utilization of a server often leads to an unnecessary cooling process of the device, causing extra energy consumption. In severe cases, it makes the server to be a “dead node” that cannot undertake new tasks. The energy consumption of the server in the time slot \([0, t]\) is:

\[
C_{\text{energy}} = E \cdot \int_{0}^{t} (1 + w_n \cdot u_n(t)) \cdot P_v(t) \sum_{v \in V} F_v dt \tag{18}
\]

where \( E \) is the cost per consumed power watt, \( F_v \) is the binary variable: 1 represents node \( v \) is in use, otherwise 0. \( u_n(t) \) is the utilization rate of node \( v \) at time \( t \).

Energy consumption is mainly determined by the utilization and power of the server. Thus, we can set \( w_n \) to adjust the impact of utilization on energy consumption in different traffic phases. \( w_n \) is calculated as follow:

\[
w_n = \begin{cases} 1 - \beta \cdot n, & n \in \{0, (N + 1)/2, 1 + \beta \cdot (n - N - 1)\} \\ 1 + \beta \cdot (n - N - 1), & (N + 1)/2 < n \leq N \end{cases} \tag{19}
\]

Here we set \( \beta = 0.05 \). Besides energy consumption, migration cost also includes the loss of service revenue while migrating. If the migration process cannot be finished in the permitted time, the degradation of service will occur, and this will cause the downtime of service \( \Delta t \). The revenue loss can
be calculated as:

$$Pro_{loss} = \sum_{s_i \in S} pro_{s_i} \cdot |(\Delta t) \cdot X_{s_i,f_m}| \cdot \Delta t > 0$$  \hspace{1cm} (20)$$

where $pro_{s_i}$ is Loss of profit per unit time of service $s_i$, $X_{s_i,f_m}$ indicates whether the $f_m$ function of $s_i$ is temporarily suspended due to migration. 1 represents terminate, otherwise 0.

### E. CONSTRAINTS

**Constraint (1) (Capacity Constraint):** For any node in the mapping scheme, its remaining capacity should be higher than service needs:

$$p_{res(x)} \geq p_{s_i(x)}; \quad \forall x \in l_i$$  \hspace{1cm} (21)$$

$$cap_{res(x)} \geq cap_{s_i(x)}; \quad \forall x \in l_i$$  \hspace{1cm} (22)$$

**Constraint (2) (Bandwidth Constraint):** For any link in the mapping scheme, its remaining bandwidth should be higher than service needs:

$$b_{res(n_i,n_{i+1})} \geq B_i; \quad \forall n_i, n_{i+1} \in l_i$$  \hspace{1cm} (23)$$

**Constraint (3) (Order Constraint):** The order of traffic through VNFs should be consistent with the required order of the SFC request:

$$l_i \left(n_{i_1} \ldots n_{i_2} \ldots n_{i_l}\right) \equiv SFC\text{-}Order$$  \hspace{1cm} (24)$$

where $n_{i_j}$ represents VNF $f_i$ on the node $n_i$.

**Constraint (4) (Delay Constraint):** The end-to-end delay should be smaller than the maximum tolerance delay of service:

$$d_{end-\text{end}}(l_i) < D_{s_i}$$  \hspace{1cm} (25)$$

**Constraint (5) (Traffic Conservation Constraint):** Two VNF D0 and D1 are added to each SFC request, and the SFC should start from D0 to D1:

$$\sum_{v_1 \in V, f_m \in F_{s_i}} x_{s_i}^{f_{m_1}v_1f_{m_2}} - \sum_{v_2 \in V, f_m \in F_{s_i}} x_{s_i}^{f_{m_2}v_1f_{m_1}} = \begin{cases} 1 & \text{if } f_{m_1} = D_0 \\ -1 & \text{if } f_{m_1} = D_1 \\ 0 & \text{otherwise,} \end{cases}$$  \hspace{1cm} (26)$$

**Constraint (6) (Mapping Constraint):** A function of an SFC request is embedded only once (there is no one function assigned to two different nodes):

$$x_{s_i}^{v_{f_{m_1}}v_{f_{m_2}}} = 0 \quad \forall S_i \in S, f_m \in F_{s_i}, v_1 \in V, v_2 \in V, v_1 \neq v_2$$  \hspace{1cm} (27)$$

**Constraint (7) (VNF Traffic Constraint):** The traffic through a VNF must be less than the throughput preset by the node for the VNF:

$$TP_{f_m} > \sum_{s_i} b_{s_i} \cdot x_{s_i}^{v_{f_{m_1}}} \quad \forall v \in V, s_i \in S, f_m \in s_i$$  \hspace{1cm} (28)$$

**Constraint (8) (Cross Constraint):** Backup and mapping schemes should not have replicated nodes:

$$x_{s_i,n\in F_{s_i}} \neq 1$$  \hspace{1cm} (29)$$

**Constraint (9) (Number Constraint):** The backup sharing limit is:

$$0 \leq backup_k \leq 2$$  \hspace{1cm} (30)$$

**Constraint (10) (Capacity Constraint-Node):** There must be no useless node, that is:

$$Cap_v > cap_{s_i,f_m}; \quad \forall v \in V, s_i \in S, f_m \in F_{s_i}$$  \hspace{1cm} (31)$$

$$Cap_v > cap_{s_i,f_m}; \quad \forall v \in V, s_i \in S, f_m \in F_{s_i}$$  \hspace{1cm} (32)$$

**Constraint (11) (Capacity Constraint-Link):** There must be no useless link, that is:

$$B_{s_i,v_1,v_2} > b_{s_i}, \quad \forall v_1, v_2 \in V, s_i \in S$$  \hspace{1cm} (33)$$

**Constraint (12) (QoS Constraint):** The delay of the migration path for each SFC must not exceed its maximum tolerance delay:

$$d_{s_i,SFCM-CBR} < d_{tol(s_i)}$$  \hspace{1cm} (34)$$

**Constraint (13) (Capacity Constraint):** The bandwidth constraint if migrating is:

$$B_{s_i,v_1,v_2} > b_{s_i}, \quad \forall (v_1, v_2) \in \text{Scheme}_{s_i,f}$$  \hspace{1cm} (35)$$

In a word, the optimization problem model can be derived as:

$$\max \left\{ \frac{\psi \cdot b}{\text{Cost}} \right\} \quad \text{s.t. : } C_1 - C_{13}$$  \hspace{1cm} (36)$$

where Cost is the total cost of all SFC mapping schemes; $b$ is the load of the network; $\psi$ is the weight.

### IV. ALGORITHM DESIGN

The REB-SFCM contains two sub-algorithms, which are applied in the mapping and migration process, respectively.

#### A. SFC MAPPING ALGORITHM

In this section, we propose a two-stage SFC mapping algorithm based on cost optimization, load balancing, and reliability (SFCM-CBR), which includes SFC mapping and backup.

The steps of SFCM-CBR are described as follows. First, it gets the arrived SFC requests and uses (38) to calculate the mapping range $m_i$ of each SFC, which can mark the candidate node-set. Then, an improved Dijkstra algorithm is designed to manage the node-set and generate a candidate set for the SFC mapping scheme. Next, the scheme with the smallest MOI is selected as the SFC mapping scheme. Afterward, SFCM-CBR calculates the reliability of this scheme. If the scheme does not meet the reliability requirements of SFC, it will provide a backup for the SFC. The backup selection needs to follow the following conditions: 1) has the largest cost-efficiency; 2) has no same node with the SFC mapping
scheme. Finally, it rechecks the updated reliability of SFC to judge whether it meets the requirement. If it does, the backup process will be finished. Otherwise, SFCM-CBR will prepare another backup. If there is no candidate link backup, we will choose VNF backup. The $O(n^2 \log n)$ algorithm process is detailed in Alg. 1.

Mapping range can be derived from (8):

$$m_i = \frac{D_{s_i}}{d_{s_i(s)}}$$  \hspace{1cm} (37)

where $D_{s_i}$ is the maximum tolerated delay of $s_i$.

**Algorithm 1** SFC Mapping Algorithm Based on Cost Optimization, Load Balancing, and Reliability (SFCM-CBR)

**Input:** physical network $G$, SFC requests $s$  
**Output:** SFC mapping scheme $I_{s_i} \forall s_i \in S$  
1: network initialization $T_{s_i} = \emptyset$, $Rel_{s_i} = 0$  
2: accept the service request  
3: for $s_i \forall s_i \in S$ do  
4: calculate $m_i$, $\alpha$, mark unavailable node or link  
5: generate candidate node-set $T_{s_i}$  
6: while $C_{s_i} = \emptyset$ do  
7: Dijkstra prune $(m_i, \alpha)$  
8: $\alpha = \alpha + 1$  
9: end while  
10: select $s_i$’s mapping scheme  
11: update network  
12: check whether $r_{s_i}$ satisfied its requirement, if yes jump to step 20  
13: while $B_{s_i} \neq \emptyset$ do  
14: select link backup  
15: end while  
16: while $r_{s_i} < R_{s_i}$ do  
17: select VNF backup  
18: end while  
19: end while  
20: end for

**B. SFC MIGRATION ALGORITHM**

The energy-QoS-based SFC migration method proposed in this paper is based on the Monte Carlo Tree Search Strategy (MCTS). The main advantage of MCTS is that it builds the tree quickly and has a feedback mechanism based on the search results, and this would mark the decision tree with status. At present, MCTS is mainly applied in the field of game theory, such as computer games, to realize the prediction of the selection of the next round. Based on this strategy, we propose an SFC migration method based on energy-cost and QoS aware (SFCM-EQA) with a feedback mechanism.

SFCM-EQA gradually builds a decision tree through a four-stage process and stops the building process when the number of iterations reaches the limit, then selects the optimal scheme in the decision tree. The algorithm is mainly divided into the following steps: 1) initialize node of tree; 2) select child-node to explore or expand based on UCT and delay; 3) operation step; 4) update structure and status of the tree; 5) select the optimal solution.

1) **INITIALIZATION**

Map the VNF to be migrated to its corresponding server and treat the server as the root of the decision tree.

2) **NODE SELECTION**

Selection is an essential part of building the decision tree. With the guide of UCT and delay, we select a child-node of root for generating a sub-branch. By making a compromise between exploration and expansion, SFCM-EQA realizes the balance of the nodes with the lowest EQ cost and the nodes which have not been fully explored. In this way, the decision tree can find a lower energy consumption migration scheme while avoiding local optimization. At the same time, we introduce the delay to the judgment. The prior probability is calculated by the success and failure frequency of the delay-based judgment in (40). SFCM-EQA tends to select the node whose overall delay meets the requirements to increase the number of successful schemes in the final decision tree.

To achieve the above objectives, we define the factor $UCT$ to represent the importance degree of each node $v_i \in V$ in the decision tree:

$$UCT_{v_i} = \frac{N_{v_i}}{C_{EQ,v_i} + \Delta} + q \cdot \frac{\log_2 N_{v_i} + \beta \cdot \frac{N_{v_i}}{C_{EQ,v_i} + \Delta}}{N_{v_i} + \alpha \cdot p_{v_i} + \Delta} \hspace{1cm} (38)$$

$N_{v_i}$ and $N_{v_i}$ represent the visiting frequency of node $v$ and $v_i$ currently, respectively. The left part is the expansion factor, and it is larger when the EQ cost of node $v_i$ is smaller. The larger the value, the higher the probability that the node is selected. The right part is the exploration factor, which is larger when the node $v_i$ is less visited or $p_{v_i}$ is smaller. Note that, $q$ is the adjusting factor of exploration factor to select a mapping child node. The value of $q$ should be set in $(0,1]$, the reason is verified in the simulation.

When the $UCT$ of each node is equal, we will pick the child-node according to delay, rather than select it in random-mode. In the case that the delay affects the selection of child-node, we also calculate the $UCT$ without considering the delay in the selection of child-node, so as to get the prior probability of failure:

$$UCT_{v_i}(\text{without\_delay}) = \frac{N_{v_i}}{C_{EQ,v_i} + \Delta} + c \cdot \frac{\log_2 N_{v_i} + \beta \cdot \frac{N_{v_i}}{C_{EQ,v_i} + \Delta}}{N_{v_i} + \Delta} \hspace{1cm} (39)$$

$$p_{v_i} = \frac{N_{\text{fail}(UCT_{v_i})}}{N_{\text{diff}}(UCT_{v_i} - UCT_{v_i}(\text{without\_delay}))} \hspace{1cm} (40)$$

where $N_{\text{fail}(UCT_{v_i})}$ is the number of failures that takes delay into account; $N_{\text{diff}}(UCT_{v_i} - UCT_{v_i}(\text{without\_delay}))$ is the total number that the results generated by $UCT$ and
The available connections are contraindicated, then a set of taboo length is 0, or the occupied resource is set free. If all connection from case 3) is immediately released when the taboo length is 0. And the taboo table considers the connections in an interrupted state indicates the connections between server k and a series of

\[ \{ v_k \} \] differ. The specific selection is as follows: When the UCT of the child nodes of node v are the same or all the child nodes of node v are not explored (VN_{child} == 0), the node with the smallest processing delay is preferentially selected. Otherwise, select the child node with the largest UCT value.

3) OPERATION STEP

This step generates sub-branch as candidate migration schemes for saving energy. We will explore the sub-branch until achieving its leaf node for the child node selected in step 2. Our model has demonstrated that energy consumption has a nonlinear relationship with the utilization of a server. Therefore, in the child candidate set \( e \in \text{candidate}(v_i) \), the selected child node should satisfy \( \min[C_{\text{energy}, v_i}/((C_{\text{v, b}}+C_{\text{b, v}})+u_{v_i})]. \) The leaf node represents one of the following four cases: 1) reach the manageable server, and the migration scheme is feasible; 2) extend to a migratable server whose delay has exceeded the maximum tolerance delay of SFC; 3) expand to a non-migratable server whose remaining capacity of processing or link is less than the requirement; 4) expand to a server that has been visited in this branch. In the latter three cases, the migration schemes are all infeasible.

To find a better solution within the limited iterations, we design a taboo table of server connection in SFCM-EQA. The data in this table is derived from the above case 2), 3), and 4). The data format is \((v_k, v_1, v_2, \ldots, v_{kn}, \ldots, v_k)\) and indicates the connections between server k and a series of servers are not available. Within the constraint length, the taboo table considers the connections in an interrupted state and uses different rules to release the connections that derive from different sources. The connection from case 2) and 4) will be released when the taboo length is 0. And the connection from case 3) is immediately released when the taboo length is 0, or the occupied resource is set free. If all the available connections are contraindicated, then a set of servers \( v_k \) is reconnected to the servers \( v_{kn} \) with the highest UCT.

By applying the taboo rule, SFCM-EQA can generate a splendid branch for the decision tree and find a feasible solution in a more extensive range.

4) FEEDBACK STEP

After generating the new sub-branch, we calculate the status of the leaf node immediately and update the UCT of nodes in the branch. At the same time, the visiting frequency and the prior probability of failure are also updated.

5) SELECT THE OPTIMAL SOLUTION

When the number of iterations is reached, the algorithm jumps out of step 2)-4). At this time, a decision tree has been built. In a large-scale network, the decision tree may be incomplete due to iteration constraints. Nevertheless, owing to the step 2)-4), the decision tree will contain sub-optimal solutions for the SFC migration. The leaf node of the optimal migration scheme should satisfy \( \min[C_{\text{EQ}, v_i}] \). The path from the root to the leaf node is the path for migration. The algorithm can be described as follow:

**Algorithm 2 SFC Migration Method Based on Energy-Cost and QoS Aware (SFCM-EQA)**

| Initial: decision tree \( T \) |
|--------------------------------|
| Input: \( G(V, L), G_{vi}(V_{vi}, L_{vi}), S \) |
| Output: Schemes_{s,f} |
| 1. While in iterative budget do |
| 2. \( v_i \) ← select node by EQ cost and delay \( (T) \) |
| 3. \( p_{v_i,v} \) ← calculate the prior probability by \( UCT_{v_i,v} \) and \( UCT_{v_i,v}(\text{without_delay}) \) |
| 4. Simulate sub-branch \( path_{sub} \) |
| 5. While not reach leaf node do |
| 6. Select cur node’s child node |
| 7. Update \( path_{sub} \) and taboo table |
| 8. End while |
| 9. Update taboo table |
| 10. End simulation |
| 11. \( T \) ← \( T \cup path_{sub} \) |
| 12. Feedback \( (T) \) |
| 13. update taboo table |
| 14. End while |
| 15. \( path_{res} \) ← select the best migration scheme |
| 16. return \( path_{res} \) |

V. NUMERICAL RESULTS

This simulation includes the following four steps: Above all, use SFCM-CBR for SFC mapping. Then, evaluate the effectiveness of SFCM-CBR in terms of acceptance ratio and cost. After that, use SFCM-EQA to generate migration strategies. Finally, evaluate the performance of SFCM-EQA and compare it with the VNFI minimum power migration algorithm (VMMPC) [10].

A. MAPPING RESULTS

1) SIMULATION SETTINGS

We use Java for simulation. The element of our simulation mainly includes network node (defined by capacity, actual usage capacity, reliability), link (defined by bandwidth capacity, actual bandwidth usage, connected nodes at both ends), SFCs (mainly include the required VNF functions, function sequence, reliability requirements, delay requirements, computation ability, and bandwidth capability requirements). The inputs are the number of nodes \( n \), and the number of services \( \lambda \) reached per second. After that, a network with \( n \) nodes and corresponding SFC requests will be generated. The reliability of the nodes, the number of functions, and reliability requirements of the SFC are randomly generated according to the set range. After completing the resource allocation by applying the proposed algorithm, the evaluation indexes such as the number of successful services, cost, and utilization rate will finally be the outputs.
The network simulated contains 2000-20000 nodes, and these nodes can support 40-80 types of VNFs. λ is the number of SFC requests arriving per second. Each service requests 5-10 VNFs. The requirement of reliability is between [0.9, 1). The reliability of VNF is consistent with the mapped node, randomly set over a range of [0.95, 1). We compare the proposed algorithm with the existing two algorithms: JP + picker [6] and SFCM-POOL [7]. JP + Picker aimed at reducing cost and providing economic backup, while SFCM-POOL focused on the backup pool to assure reliability.

The evaluation indexes of the mapping mechanism include: SFC request acceptance rate and the total cost. Among them, the SFC request acceptance rate reveals the reliability of the mapping scheme with positive relationship. And the total cost is given in formula (5).

2) SIMULATION RESULTS
The acceptance ratio of SFC requests for three algorithms at different λ is shown in Fig.5. It reveals that our proposed algorithm outperforms others in terms of availability of service. Fig. 5.a shows the SFC request acceptance rates of the three algorithms under different λ (vn = 2000). By comparing the performance of the algorithms under the same conditions, we can find that the acceptance rate of SFCM-CBR is higher than others. And, with the increase of SFC requests, the advantages are more obvious. Specifically, when λ = 2000, SFCM-CBR is 7.5% and 16% higher than the other two algorithms respectively. When λ = 5000, SFCM-CBR is 16% and 62.7% higher than the other two algorithms respectively. This is because that, a higher acceptance rate requires better load balance adjustment capability and reliability assurance capability, which are both considered in the proposed SFCM-CBR.

Fig.5.b and Fig.5.c verify the performance of the three algorithms in networks of different sizes, namely, the networks with different traffic loads. We evaluate the performance of the method from two dimensions. Dimension 1: the performance of algorithms under different network scales. When λ = vn, as the number of network nodes increases, the service acceptance rate of SFCM-CBR is always higher than 80%. During the network scale change from e = 2000 to vn = 20000, the reception rate decreases by 6.97%. However, the acceptance rate of the other two algorithms both fell below 80%. The acceptance rate of SFCM-POOL decreased by 10%, while JP + picker even decreased by 13.51%, almost double that of SFCM-CBR. When λ = 2vn, the decreasing trend of acceptance rate is more obvious. SFCM-CBR, SFCM-POOL, and JP + picker decrease by 6.04%, 13.66%, and 20.63% respectively. It can be seen that SFCM-CBR outperforms the other two algorithms when adapting the networks size. Dimension 2: the performance of algorithms for different traffic λ in the network of the same size. For vn = 2000, when λ is vn and 2vn, the acceptance rates of SFCM-CBR, SFCM-POOL, and JP + picker are reduced by 12.79%, 13.66%, and 20.63% respectively. For vn = 20000, it becomes 12.5%, 23.61% and 37.5%. In contrast, SFCM-CBR is the most stable in terms of acceptance rate index with the changing network scale and traffic. The reason is that the
proposed SFCM-CBR balances the cost and the equilibrium factor, that is, it not only provides reliable resources for the service chain economically but also reserves resources for the upcoming service requests to ensure the QoS of subsequent services.

Fig. 6 gives the total cost of the three algorithms. Overall, the order of cost from low to high is SFCM-CBR, JP + Picker, and SFCM-POOL. The reason is that the JP + Picker mainly focuses on reducing costs and improving reliability but ignoring the load, which will lead to the dead node and inversely increase costs. Compared to the other mentioned algorithms, SFCM-POOL spends a lot of redundant backup costs improving the availability of SFC requests.

B. MIGRATION RESULTS

The following list outlines the different types of graphics published in IEEE journals. They are categorized based on their construction, and use of color / shades of gray:

1) SIMULATION SETTINGS

The node and service type settings for the migration scenario are the same as mapping scenario. The simulation includes two parts. Firstly, simulate the network changes after SFC mapping for a while. Then, apply the proposed algorithm in the network to compare with VMMPC. As far as VMMPC is concerned, it proposes a heuristic migration strategy based on the Markov chain and takes the quality of service into account. For comparison of improvement, we also execute the Baseline method that does not implement migration strategies.

We focus on three indicators while simulating: cost, node utilization, and utilization deviation degree. Cost includes energy consumption and revenue loss. The utilization deviation degree means the variance of resource utilization. The performance of the algorithm is evaluated in three levels. The first is the comparison between SFCM-EQA and VMMPC to observe the performance difference between the two migration algorithms. The second is to compare SFCM-EQA and Baseline to observe the improvement degree. The third is the self-comparison of SFCM-EQA to verify the effectiveness of the expansion and exploration mechanism.

2) SIMULATION RESULTS

Fig. 7 reveals the cost of SFCM-EQA and VMMPC. As far as total cost, SFCM-EQA saves 12% to 38% more than VMMPC. Then, we analyze the revenue loss and energy consumption separately. It could be seen that in the high-traffic phase (such as 0, 1, 22, 23), the energy saved in VMMPC is higher than that in SFCM-EQA. While in other phases, the energy saved is less than SFCM-EQA. At the same time, although VMMPC saves energy in the high-traffic phase, the revenue loss caused by frequent migration is much higher than that of SFCM-EQA. In return, it leads to a higher total cost. VMMPC regards the importance of energy consumption and revenue loss as equal and does not adjust the importance of the traffic trend. Therefore, the SFCM-EQA algorithm performs well in reducing total cost.

The result in Fig. 7 appears distinct troughs and crests. We analyze the reason in two dimensions. On the one hand, the total cost of the two algorithms has consistent trends, where crests at \( n = 0 \), \( n = 24 \), and trough at \( n = 12 \). Crests occur in the period of highest-traffic, where the service number is so large that both the online number and the utilization of devices are drastically increasing to lead a high cost. Similarly, it is the lowest-traffic phase at \( n = 12 \), which will result in low cost and the presence of trough. On the other hand, the trend of revenue loss is the second point to be analyzed. For VMMPC, crests occur at \( n = 0 \), \( n = 24 \), and trough at \( n = 12 \). While for the SFCM-EQA, a completely different trend has emerged. The reason is that VMMPC follows the principle of minimum energy consumption and does not consider the traffic status. There is a large amount of revenue loss caused by frequent migration in the large-traffic phase, so crests appear. SFCM-EQA avoids the revenue loss.
in the large-traffic phase at the cost of sacrificing some energy consumption, so the loss curve at this phase is in a trough. When \( n = 12 \), SFCM-EQA focuses on closing idle servers to reduce energy consumption, while this will increase revenue loss, so the crest exists.

**FIGURE 8.** Cost before and after applying SFCM-EQA.

Fig. 8 shows the cost before and after applying SFCM-EQA. Compared with Baseline, our algorithm can reduce 15% to 43% of the cost. In the highest-traffic phase \( (n = 0) \), SFCM-EQA reduces the cost by 15.68%. SFCM-EQA focuses on maintaining the QoS and selects the migration strategy that has little impact on the QoS in the large-traffic phase, which reduced the revenue loss. As traffic decreases, the performance of optimization is improving. For example, at \( n = 12 \), the cost is reduced by 41.66%. Besides, it can be seen that the QoS based migration strategy is changing with the traffic. In the high-traffic phase, SFCM-EQA focuses on maintaining QoS. While in other traffic phases, it focuses on reducing energy consumption by shutting down idle servers to avert resource waste.

**FIGURE 9.** The average utilization and utilization deviation degree of working nodes.

Fig. 9 shows the average utilization and utilization deviation degree of working nodes. Three methods are simulated: Baseline, the VMMPC, and the proposed SFCM-EQA. As seen in the figure, the utilization of Baseline is 90% at \( n = 0 \). After applying VMMPC and SFCM-EQA, it is reduced to 80% and 76%, respectively. The reason is that, in the largest-traffic phase, SFCM-EQA pays more attention to QoS than energy consumption, making it involve a small migration scope and slightly inferior to VMMPC in reducing energy consumption. In \([0, 12]\), the average utilization of all these algorithms is decreasing, and the utilization of SFCM-EQA is mainly between 60% to 70%, which is more stable than the others. Besides, its deviation degree is the smallest. The reason is that SFCM-EQA comprehensively considers the utilization of multiple resources, while the other two methods do not.

Fig. 10 shows the cost by varying \( q \) in the UCT (39), where \( q \) determines the incline of expanding or exploring. We test the strategy with \( q = 0, q = 0.3, q = 0.5, q = 0.7, \) and \( q = 1 \) respectively. When \( q = 0 \) and \( q = 1 \), only the node with the largest expansion or exploration factor is considered. At that time, the reduction is only between 3% and 10%, since the one-side search may result in local optimization. When \( q = 0.3 \), the underdeveloped nodes are also extended when the most valuable node is expanded, so the cost is the lowest.

**FIGURE 10.** Cost with different \( q \).

**VI. CONCLUSION**

The application of IoT is developing rapidly in smart grid, transportation, smart manufacturing, healthcare, finance, smart home, and etc. To provide flexible and reliable IoT service, we propose a two-stage REB-SFCM method for the IoT edge network. First of all, aiming at reducing resource consumption, balancing load, and maintaining reliability, we establish a mapping model that includes SFC mapping and multi-mode backup, and further design SFCM-CBR to search for feasible solutions and share the backup resources. Then, in order to reduce energy consumption, we build a migration model combined with the traffic model to integrate resources. It can dynamically adapt to different traffic
phases to reduce energy consumption and avoid revenue loss. Finally, we design SFCM-EQA on the basis of delay judgment and improve the UCT factor to search for suboptimal solutions. The simulation results show that the proposed algorithm can provide reliable service with relatively low costs. The proposed system model and SFC mapping method can be applied in the IoT edge network. And, it is especially suitable for delay-sensitive services, such as the local energy transaction control in a microgrid, and the fast face authentication.

In future works, we will introduce the traffic prediction model and burst traffic in SFC mapping and migration mechanism design, thus to enable the resource allocation mechanism to perform offline calculation and online adjustment.

REFERENCES

[1] W. Zhou, Y. Yang, and M. Xu, “Network function virtualization technology research,” J. Comput. Res. Develop., vol. 55, no. 4, pp. 675–688, Apr. 2018.

[2] R. Sairam, S. S. Bhunia, V. Thangavelu, and M. Gurusamy, “NETRA: Enhancing IoT security using NFV-based edge traffic analysis,” IEEE Sensors J., vol. 19, no. 12, pp. 4660–4671, Jun. 2019.

[3] X. Fu, F. R. Yu, J. Wang, Q. Qi, and J. Liao, “Service function chain embedding for NFV-enabled IoT based on deep reinforcement learning,” IEEE Commun. Mag., vol. 57, no. 11, pp. 102–108, Nov. 2019.

[4] D. T. Nguyen, C. Pham, K. K. Nguyen, and M. Cheriet, “Placement and chaining for run-time IoT service deployment in edge-cloud,” IEEE Trans. Netw. Service Manage., vol. 17, no. 1, pp. 459–472, Mar. 2020.

[5] W. He, S. Guo, Y. Liang, and X. Qiu, “Markov approximation method for optimal service orchestration in IoT network,” IEEE Access, vol. 7, pp. 49538–49548, 2019.

[6] I. Sarrigianinis, K. Ramantas, E. Kartasiki, P.-V. Mekikis, A. Antonopoulos, and C. Verikoukis, “Online VNF lifecycle management in an MEC-enabled 5G IoT architecture,” Internet Things J., vol. 7, no. 5, pp. 4183–4194, May 2020.

[7] M. T. Beck, J. F. Botero, and K. Samelin, “Resilient allocation of service function chains,” in Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN), Palo Alto, CA, USA, Nov. 2016, pp. 128–133.

[8] J. Fan, C. Guan, Y. Zhao, and C. Qiao, “Availability-aware mapping of service function chains,” in Proc. IEEE Conf. Comput. Commun. (INFOCOM), Atlanta, GA, USA, May 2017, pp. 1–9.

[9] N. Hui, B. Jaumard, and F. Giroire, “Optimization of network service chain provisioning,” in Proc. IEEE Int. Conf. Commun. (ICC), Paris, France, May 2017, pp. 1–7.

[10] J. Liu, W. Lu, F. Zhou, P. Lu, and Z. Zhu, “On dynamic service function chain deployment and readjustment,” IEEE Trans. Network Service Manage., vol. 14, no. 3, pp. 543–553, Sep. 2017.

[11] Z. Zhu, H. Lu, J. Li, and X. Jiang, “Service function chain mapping with resource fragmentation avoidance,” in Proc. IEEE Global Commun. Conf. (GLOBECOM), Dec. 2017, pp. 1–6.

[12] P. Yu, F. Zhou, X. Zhang, X. Qiu, M. Kadoch, and M. Cheriet, “Deep learning-based resource allocation for 5G broadband TV service,” IEEE Trans. Broadcast., early access, Feb. 10, 2020, doi: 10.1109/TBC.2020.2968730.

[13] J. Kong, I. Kim, X. Wang, Q. Zhang, H. C. Cankaya, W. Xie, T. Itakura, and J. P. Jue, “Guaranteed-availability network function virtualization with network protection and VNF replication,” in Proc. IEEE Global Commun. Conf. (GLOBECOM), Singapore, Dec. 2017, pp. 1–6.

[14] L. Qu, C. Assi, K. Shaban, and M. Khabbaz, “Reliability-aware service provisioning in NFV-enabled enterprise datacenter networks,” in Proc. 12th Int. Conf. Netw. Service Manage. (CNSM), Montreal, QC, Canada, Oct. 2016, pp. 153–159.

[15] M. Barrere, R. V. Steiner, R. Mohsen, and E. C. Lupu, “Tracking the bad guys: An efficient forensic methodology to trace multi-step attacks using core attack graphs,” in Proc. 13th Int. Conf. Netw. Service Manage. (CNSM), Tokyo, Japan, Nov. 2017, pp. 1–7.

[16] W. Ding, H. Yu, and S. Luo, “Enhancing the reliability of services in NFV with the cost-efficient redundancy scheme,” in Proc. IEEE Int. Conf. Commun. (ICC), Paris, France, May 2017, pp. 1–6.

[17] J. Fan, M. Jiang, and C. Qiao, “Carrier-grade availability-aware mapping of service function chains with on-site backups,” in Proc. IEEE/ACM 25th Int. Symp. Qual. Service (IWQoS), Vilanova i la Geltrú, Spain, Jun. 2017, pp. 1–10.

[18] H. Ibn-Khedher, E. Abd-Elrahman, and H. Afifi, “OMAC: Optimal migration algorithm for virtual CDN,” in Proc. 23rd Int. Conf. Telecommun. (ICT), Thessaloniki, Greece, May 2016, pp. 1–6.

[19] J. Xia, D. Pang, Z. Cai, M. Xu, and G. Hu, “Reasonably migrating virtual machine in NFV-featured networks,” in Proc. IEEE Int. Conf. Comput. Inf. Technol. (CIT), Nadi, Fiji, Dec. 2016, pp. 361–366.

[20] V. Eramo, E. Miucci, M. Ammar, and F. G. Lavacca, “An approach for service function chain routing and virtual function instance migration in network function virtualization architectures,” IEEE/ACM Trans. Netw., vol. 25, no. 4, pp. 2008–2025, Aug. 2017.

[21] J. Zhang, D. Zeng, L. Gu, H. Yao, and M. Xiong, “Joint optimization of virtual function migration and rule update in software defined NFV networks,” in Proc. IEEE Global Commun. Conf. (GLOBECOM), Singapore, Dec. 2017, pp. 1–5.

[22] F. Carpio, A. Jukan, and R. Pries, “Balancing the migration of virtual network functions with replications in data centers,” in Proc. IEEE INFOIP Network Oper. Manage. Symp., Taipei, Taiwan, Apr. 2018, pp. 1–8.

[23] X. Zhou, B. Yi, X. Wang, and M. Huang, “Approach for minimising network effect of VNF migration,” IET Commun., vol. 12, no. 20, pp. 2574–2581, Dec. 2018.
CONG HAN was born in Hengshui, Hebei, China, in 1995. She received the B.S. degree in communication engineering from Jilin University. She is currently pursuing the master’s degree with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. Her research interests include smart grid communication network management, SDN/NFV, and the Internet of Things.

CHAO YANG was born in 1988. He received the Ph.D. degree in computer applications technology from Dalian Maritime University. He is currently an Engineer with the Information and Communication Branch, State Grid Liaoning Electric Power Company. His research interests include smart grid informatization, the Internet of Things, and artificial intelligence.

ZHILI WANG is currently an Associate Professor with the Beijing University of Posts and Telecommunications, engaged in research and standardization work in communication networks and computer science and technology. His main research directions include network management, communications software, and interface testing. He won one National Science and Technology Progress Awards and wrote more than 8 ITU-T international standards, and currently served as the Working Party 2 Chairman of ITU-T Study Group 2.

AO XIONG (Member, IEEE) was born in Jiujiang, Jiangxi, China, in 1974. He received the Ph.D. degree in computer science from the Beijing University of Posts and Telecommunications, Beijing, China in 2013. He is currently an Associate Professor with the State Key Laboratory of Networking and Switching Technology, BUPT. He has authored about 30 SCI/EI index articles and received 14 national and provincial scientific and technical awards. His research interests include communication network and communication software, the Internet routing and applications, and network management.