IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 October 2022, accepted 7 November 2022, date of publication 14 November 2022, date of current version 17 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3221744

RESEARCH ARTICLE

Delay-Aware Multipath Parallel SFC Orchestration

SONGLIN WEI, JINHE ZHOU, AND SHUO CHEN, (Member, IEEE)
Key Laboratory of the Ministry of Education for Optoelectronic Measurement Technology and Instrument, School of Information and Communication Engineering, Beijing Information Science and Technology University, Beijing 100101, China

Corresponding author: Jinhe Zhou (zhoujinhe@bistu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61901043 and Grant 61872044.

ABSTRACT With the development of network functions virtualization and software defined networking, the service function chain (SFC) orchestration issue is a big challenge for high reliability and low latency services. At present, many studies propose solutions in terms of physical node mapping or link mapping for SFC. In this paper, we consider parallel transmission by dividing the SFC request flow into multiple sub-flows. To solve the problem of orchestrating parallel SFC under the premise of being able to meet the delay requirements of delay-sensitive classes of services, we divide the problem into two parts: virtual network functions mapped to physical servers and virtual links mapped to physical links. In the first part, we find suitable physical nodes for deployment by the simulated annealing algorithm. In the second part, we construct the link mapping problem as a multi-objective optimization problem. We solve this multi-objective optimization problem by quantum genetic algorithm. Finally, the mapping scheme for parallel SFC is generated. We have conducted comparative analyses of the algorithms through simulation experiments. The results show that the method proposed in this paper can effectively improve the orchestration efficiency of parallel SFC. The algorithm we build can not only minimize the resource consumption and routing energy consumption but also meet the delay requirements well. Therefore, this paper has high practical significance in diverse delay-sensitive service applications and provides a solution for future multipath parallel SFC orchestration.

INDEX TERMS SFC orchestration, VNF instance deployment, virtual link mapping, resource allocation, SRv6.

I. INTRODUCTION

Network functions virtualization (NFV) [1] is an emerging technology based on virtualization, where network functions (NFs) are implemented in software and are no longer limited to the hardware devices. With the development of science and technology, technologies such as the Internet of Vehicles and tactile communication applications such as remote surgery emerge one after another. More and more service function chain (SFC) requests will be generated under the combination of software-defined networking (SDN) and NFV. For the traditional method of virtual network function (VNF) instance deployment, more resources are wasted and it is difficult to achieve global control. So traditional methods are difficult to solve a large number of SFC deployment problems and are insufficient user satisfaction guarantee [2]. With the rapid increase of digital content in the network, effectively reducing redundant network traffic and improving user satisfaction are gradually becoming important directions for network development. In 5G/6G [3], the combination of NFV and network slicing has emerged, but the emergence of a large number of service functions has led to poor control and low scalability of network slicing. So in this paper for the multipath SFC orchestration problem, we added the SDN controller for global control. SDN is currently used in many scenarios. SDN has three core concepts, the first is the separation of controlling and forwarding, the second is centralized control of the controller, the third is open application programming interface. With the rapid development of SDN, SDN has been applied to various network scenarios, extending from small enterprises to data centers and from wired to wireless networks, all of which have adopted the separation of control

The associate editor coordinating the review of this manuscript and approving it for publication was Barbara Masini.

plane and data plane to obtain a global view to manage their networks [4]. In traditional networks, network traffic passes through various network middleware in a specific order to complete a NF. Different network middleware providing their packet processing functions. In NFV, network services are implemented in the form of service chains, also known as SFCs [5]. The core concept of NFV is to decouple NFs from hardware devices through virtualization technology. In contrast to traditional networks where NFs run on dedicated hardware (network middleware or network devices), NFV architectures allow NFs to run on commercial off-the-shelf hardware devices in a software manner. As a result, NFV can increase network flexibility and significantly reduce network capital expenditures and operating expenditures.

Under the technical architecture of NFV and SDN, relying on the separation of controlling and forwarding of the control plane and data plane of SDN, resources can be flexibly scheduled and allocated on demand. The network can be dynamically expanded or shrinked, reducing the cost of network construction and operation while meeting the delay constraint. SDN/NFV service chains have the following advantages over traditional data center SFCs: (1) Based on the centralized control of SDN, the type and order of VNF on SFCs can be modified in real-time by formulating policies based on service requests. SDN eliminates coupling among network devices and reduces reliance on topology. (2) When data packets are transmitted in the network according to VNFs order, traditional networks need to go through multiple capsulations and decapsulation between each physical server that carries different VNFs, which will consume more resources and energy. For the SFC construction of segment routing over IPv6 (SRv6)/NFV, the demand flow only needs to be classified once. The SDN controller deploys the SFC according to the different flows of the data flows and the collected network states, then sends the routing and forwarding paths of the data flows to the access device. The controller uses segment routing (SR) to encapsulate and label the demands and paths of the data flows, which is mainly based on the global control capability of the SDN. The controller not only reduces the latency and reduces the consumption of resources and energy but also greatly improves the performance of the SFC.

In recent years, SR has attracted a lot of scholarly attention as a new means of implementing SDN [6]. SR is a source routing technique and it simplifies the protocol, where the data message is ‘‘encoded’’ by the source router and insert segment information into the message at the beginning of the path, i.e., insert an ordered list in the header to indicate the forwarding path of the message. The intermediate network nodes only need to forward the message according to the segment information carried in the message. SR assigns segments to each node or link. The head node combines these segments to form a segment sequence (segment path) and identifies them by segment identifier (SID) to guide the forwarding of messages according to the segment sequence. It is identified by SID, which guides the packet to be forwarded according to the segment sequence, thus realizing the programming capability of the network. SR is considered a key technology for SDN2.0. Because it reduces control redundancy compared to traditional resource reservation and distributed protocols. SR is simple to operate, scalable and has powerful programmability. SRv6 combines the advantages of SR and IPv6's 128-bit programmability [7], which can significantly reduce network complexity.

In order to implement SR technology based on the IPv6 forwarding plane, a new extension header of segment routing header (SRH) is added to the IPv6 routing extension header to store IPv6 segment list information and program the combination of segments to form SRv6 paths. The content stored in the SRH extension header is equivalent to a computer program, which is the solution to the end-to-end connectivity problem of the business. Segment left is equivalent to the PC pointer of a computer program, which always points to the instruction currently being executed.

SRv6 uses a 128-bit segment to define the NFs and then arranges the Segments to achieve a series of forwarding and processing behaviors of the network devices, thus completing the orchestration of the SFC. When designing SRv6 programming, it is necessary to define a network instruction - SRv6 segment, which is identified as SRv6 SID. SRv6 SID is a 128bit value and each SRv6 SID is a network instruction, usually containing three parts: Locator, Function and Argument, then we subsequently define them for the SFC arrangement in this paper.

With the combination of NFV and SFC, it is a big challenge for the massive SFC orchestration problems. With a large number of latency-sensitive services, it will lead to VNF deployment problems and cause a lot of congestion in the network, resulting in network performance problems such as rising transmission delay. At the same time, problems in network architecture and resource allocation management lead to high system energy consumption and low network resource utilization. Therefore, we build the system model based on SRv6. The global view of the SDN controller can realize the optimal allocation of resources, the load status of storage resources and the congestion of physical links can be obtained by collecting the network state in the physical network in multiple dimensions. The controller in SRv6 can dynamically change the mapping path of SFC flows according to user requests and load dynamics in the network. Ultimately, effective adaptation of routes and rational allocation of resources and global load balancing are achieved. For operators, the minimum resource consumption can greatly reduce the deployment cost, but it may lead to over-concentration of load, resulting in excessive load on some links [8], which reduces the reliability and the deployment success rate for the arrival of subsequent SFC requests and affects the revenue of network operators. So, we take the load problem and the number of node hops [9] into consideration in the subsequent contents of the multi-objective SFC link mapping. In this paper, the optimal solution is found among the factors such as minimum resource occupation and balanced network load distribution. In the follow-up content of this article, we focus on the optimization problems of resource consumption, network energy consumption and load allocation caused by the mapping of multiple sub-flows of SFC to multiple physical paths under different delay services. We model the resource allocation problem for SFCs and solve this non-deterministic polynomial hard (NP-hard) problem by a heuristic algorithm.

A. MOTIVATIONS

This paper addresses the orchestration of a multipath parallel SFC [10]. Because the traditional single-path routing method is difficult to accomplish for many high reliability and low latency services. Such as, Baumgartner et al. [11] proposed a mixed integer linear programming formulation aimed at minimizing the cost while meeting the latency requirements of the service. Some of these works focus on latency-constrained services, but they all use a single-path routing approach and do not consider service splitting, so they may not be able to meet the strict latency requirements. In the problem of parallel SFC orchestration for multiple paths, most of the existing literature focuses on VNF instance replication, then proposes a multi-flow backup model to provide backup VNFs for working VNFs [12], thus improving reliability. The placement algorithm with better performance not only maps the SFCs to the data center but also effectively reduces resource consumption. By using a parallel VNF processing method that replicates multiple copies of service flows and transmits them in parallel. The reliability of packet delivery will be guaranteed in case of severe blocking in a path or node failure in the path. Because it is a complete copy of the service flow in multiple copies, then transmit in parallel, the complete service flow can be sent to the end device through SFC as long as there is no failure of the physical device or VNF instance in one path, which can also ensure reliability. But SFC backup will not only add more physical nodes to host the backup VNF instances but also replicate multiple flows for transmission when the data flows are transmitted, which not only does it take up a lot of computing resources as well as bandwidth resources in the physical network but also generates a lot of discarded packets, which easily causes network path blocking and leads to network paralysis. At the same time, by replicating multiple flows, it not only causes waste of resources but also generates a lot of energy consumption, which is also a big challenge for low latency and high reliability services. For latency-sensitive classes of services, such as remote surgery, driverless, etc., service requests may be subject to latency constraints. When latency-sensitive services are considered as a collection of multiple VNFs, a key challenge is to deploy VNFs and direct traffic through these service functions while meeting strict latency conditions.

For single-path SFCs, it is difficult to solve the problem of SFC deployment of latency-sensitive services, so parallel VNF processing is considered a promising approach. Therefore, we aim to integrate the endpoint by replicating multiple instances of VNFs and then splitting the data flows through multipath parallel processing through these VNFs. We propose the deployment of VNF with minimal resource utilization (MRU-VNF) solution, which aims to determine the latency sensitivity of the service after latency awareness, then calculate the link mapping location that takes up the least resources and has a balanced load in the link as well as high availability/ reliability of the VNF instance deployment location. The proposed multipath refers to more than one path and contains one path. For delay-insensitive services, the solution proposed in this paper can also be used without applying to multipath routing planning. Even if the SFC scheduling problem is solved in single-path routing, the performance of the solution proposed in this paper is the best in terms of resource consumption, load distribution and cost in general compared with other scheduling methods.

The multipath SFC proposed in this paper first needs to solve the problem of VNF deployment. First, the simulated annealing (SA) algorithm is used to find the physical server with the highest availability under the constraint conditions to deploy the VNF instance. If the VNF instance is not effectively deployed on the physical server, the network performance will have a great impact. For example, when a physical server or virtual machine (VM) fails, the VNF instance needs to be re-deployed, which will prolong the overall transmission time of SFC requests. The forwarding rules stored in the router will increase sharply. SFC requests require more rules to reroute, it reflects the importance of reliability in instance mapping [13], so we first deploy the instance. Then we use the quantum genetic algorithm (QGA) to solve the virtual link mapping problem under the condition of satisfying user requirements and network condition constraints. The purpose of this paper is to find a physical link that can meet the least resource occupancy and balance the link load distribution for SFC mapping by judging the number of offloads through delay sensing under the constraint conditions.

Nowadays, many engineering design problems and practical engineering applications are emerging. The highly nonlinear optimization problems can be found everywhere in many engineering applications. Most of these engineering problems and the SFC deployment problems mentioned in this paper are solved by using heuristic optimization methods. There are some better heuristic optimization algorithms that currently receive wide attention, such as Yuan et al. to tackle these problems, a novel assisted optimization strategy, named elite opposition-based learning and chaotic k-best gravitational search strategy (EOCS) [14], is proposed for the grey wolf optimizer algorithm. In the EOCS-based grey wolf optimizer algorithm, the elite opposition-based learning strategy is proposed to take full advantage of better-performing particles for optimization in the next generations. A chaotic k-best gravitational search strategy is proposed to obtain the adaptive step to improve the global exploratory ability. They also proposed a novel population intelligence optimization algorithm, which is named alpine skiing optimization (ASO) [15]. The main inspiration of the ASO originated from the behaviors of skiers competing for the championship. In the ASO, physical stamina and sprint are two essential factors for skiers to win the tournament, which are similar to the two stages of exploration and exploitation. In our work, we mainly design and use optimization algorithms to solve the proposed multipath SFC orchestration problem, after which we use two heuristic algorithms to solve the problem.

There are many discussions on VNF deployment issues, mostly considering the cost issues in the deployment problem. There is very little literature to consider the cost while jointly considering the energy consumption of physical devices, especially in the parallel SFC deployment problem. Because the replication of VNF instances will lead to occupying more physical devices, which will lead to generating more energy consumption. Therefore, which will lead to a significant increase in the cost of physical devices required by network service providers to deploy SFC. We aim to minimize the cost and energy consumption of network service providers while meeting the latency service requirements. Minimizing the resource consumption to map SFCs while satisfying the constraints, which will also help us to build green networks in the future. Specifically, we use SFC mapping and routing strategies based on SRv6 technology. The importance of quality of service (QoS) cannot be ignored when mapping VNF instances and virtual links [16]. Service providers should consider QoS requirements to meet the service level agreements promised to applications. For networks with limited capacity, QoS assurance is crucial. Its key indicators include reliability, throughput, delay, packet loss rate, etc. Therefore, we set the relevant QoS demand conditions as constraints, set the objective function under the conditions that satisfy the constraints, and then we use a heuristic algorithm to solve the deployment problem. Finally, we can see that we proposed algorithm can well solve the SFC scheduling problem with multipath splitting through algorithm comparison.

B. CONTRIBUTIONS

The main contributions of this paper are as follows: In order to solve the problem of SFC deployment under different latency requirements, we proposes a parallel SFC orchestration scheme with multiple paths, which can not only solve the demand problem for delay-insensitive services but also get a good solution for the services with higher delay requirements. Then, We built a system model based on SRv6 to take advantage of its SR and the global control of IPv6. Next, we propose the minimum resource occupation VNF deployment algorithm to seek solutions to the proposed availability objective function as well as the multi-objective optimization problem, which can well find the highly available physical servers as well as the routing path with the shortest latency. Specifically, we divide the multipath SFC orchestration problem into the VNF instance mapping problem and the virtual link mapping problem. First, we use availability as the main judgment parameter to find VNF instance mapped physical servers to ensure the effectiveness of VNF instance deployment, which can reduce unnecessary delays. Next, we construct a multi-objective optimization problem aiming to find the physical link that consumes the least resources and can make the network link load evenly distributed. Our solution of this problem can reduce the user cost as much as possible and protect the profit of the network operators.

The rest of the paper is organized as follows: in Section II, we provide an overview of related work. In Section III, our contributions and system model related to multipath SFC orchestration are presented. In Section IV, our network model is presented along with the problem construction, then heuristic algorithms are proposed for the solution. In Section V, simulation evaluation is performed. Section VI concludes.

II. RELATED WORKS

Although energy-efficient VNF layout is an area that has been studied by many scholars in the current NFV environment, the VMs layout and VNF layout issues in the NFV/SFC paradigm are different in many ways. The solution to this layout problem should include routing and path assignment through these ordered VNFs, which makes it a fundamentally difficult problem to solve [8], [17]. Ruizen et al. [18] analyze the layout of VNFs in a 5G network and propose a traffic prediction algorithm, which is to estimate the maximum value of traffic in a network slice and the maximum number of VNFs in a certain period [1]. In addition, the authors propose a VNF deployment strategy that can significantly reduce the probability of service blocking and consumption of resources [1]. However, the authors' proposed method only considers the mapping problem of VNFs and does not consider the routing problem of virtual link mapping. The cost and routing energy consumption of the operator are also important when deploying SFCs. Yalaet et al. [19] proposed that the trade-off between the deployment problem of SFCs and the availability of the service. In their work, they further proposed a polynomial-time heuristic by which it facilitates the proper allocation of computational resources to VNF instances [1]. However, they do not consider the latency problem of orchestrating SFCs. Gao et al. [20] consider cost-effective VNF layout and scheduling in public cloud networks. They also consider the dynamic request of the ordered sequence of VNFs, then propose a cost-effective scheme to solve the VNF layout and scheduling problem in public cloud networks. Finally, they show that the algorithm has better performance through simulation results. However, the authors do not consider the problem of routing traffic in link mapping. The energy consumption of routing and the cost of the network are also critical factors. Nowadays, a large number of works have studied the VNF layout and routing problem with different objectives, such as minimizing the total deployment cost [21], [22], [23], minimizing network resources [22], and maximizing reliability [25], [26], [27], [28], etc., but none of these research works consider these properties well together. However, in this paper, by dividing the SFC orchestration problem into two sub-problems, then the MRU-VNF algorithm proposed in this paper solves the above problem very well.

The network slicing embedding [29] and NFV resource allocation (NFV-RA) problems [30] have also received a lot of attention under the consideration of NFV in combination with SFC. The existing solutions for these two problems are divided into two main types, single-path routing-based approaches and multipath routing-based approaches. The authors consider a single-path routing-based approach in [11], [22] [31], [32] [33], [34], and [35], although delay is considered in their paper, performance such as availability or reliability is not considered. Gouareb et al. [22] propose the heuristics to solve the VNF layout and routing problem in edge cloud networks based on NFV with the aim of minimizing the overall delay. Such as Baumgartner et al. [11], some of these works exist that focus on latency-constrained services, but they all use a single-path routing approach and do not consider service splitting, so they may not be able to meet the strict latency requirements. The multipath routing is further divided into two main types of operations: multipath routing that replicates service flows in multiple copies and another type of multipath routing that splits service flows into multiple copies. Authors considered multipath routing through backup instances or backup paths [25], [36] [37], [38], [39], Guerzoni et al. [37] focused on finding the minimum number of link backups required while satisfying SFC reliability requirements, but their work only guaranteed reliability and did not consider the performance of latency. Qu et al. [36] proposed to prevent node failures by backing up multiple VNF instances, the QoS is not guaranteed by backing up multiple VNF instances and using multipath routing for transmission. They aim to maximize the resource utilization with guaranteed reliability and latency requirements. It is worth noting that these efforts are only considering multipath routing through backups, rather than splitting service flows for parallel transmission to accelerate processing latency, which may not satisfy service flows with higher latency requirements.

Chua et al. [40] considered the case of replication of each VNF instance in SFC and traffic splitting, they proposed heuristic algorithms to solve the problem with the aim of reducing resource usage while improving network performance. Zhang et al. [41] also proposed heuristic algorithms, although they also used to split the service flow into multiple copies for multipath parallel transmission, but only one instance of each VNF type can be used for data processing, rather than multiple instances together. Their traffic partitioning aims to minimize link traffic. The current work that focuses on considering partitioned traffic is [40], [41] [42], [43], [44], and [45]. However, all these above mentioned works do not consider factors such as physical device availability or reliability well while considering latency, so they cannot guarantee the overall availability of SFC after a good VNF instance of deployment.

In their paper, Promwongsa et al. [10] propose the slicing technique introduced in 5G. Deploying SFC in delay-constrained slicing is a critical issue. The authors think parallel VNF processing as a promising approach, then pose the problem of minimizing cost while ensuring strict delay constraints. Although they consider VNF mapping and link mapping jointly, then propose a good solution for solving the construction problem of delay-constrained slicing, they do not take into account issues such as device availability well when mapping. When building SFCs over network slices via NFV technologies, the two issues of network slicing embedding and NFV-RA need to be considered in combination. Since the NFV-RA problem also requires considering the number of VNF instances needed in the SFC and the allocation with scheduling of VNFs to the NFV-based infrastructure [30]. Some solutions consider service splitting when mapping VNF instances to the infrastructure [40], [42] [43], [46]. However, none of them consider the latency requirements or the availability of the devices when considering splitting the traffic. Therefore, these solutions are unlikely to be applicable to the implementation of latency-constrained services for certain critical applications.

In this SFC mapping problem, a set of virtual nodes are deployed, virtual links are mapped and the physical devices carrying VNFs in the physical network are required to be linked in an orderly manner. In practice, the SFC mapping problem is performed dynamically for massive requests. So dynamic SFC mapping problem can be reduced to a graph embedding problem, which is usually NP-hard and unapproachable [47], [48]. Therefore, most algorithms are not suitable for solving this highly reliable and low latency problem with large size in practical applications. So, it cannot solve the problem in a reasonable amount of time. This problem can lead to low success rate and poor performance of deployment. Therefore, in the next sections, we propose an online MRU-VNF algorithm to solve the above problem in polynomial time.

III. SYSTEM MODEL

The orchestration of a multipath parallel SFC consists of three parts: VNF ordering, deployment, and routing. Most papers consider only the deployment problem of VNFs or only the routing problem of service flows through ordered VNFs. However, we divide the orchestration problem of parallel SFC into two sub-problems [49], the first one is the mapping problem of VNFs, the second one is the virtual link mapping and routing problem of SFC service flows. We consider the number of hops of routing nodes in the physical network and the energy consumption of routers/switches. We combine SRv6 technology to improve the performance of delayed services, minimize the consumption of resources, improve the resource utilization through the global control capability and network state collection capability of SRv6.

A. SFC FORWARDING CHART

This paper divides the parallel SFC orchestration problem into two sub-problems, the VNF instance deployment problem and the link deployment problem [10]. We aim to deploy VNFs when only limited resources are available in the network, to meet the service delay requirements. Because from

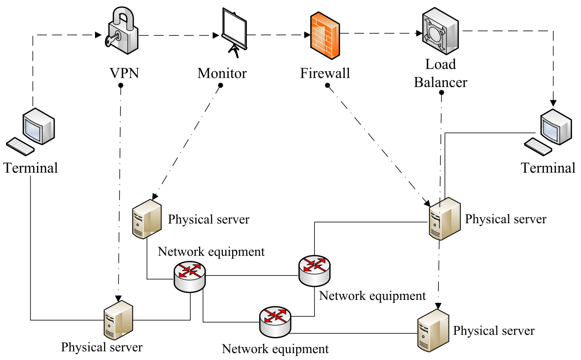


FIGURE 1. VNF deployment model.

the perspective of network service providers, when the link load distribution is balanced, SFCs can still be well deployed when some SFCs have been deployed. The algorithm proposed in this paper has a higher deployment completion rate and can improve the service provider revenue. The idea of parallel SFC in this paper is mainly by splitting data flows into multiple data sub-flows, then passing through SFC in parallel and consolidating them at the terminal. The method we proposed not only reduces the residual bandwidth requirement of the physical link but also improves the transmission speed of multiple sub-flows, which is a good advantage in guaranteeing delay-sensitive services. Unlike the previously mentioned method that improves reliability by replicating multiple data request flows for parallel transmission, our method can better save resources and guarantee latency.

A network service is a complete end-to-end processing service that can include one or more VNFs or middleware devices, such as a "network protection system" service that includes a firewall, deep packet inspection and a virus scanner. As shown in Figure 1, this is a collection of VNF instances required for remote surgery, which contains a variety of VNFs, virtual private network (VPN), monitor, firewall and load balancer. In the shortest possible time to find the most suitable mapping VNF VM, which is a key issue. The subsequent discussion in this paper uses each SFC sub-chain in each VNF separately deployed in a different physical server on a certain VM. The VM can only deploy a VNF in that SFC request. After the discussion, the equipment that hosts the VNF in the sub-chain can be directly represented by the physical server.

1) DEPLOYMENT MODEL OF VNF

In the deployment problem, most of the existing solutions to the VNF deployment problem use exact solutions, approximate solutions, or heuristic algorithms to solve the problem. In this paper, we construct the objective function to maximize SFC availability and use a heuristic optimization algorithm to solve the objective function in the deployment problem. Because the solution algorithm proposed in this paper is mainly for the delay-sensitive class of services. However, the time consumed by the exact solution method is too long. Therefore, in this problem, we use a SA algorithm to solve the problem and find the optimal physical nodes for each VNF deployment by iterating continuously. According to simulation, we can see that the algorithm has a good solution performance that can effectively reduce the deployment cost and improve reliability, which will be discussed in detail in the subsequent sections on VNF deployment. Using NFV technology, an SFC instance can be defined as a collection of end-to-end traffic processed through an ordered set of VNFs, also known as a VNF forwarding graph, as shown in Figure1.

2) SFC LINK MAPPING MODEL

In the SFC mapping problem, the objective is to find a reliable, low-blocking physical link while meeting the latency requirements and minimizing the occupied network resources. Therefore, in the subsequent content, the problem is solved by using heuristic optimization algorithms. Traditional ant colony algorithm and genetic algorithm (GA) have a distributed nature, strong robustness. These algorithms are easy to combine with other algorithms, but has disadvantages such as slow convergence rate and easy to fall into local optimum. Therefore, we use the QGA, which not only has a global search capability but also has powerful programmability, which enriches the performance of the algorithm. We also effectively combine the QGA and SRv6, which can obtain the current network condition in real-time and fast when solving, then solve the optimal link mapping problem, i.e., the routing path of SFC request flow through the routing node. Specifically, the network state in the physical network is first collected by SRv6. The QGA is then allowed to perform iterative solution, giving the fitness function through the constructed multi-objective optimization problem. The fitness function can lead the iterative evolution in the algorithm. The diversity of solutions is ensured by the update operation of chromosomes implemented by quantum revolving gates.

B. SYSTEM MODEL

In this paper, the system model is constructed as an SRv6-based system model. As shown in Figure 2, we use the separation of the control plane and the data plane to construct the system model. The network control plane is an SDN controller, which is responsible for collecting the network state in the data network, then calculating the routing path and issuing commands to the access devices. The data network layer uses SRv6, taking advantage of its 128-bit programmability. The SDN splits the control plane from the data forwarding plane, enabling the management and allocation of global resources through a centralized controller. The design and operation of the network are greatly simplified because the instructions are provided by the SDN controller rather than by the devices and protocols used by multiple network providers.

For IP services, in addition to requiring the network to deliver IP messages to their destinations accurately, it is usually required that the messages be forwarded in a specified order through a series of physical devices carrying VNFs for processing, such as VPNs, firewalls, load balancers, etc.

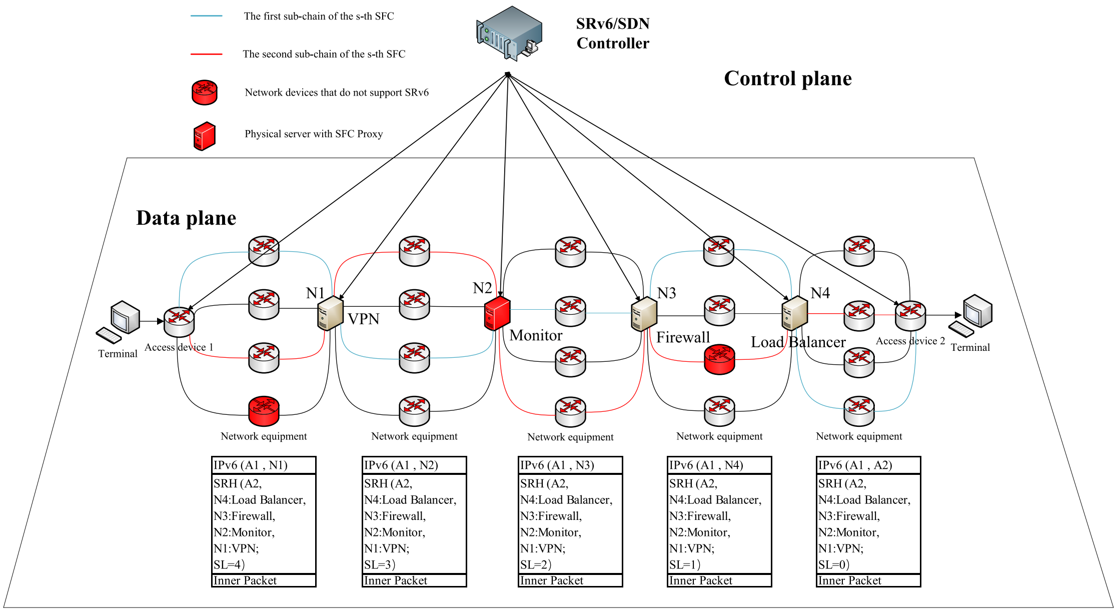

The first sub-chain of the s-th SFC
The second sub-chain of the s-th SFC
Network devices that do not support SRv6
Physical server with SFC Proxy
SRv6/SDN Controller
Control plane
Data plane
Terminal
Access device 1
N1
VPN
N2
Monitor
N3
Firewall
N4
Load Balancer
Access device 2
Terminal
Network equipment
Network equipment
Network equipment
Network equipment
Network equipment
IPv6 (A1 , N1)
SRH (A2, N4:Load Balancer, N3:Firewall, N2:Monitor, N1:VPN; SL=4)
Inner Packet
IPv6 (A1 , N2)
SRH (A2, N4:Load Balancer, N3:Firewall, N2:Monitor, N1:VPN; SL=3)
Inner Packet
IPv6 (A1 , N3)
SRH (A2, N4:Load Balancer, N3:Firewall, N2:Monitor, N1:VPN; SL=2)
Inner Packet
IPv6 (A1 , N4)
SRH (A2, N4:Load Balancer, N3:Firewall, N2:Monitor, N1:VPN; SL=1)
Inner Packet
IPv6 (A1 , A2)
SRH (A2, N4:Load Balancer, N3:Firewall, N2:Monitor, N1:VPN; SL=0)
Inner Packet

FIGURE 2. System model.

If the traditional routing protocols are used to direct these service flows to these physical devices for processing, it will lead to complex configuration, poor flexibility and scalability. Traditional routing protocols may be difficult to achieve delay-sensitive services, so the technology and standard of common SFC defined by IETF are adopted. It is mainly divided into two major parts namely: the control plane and the data plane. The control plane adopts an SDN controller. The data plane divides the forwarding devices of service chain messages into the following types according to the realized functions: service function is responsible for adding the relevant encapsulation of the SFC to the specified service messages and diverting them to the service chain. Service function forwarder is responsible for forwarding the relevant messages to the corresponding service function processing, which in this paper refers to physical servers and VMs. Service function refers to a specific service function, which in this paper is in the form of VNF. We set the SFC proxy device when VNF does not support the SRv6 message. When VNF does not support SRv6 message encapsulation format, it is necessary to implement and deploy a proxy device. The proxy device is responsible for removing/restoring the SFC-related message encapsulation format. SRv6 SID is the instruction to guide the forwarding of SRv6 messages. In the process of programming SFC, SRv6 SID can be divided into two semantics, i.e. topology semantics and service semantics. SID topology semantics is usually used to guide the forwarding path of SRv6 messages in the network. SID service semantics is used to instruct SRv6 messages to implement specific network service functions on a specified device, such as VPN, firewall, etc. Therefore, in SRv6 networks, using the SID service semantics to implement service functions is the best choice. Compared with other different functions in SRv6, in SRv6, SFC is to define both the service semantics and topology semantics of SRv6 SID. SRv6 service function chaining is based on SRv6 SID service semantics (called service SID) to implement the relevant functions of SFC, i.e., using SID List to represent a series of VNFs that need to be executed in order. Each VNFs exist as separate segments and are identified by SIDs.

First, in this paper, the SDN controller uniformly orchestrates the service SID and topology SID, simplifying the implementation and maintenance of traditional technologies, then distributes it to the access device 1 in the form of SRv6 policy. When deploying the SFCs, the access device 1 acts as the flow classifier of the SFC. When the access device receives a message from a user service request, it iterates to the SRv6 policy, completes SRH encapsulation, then starts forwarding. For intermediate network forwarding devices (such as switches, routers, etc., but they do not all necessarily exist), which are handled according to the regular forwarding process of SRv6. For the topology semantics, we define the Locator in the topology SID as the next hop address in the network, Function as the forwarding action, Arguments to store the user requirements of the SFC request flow and the number of diversions. For the service SID, we define its Locator as the identification of the SFC flow requirements and the address of the VNF, Function as the index of the mark-bearing VNF and the functional parameters, Arguments to store the requirements of the service flow and other relevant information such as parameters or services corresponding to the processing in the VNF.

In this paper, physical devices and VM hosting VNFs are defined as service function forwarder. In the following discussion, we assume that some network devices and physical devices support SRv6, those that do not support SRv6 only

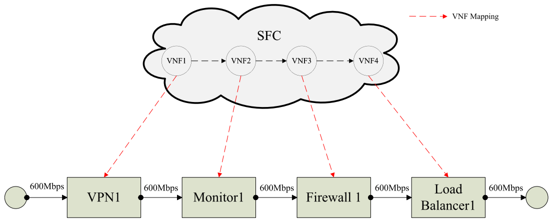


FIGURE 3. SFC without splitting.

need to be responsible for the normal forwarding function. Some VNFs do not support SRv6, so SFC proxy will be added to the physical devices that host VNFs to implement the proxy function. The proxy function is responsible for the function of decapsulating data flow packets before VNF processing and encapsulating them after processing. VNFs supporting SRv6 can directly process SRv6 packets to match SFC-related functions, which makes the model more realistic.

Data plane devices consist of customer equipment and routing device. The Layer 2 data link layer terminal device is responsible for sending and receiving SFC service flows. The Layer 3 network layer contains access devices and trunk devices. Specifically, the access device 1 sends the demand characteristics of the service flow sent by the sender to the SDN. The SDN gives the link mapping and service flow routing path results, then sends them back to access device 1. Finally, the entire path is encapsulated into F-SL via SRv6. The routing device performs forwarding according to the SRv6 SID List. The access device 2 unencapsulated SRv6 and forwards the SFC service flow to the end device.

C. PARALLEL SFC MODEL

The parallel transmission uses shunting for data transfer. The SDN collects the network state, path blockage and bandwidth resources in the network to determine the weight of each branch, calculates the routing path, then performs parallel transmission. As shown in Figure 3, this figure shows an SFC under normal operation, assuming that the bandwidth of the service flow is 600 Mbps and the time for the service flow to pass through the SFC is 4s.

When the service flow is a latency-sensitive service with high requirements for latency, it is necessary to complete the service in a shorter time effectively by splitting the flow and transmitting it in parallel. We divide all requested services into three latency-sensitive cases: the first is a particularly latency-sensitive service, then set the split data flow to three. The second half-sensitive service, then split the data flow into two. The third is not a latency-sensitive service, then do not split the data flow, using single-path routing mode of transmission. The settings are stored in the SDN controller, when a service request arrives, the controller determines the number of split flows according to the user's demand, then uses the algorithm proposed in this paper to determine the location of each SFC sub-chain mapping.

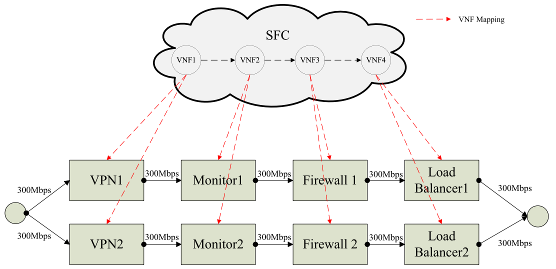


FIGURE 4. SFC in the case of diversion.

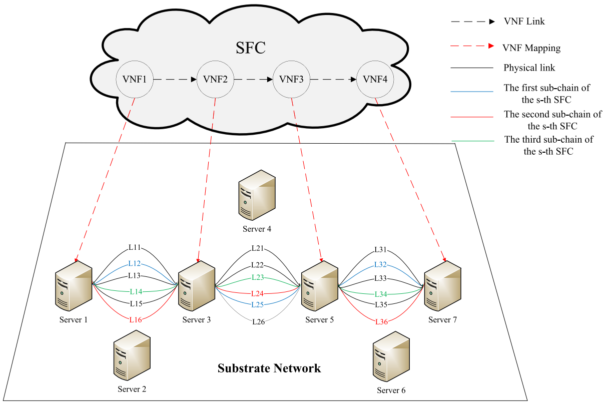


FIGURE 5. Multipath parallel service function chain.

As shown in Figure 4, assuming a good network state and sufficient bandwidth resources for the path, the service flows can be bifurcated according to a 1:1 weighting. As shown above, with each branch responsible for transmitting 50% of the service flows and consolidating them at the terminal. In the ideal case, the time consumed for that service is 2s, effectively reducing the service time. Each path only needs to work with 300Mbps of the remaining bandwidth of the link, at which point more remaining bandwidth resources can be saved and then allocated to other services.

IV. NETWORK MODEL AND ALGORITHMS

A. NETWORK MODEL

In this paper, the network model is constructed as an undirected weighted graph $G = (N, L)$ to represent the physical transmission network. Considering each physical server and VM as one node, V VMs can be deployed on each physical server, one type of NF can be deployed on each VM. Depending on each user request, different SFCs are constructed. Different VNFs on the SFCs process individual data flows with different latency. The same kind of VNFs process individual data flows with the same latency. Detailed parameters can be referred to Table 1.

B. PROBLEM DESCRIPTION

As can be seen from Figure 5, we show the multipath parallel model and set the link-related parameters in Tables 2, 3, and 4. The SFC request flow is set to 600Mbps.

TABLE 1. Main notations.

Notation	Description
G	physical network topology diagram
N	the set of physical nodes
L	the set of links on the physical network
F	the set of VNFs of SFC sub-chain
f_i	the $i-th$ NF
V_{n_i}	the set of VMs on the $i-th$ server
$v_m^{n_i}$	the $m-th$ VM on the $i-th$ physical server
$l_{ij}=(n_i,n_j)$	between physical node n_i and node n_j physical link
$B_{l_{ij}}$	the available bottleneck physical link bandwidth between physical nodes n_i and n_j
$T_{l_{ij}}$	the transmission delay of the physical link between physical nodes n_i and n_j
SF	the set of SFC requests
S	$S\in SF$ represents an SFC request
s_i	$i-th$ sub-flow of the S request
F^s	the set of VNFs in S
L^s	the set of logical links in S
$l_{ab}^s=(f_a^s,f_b^s)$	the logical link in f_a^s and f_b^s
$cpu_{f_a^s}$	the CPU capacity that f_a^s needs to occupy
$mem_{f_a^s}$	the memory capacity that f_a^s needs to occupy
CPU_{n_i}	the available CPU capacity of n_i
MEM_{n_i}	the available memory capacity of n_i
$t_{f_a^{s_i}}$	Processing latency on VMs mapped by each VNF of each SFC $sub-chain$
τ_{s_i}	Transmission delay of each SFC $sub-chain$ on the physical network
$bw_{l_{ab}^s}$	the required bandwidth consumption of the links in f_a^s and f_b^s
$B_{l_{ij}}$	the bandwidth requirement of the link between physical nodes n_i and n_i
C_{sfc}^s	the availability of SFC in $S\in SF$
$\alpha_{n_i}^{f_a^s}$ and $\beta_{l_{ij}}^{l_{ab}^s}$	the binary variable
$\alpha_{n_i}^{f_a^s}$	f_a^s(the $a-th$ VNF in s) whether to map to n_i (On the physical server numbered i)
$\beta_{l_{ij}}^{l_{ab}^s}$	whether l_{ab}^s is mapped to the physical link l_{ij}

TABLE 2. VNF1->VNF2 link parameters.

	Number of node hops	Link remaining available bandwidth resources(Mbps)
L11	3	300
L12	2	1200
L13	3	600
L14	2	1100
L15	5	1000
L16	3	800

First, we perform availability calculations to find a suitable physical server for VNF instance deployment. Table 2, Table 3 and Table 4 give the link-related parameters (i.e., the number of node hops and the remaining available bandwidth resources of the link) between the physical servers hosting the VNF instance. We first find the route for the first sub-chain path, the same for the second and third sub-chain paths. We find the link that satisfies the bandwidth resources required by the request flow and occupies the least resources in the network (i.e., the link with the smallest number of node hops) as the physical link of the mapped virtual link. If the number of node hops of a link are the same, we find the link that can make the load distribution in the link balanced (i.e., the link with the largest remaining available bandwidth resources). Then we can find the routing paths of all sub-chain by this method. Next, the request flow is split and passed through the link in parallel. This method of link mapping can significantly reduce the transmission delay and protect the revenue of the network operator. More and more SFCs can be deployed through balanced load distribution. The least resource consumption can reduce the cost of users and improve user satisfaction. The solution will be elaborated in more detail in the follow-up of the article.

TABLE 3. VNF2->VNF3 link parameters.

	Number of node hops	Link remaining available bandwidth resources(Mbps)
L21	5	400
L22	3	700
L23	4	1000
L24	4	1100
L25	3	1400
L26	5	1000

TABLE 4. VNF3->VNF4 link parameters.

	Number of node hops	Link remaining available bandwidth resources(Mbps)
L31	4	800
L32	2	1000
L33	5	900
L34	3	900
L35	2	300
L36	4	900

C. VNF MAPPING PROBLEM CONSTRUCTION

1) BINDING CONDITIONS

When mapping VNFs to VMs, it is necessary to consider whether the available CPU resources and memory resources of the i-th physical device satisfy the CPU and memory resources required for mapping $f_a^{s_i}$ to $v_m^{n_i}$. Among them, $f_a^{s_i}$ represents the a-th VNF in the i-th sub-flow requested by the s-th SFC. $v_m^{n_i}$ represents the m-th VM in the i-th physical device.

$$\sum_{f_a^{s_i}\in F^{s_i}} cpu_{f_a^{s_i}}\cdot \alpha_{v_m^{n_i}}^{f_a^{s_i}} \leq CPU_{n_i},\quad n_i\in N \tag{1}$$

$$\sum_{f_a^{s_i}\in F^{s_i}} mem_{f_a^{s_i}}\cdot \alpha_{v_m^{n_i}}^{f_a^{s_i}} \leq MEM_{n_i},\quad n_i\in N \tag{2}$$

In constraint (1), CPU_{n_i} represents the available CPU resources of the i-th physical device, $cpu_{f_a^{s_i}}$ represents the CPU resources required to deploy $f_a^{s_i}$. $\alpha_{v_m^{n_i}}^{f_a^{s_i}}$ is a binary decision variable. It is 1, which means that $f_a^{s_i}$ is deployed on $v_m^{n_i}$, otherwise it is 0. In constraint (2). MEM_{n_i} represents the available storage resources of the i-th physical device. $mem_{f_a^{s_i}}$ represents the storage resources required to deploy $f_a^{s_i}$.

In this paper, the VNFs on the SFC in the set SF are mapped to the VMs by the mapping algorithm. When the NFs deployed on the $v_m^{n_i}$ are the same as those on f_a^s, or no NFs are deployed on $v_m^{n_i}$ yet, then f_a^s can be deployed to the $v_m^{n_i}$ if the constraints of inequality (1) and inequality (2) are satisfied.

$$\alpha_{n_i}^{f_a^s} = \begin{cases} 1, & \text{if VNF type a is hosted on physical server } n_i \text{ of traffic flow s} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

A binary variable $\alpha_{n_i}^{f_a^s}$ of 1 indicates that the VNF_a of the s-th SFC is deployed in n_i, otherwise it is 0.

In this paper, each physical server and VM is set as one node. Although there are multiple VMs in each physical server, only one VM in each physical server can deploy a VNF in a certain SFC sub-chain, which means that different

VNFs of each sub-chain in SFC need to be deployed in different physical servers.

$$\alpha^{f_a^{s_i}}_{v_m^{n_i}} = \begin{cases} 1, & \text{if } f_a^{s_i} \text{ is hosted on the m-th VM of the i-th physical serve} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

In SRv6/NFV networks, it is important to ensure that each VNF in each sub-chain of the SFC is deployed on only one VM in one physical device.

$$\sum_{n\in N}\sum_{v_m^n\in V_n} \alpha^{f_a^{s_i}}_{v_m^{n_i}} = 1, \quad \forall s_i \in s, \quad i = 1, 2, \ldots, k \tag{5}$$

2) THE OBJECTIVE FUNCTION OF THE VNF MAPPING PROBLEM

The availability of VNF instances, as well as hardware, is also a key factor when mapping SFCs. Lower availability will lead to lower QoS for users and lower user satisfaction. So the reliability of service needs to be guaranteed under the requirements of securing time delay. The availability formula is proposed here. The availability is defined in this paper as the proportion of the running time of a component in a period to the total period. The total period is the sum of the running time and the maintenance time of machine.

$$C = \frac{Uptime}{Uptime + Downtime} \tag{6}$$

The *Uptime* indicates the running time of the equipment. The *Downtime* indicates the time used for equipment failure maintenance. This gives the availability of a single VNF, physical device or VM in the i-th sub-flow of the S-th SFC.

$$C_{f_a^{s_i}} = C_{n_i} = C_{v_m^{n_i}} = \frac{Uptime}{Uptime + Downtime} \tag{7}$$

In practice, the availability of different servers and VMs vary depending on their age or hardware quality, thus the availability of a SFC sub-chain under normal operation is:

$$C^s_{sfc} = \prod_{n_i\in N}\prod_{v_m^n\in V_n}\prod_{f_a^{s_i}\in F^{s_i}} C_{n_i}\cdot C_{v_m^{n_i}}\cdot \alpha^{f_a^{s_i}}_{v_m^{n_i}} \tag{8}$$

Among them, $\alpha^{f_a^{s_i}}_{v_m^{n_i}}$ is a binary decision variable, if it is 1, it means that $f_a^{s_i}$ is deployed on $v_m^{n_i}$, otherwise it is 0. C^s_{sfc} represents the availability obtained after deploying the s-th SFC request. C_{n_i} represents the availability of the i-th physical device. $C_{v_m^{n_i}}$ represents the availability of the m-th VMs on the i-th physical device. The availability of each sub-flow can be obtained by multiplying the availability of each VNF instance, the availability of the physical device hosting the VNF and the availability of the VMs on that server with the binary decision variable. The overall availability of that SFC can be obtained by multiplying the availability of each sub-flow. Because availability is a fraction less than one, so as the number of sub-chains increases it will lead to a decrease in availability.

For services that require high reliability and low latency, service providers need to ensure that the reliability of parallel SFC meets user requirements.

$$C^s_{sfc} = \prod_{s_i\in S} C^{s_i}_{sfc} \geq C_{order} \tag{9}$$

Because the reliability of each physical device and VM is different, the goal of the VNF mapping problem is to maximize the availability of SFC mapping.

$$D = MAX\ C^s_{sfc} \tag{10}$$

This objective function has to satisfy the constraints of (1) (2) (5) (9).

3) THE SOLUTION PROCESS OF VNF MAPPING PROBLEM

Expression (11) gives the fitness function needed for the SA algorithm.

$$fit = \frac{1}{C_{order}} \tag{11}$$

The value of fitness is the inverse of the minimum deployment availability required by the user. The solution algorithm taken in this paper for the VNF mapping problem is the SA algorithm in the heuristic algorithm. The hill-climbing algorithm is used in some of the papers that currently exist to solve the VNF deployment problem. But the simple hill-climbing algorithm never moves to a lower value, so the algorithm is prone to the problem of local convergence. Problems with local convergence may lead to incomplete solutions. The random hill-climbing algorithm applies random wandering by moving, it may complete the solution but it is not efficient. The focus of this paper is on parallel SFC deployment for the delay-sensitive class of services. The hill-climbing algorithm is the exact solution, but for the deployment problem of NP-hard problem, the algorithm is unrealistic to solve the NP-hard problems in effective time and within a large-scale scenario. So a heuristic algorithm SA is used here to solve the problem.

The authors of that paper [50] spoke about the main idea of the SA algorithm. Also known in the industry as a cooling process for metals. In other words, when solving an objective problem, we can use a heuristic algorithm to solve it by iterating over and over again. This iterative process makes smaller changes at each step of the process, then judges which of the newly obtained results is better than the previous one, if it is a better solution obtained, it is always accepted, as the hill-climbing algorithm. However, when we get a new result that is worse than the current one, we accept it with a certain probability expecting it to become better, but this probability decreases with time. This is where it differs from the hill-climbing algorithm, ensuring the diversity of results and avoiding local convergence.

The specific algorithm is shown in Algorithm 1. The algorithm terminates when it reaches the number of iterations.

Algorithm 1 VNF Deployment Algorithm - SA Algorithm

```
Input: physical network resources $G$, physical server resources, number of SFC sub-chains $I$, number of VNF instances in each sub-chain of SFC $m$, demand for each VNF instance, define $T$ as the number of iterations.
Output: VNF deployment scheme $C(t)$
the number of initialization iterations is $t = 0, a = 1$, $a$ is the number of the VNF, $i = 1$, $i$ is the number of sub-chain.
for $t = 0 : T$ do
    for $i = 1 : I$ do
        if $a \leq m$ then
            Generate the deployment scheme for each $VNF_a$ in each SFC sub-flow and store it in $v[a]$
        end if
        Store the deployment plan for each SFC sub-flow in the $s[i]$
    end for
    Generate the VNF deployment scheme, defined as $C(t)$, then compute the solution $D(t)$ corresponding to $C(t)$
    if $D(t+1) > D(t)$ then
        Assign the value of $(t+1)$ to $(t)$
    else
        Generate a random number r between [0, 1]
        while $r < fit$ do
            $D(t) \leftarrow D(t+1)$
        end while
    end if
    $C(t) \leftarrow D(t)$
end for
return VNF deployment scheme $C(t)$
```

D. VIRTUAL LINK MAPPING PROBLEM CONSTRUCTION

1) BINDING CONDITIONS

This section considers the energy consumption of routing nodes and switches in the link mapping when considering the more delay-sensitive service mapping SFCs. The multi-objective optimization problem proposed in this paper is solved using QGA to finally generate the link mapping scheme for parallel SFC. In the mapping problem of SFC, the aim is to find a reliable, low-blocking physical link while satisfying the link bandwidth constraints, delay constraints and minimizing the occupied network resources. Therefore, in this section, we solve the problem by using a heuristic optimization algorithm. Specifically, the network state in the physical network is first collected by the SDN controller. Then let the QGA perform an iterative solution. We give the fitness through the constructed multi-objective optimization problem, thus performing an iterative evolution in the algorithm, then implementing the chromosome update operation through the quantum revolving gate until we get the solution that we want to obtain.

$$\beta_{l_{ij}}^{l_{ab}^{s}} = \begin{cases} 1, & \text{if } l_{ab}^{s} \text{ is mapped to the physical link } l_{ij} \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

Using the binary variable β to judge whether the logical link between two adjacent VNFs in the S-th SFC sub-chain maps to the physical link between the physical devices carrying the two adjacent VNFs.

$$\beta_{l_{ij}}^{l_{ab}^{s_i}} = \begin{cases} 1, & \text{if } l_{ab}^{s_i} \text{ is mapped to the physical link } l_{ij} \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

The physical link consumes mainly bandwidth resources. So the bandwidth consumption of logical links of SFC sub-chains placed on the same physical link cannot exceed the remaining bandwidth capacity of this physical link.

$$\sum_{s_i \in S} \sum_{l_{ab}^{s_i} \in L^{s_i}} bw_{l_{ab}^{s_i}} \cdot \beta_{l_{ij}}^{l_{ab}^{s_i}} \leq \sum_{l_{ij} \in L, \quad l_{ab}^{s_i} \in L^{s_i}} B_{l_{ij}} \cdot \beta_{l_{ij}}^{l_{ab}^{s_i}} = B_{max} \tag{14}$$

In constraint (14), $bw_{l_{ab}^{s_i}}$ represents the link bandwidth consumption between f_a^s and f_b^s in the i-th sub-chain requested by the s-th SFC. $B_{l_{ij}}$ represents the value of the maximum physical link remaining bandwidth of the $l_{ab}^{s_i}$ mapped by l_{ij} at this time. To ensure the path integrity of the SFC request is not split, let $b = a + 1$ indicating that $f_b^{s_i}$ is the next connected VNF of $f_a^{s_i}$ of the VNF. The i-th sub-chain of this SFC has F VNFs and needs to satisfy the flow constraints:

$$\sum_{n_j \in N} \sum_{a=0}^{f_F^{s_i}} \beta_{l_{ij}}^{l_{ab}^{s_i}} - \beta_{l_{ji}}^{l_{ab}^{s_i}} = \begin{cases} 1, & n_i = f_0^{s_i} \\ -1, & n_i = f_{F+1}^{s_i} \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

The service latency of SFC is composed of two parts, which are the processing latency $t_{f_a^{s_i}}$ of the individual VNF of each SFC sub-chain mapped on the VM and the transmission latency τ_{s_i} of SFC on the physical network. The processing latency is represented by the ratio of the total computation Q required for each VNF to be processed to the computation speed A of the VM on the physical server mapped by this VNF:

$$t_{f_a^{s_i}} = \frac{Q_{f_a^{s_i}}}{A_{v_m^{n_i}}}, \quad \forall s_i \in s, \quad i = 1, 2, \ldots, k \tag{16}$$

In expression (16), $Q_{f_a^{s_i}}$ represents the amount of computation to process $f_a^{s_i}$, $A_{v_m^{n_i}}$ represents the computing speed of the VM $v_m^{n_i}$. $T_{f_a^{s_i}}$ for the processing delay of all VNFs on each sub-chain of SFC. $\alpha_{v_m^{n_i}}^{f_a^{s_i}}$ is a binary decision variable. It is 1, which means that $f_a^{s_i}$ is deployed on $v_m^{n_i}$, otherwise it is 0, then:

$$\begin{aligned} T_{F^{s_i}} &= \sum_{f_a^{s_i} \in F^{s_i}} \frac{Q_{f_a^{s_i}}}{A_{v_m^{n_i}}} \cdot \alpha_{v_m^{n_i}}^{f_a^{s_i}} \\ &= \sum_{f_a^{s_i} \in F^{s_i}} t_{f_a^{s_i}} \cdot \alpha_{v_m^{n_i}}^{f_a^{s_i}}, \quad \forall s_i \in s \quad i = 1, 2, \ldots, k \end{aligned} \tag{17}$$

The transmission delay of SFC is obtained by summing the transmission delay over the physical link mapped from the virtual link between two adjacent VNFs on the SFC to the physical link in the physical network, using $T_{E^{s_i}}$ to denote the transmission delay of the i-th sub-flow in the S-th SFC:

$$T_{E^{s_i}} = \sum_{l_{ab}^{s_i}\to l_{ij}\in E^{s_i}} \tau^{max}_{l_{ab}^{s_i}\to l_{ij}} \cdot \beta_{l_{ij}}^{l_{ab}^{s_i}}, \quad \forall s_i \in s \quad s \in SF \tag{18}$$

Using $\tau^{max}_{l_{ab}^{s_i}\to l_{ij}}$ denotes the maximum delay of the virtual link between VNFs a and b in the i-th sub-flow of the S-th SFC mapped to the physical link in the physical network by multiplying it with $\beta_{l_{ij}}^{l_{ab}^{s_i}}$, we can obtain the transmission delay of the i-th sub-flow in the S-th SFC. $\beta_{l_{ij}}^{l_{ab}^{s_i}}$ is a binary decision variable, if it is 1, it means that if l_{ab}^{s} is mapped to the physical link l_{ij}, otherwise it is 0.

Using T_{s_i} to denote the service delay of the i-th sub-flow in the S-th SFC, then:

$$T_{s_i} = T_{F^{s_i}} + T_{E^{s_i}}, \quad \forall s_i \in s, \quad s \in SF \tag{19}$$

Using T_{total} to denote the sum of the service delays of each sub-chain set S of a parallel SFC:

$$T_{total} = Max_{\forall s_i \in s,\ \ s\in SF}\{T_{s_i}\} \le T_{order} \tag{20}$$

The total delay must take into account the delay of the longest path in these sub-flows as the total delay.

2) MULTI-OBJECTIVE OPTIMIZATION PROBLEM CONSTRUCTION

Let the required network resources consumed by the sth SFC in $s \in SF$ be R. Because the path with fewer hops and smaller delay will consume fewer network resources. The resource consumption is related to the number of hops of the selected path as well as the delay, we let the path be P. The required network resource consumption $R(P)$ of this SFC path is defined as:

$$R\,(P) = h(P) \cdot T_{total} \cdot B_{max} \tag{21}$$

The worst-case network resource consumption of the path is used as the normalization criterion and is set to $MAX(R)$:

$$MAX\ (R\,(P)) = MAX\ (h\,(P)) \cdot MAX\ (T_{total}\ (P)) \cdot B_{max}\ (P) \tag{22}$$

The normalized resource consumption of this path P is: $\frac{R(P)}{MAX(R(P))}$

In the mapping process, not only the minimum resource consumption needs to be considered. It should also ensure the quality and reliability of user services for subsequent SFC requests. The load distribution in the network links should be ensured to be balanced. If the remaining load capacity in the chain is less after the virtual link mapping, it also indicates that the required bandwidth between the nodes is close to the available bandwidth of the links. At this time, the path is unsafe. So the load needs to be reasonably distributed to each link to improve the throughput. The link load distribution can be measured by the remaining bandwidth of the link. $B_{l_{ij}}$ denotes the remaining available bottleneck bandwidth of the physical link l_{ij}. $B^*_{l_{ij}}$ denotes the remaining available bottleneck bandwidth of physical link l_{ij} after mapping $l_{ab}^{s_i}$. The link utilization of physical node i to j can be expressed as:

$$U_{l_{ij}} = \frac{B_{l_{ij}} - B^*_{l_{ij}}}{B_{l_{ij}}}, \quad l_{ij} \in L \tag{23}$$

From (23), the more available bandwidth on the path, the safer it is. When $U_{l_{ij}}$ is close to 1, it means that $B_{l_{ij}} - B^*_{l_{ij}}$ is close to the remaining available bottleneck bandwidth $B_{l_{ij}}$ of the link. The required bandwidth between nodes i and j is close to the available bandwidth, so the path is unsafe. if this mapping request is allowed, it leads to other requests that cannot be mapped to this path. Therefore, when $B_{l_{ij}} - B^*_{l_{ij}}$ is much smaller than $B_{l_{ij}}$, this path is secure and allows for the deployment of all requests.

The corresponding mean value is:

$$\bar{U} = \frac{\sum_{l_{ij}\in L,\ \ l_{ab}^{s_i}\in L^{s_i}} U_{l_{ij}} \cdot \beta_{l_{ij}}^{l_{ab}^{s_i}}}{\sum_{l_{ij}\in L,\ \ l_{ab}^{s_i}\in L^{s_i}} \beta_{l_{ij}}^{l_{ab}^{s_i}}} \tag{24}$$

$\beta_{l_{ij}}^{l_{ab}^{s_i}}$ is a binary decision variable, if it is 1, it means that if l_{ab}^{s} is mapped to the physical link l_{ij}, otherwise it is 0. Then the link utilization variance is:

$$\delta^2 = \frac{\sum_{l_{ij}\in L,\ \ l_{ab}^{s_i}\in L^{s_i}} \left(U_{l_{ij}} - \bar{U}\right)^2 \cdot \beta_{l_{ij}}^{l_{ab}^{s_i}}}{\sum_{l_{ij}\in L,\ \ l_{ab}^{s_i}\in L^{s_i}} \beta_{l_{ij}}^{l_{ab}^{s_i}}} \tag{25}$$

$U_{l_{ij}}$ represents the link utilization between physical nodes i to j, $\bar{U}$ represents the mean of link utilization between nodes. Let the optimal link utilization variance be δ_g^2, then the link utilization is normalized to:

$$\frac{\delta^2}{\delta_g^2} \tag{26}$$

In this paper, during the link mapping process, not only the resource consumption and load distribution are considered but also the number of hops of the service flow through the routing nodes and the energy consumption. In the massive applications internet, the performance of energy consumption should not be ignored. The network structure formed by the router interconnection delivers requests to each functional body to perform service tasks. Router power consumption depends mainly on the type of equipment and the number of ports. The power consumption of a router can generally be expressed as follows:

$$P_{router} = \begin{cases} P_b + P_p \cdot n_p, & \text{Router is activated} \\ 0, & \text{otherwise} \end{cases} \tag{27}$$

where P_b denotes the power consumed by the router's base hardware. P_p denotes the power consumed by one port of the router. n_p denotes the number of ports on that router that are used. A parallel SFC needs to cross multiple routers of the underlying physical network, so the total power consumption of the router is expressed as:

$$P_{router}^{total} = \sum_{l_{ij}\in L,\ \ l_{ab}^{s_i}\in L^{s_i}} if\left(1,\ \beta_{l_{ij}}^{l_{ab}^{s_i}}\right)\cdot P_b + 2\cdot \sum_{l_{ij}\in L,\ \ l_{ab}^{s_i}\in L^{s_i}} \beta_{l_{ij}}^{l_{ab}^{s_i}} \cdot P_p \tag{28}$$

$$P_{switch} = \begin{cases} P_{b^*} + P_{p^*}\cdot n_{p^*}, & \text{Switch is powered up} \\ 0, & \text{otherwise} \end{cases} \tag{29}$$

where P_{b^*} denotes the power consumption consumed by the underlying hardware of the switch. P_{p^*} denotes the power consumption consumed by one port of the switch. n_{p^*} denotes the number of ports using the switch. A parallel SFCs may need to cross multiple switches of the underlying physical network, so the total switch power consumption is expressed:

$$P_{switch}^{total} = \sum_{l_{ij}\in L,\ \ l_{ab}^{s_i}\in L^{s_i}} if\left(1, \beta_{l_{ij}}^{l_{ab}^{s_i}}\right)\cdot P_{b^*} + 2\cdot \sum_{l_{ij}\in L,\ \ l_{ab}^{s_i}\in L^{s_i}} \beta_{l_{ij}}^{l_{ab}^{s_i}} \cdot P_{p^*} \tag{30}$$

where $if\left(1, \beta_{l_{ij}}^{l_{ab}^{s_i}}\right) \in \{0, 1\}$ is a binary decision variable that determines whether a link route of s_i passes through the underlying physical link l_{ij}, which determines whether the router through which the link passes is turned on. The summation indicates the number of routers passed through, in addition, two ports need to be turned on after through one router, $2\cdot\sum_{l_{ij}\in L,\ \ l_{ab}^{s_i}\in L^{s_i}} \beta_{l_{ij}}^{l_{ab}^{s_i}}$ denotes the number of opened ports. P_b,P_{b^*} and P_p, P_{p^*} are constants.

The total power consumption of the switch and router can be obtained from the above equation as:

$$P^{total} = P_{router}^{total} + P_{switch}^{total} \tag{31}$$

Using the router when running four ports as the normalization criterion, then we set it as $P_{router}^{total'}$, the total power consumption after normalization is: $\frac{P_{router}^{total}}{P_{router}^{total'}}$.

In the process of solving the SFC orchestration problem, user QoS performance metrics are important. In addition, minimizing resource consumption and cost while satisfying user requirements can improve user satisfaction. In this paper, we define cost as the sum of link resource consumption, link energy consumption and physical equipment cost.

$$Cost_{total} = R(P) + P^{total} + \sum_{f_a^{s_i}\in F^{s_i}} \alpha_{v_m^{n_i}}^{f_a^{s_i}}\cdot\left(cpu_{f_a^{s_i}} + mem_{f_a^{s_i}}\right)\ \ n_i\in N \tag{32}$$

The problem of link mapping is constructed to find an SFC deployment path that minimizes resource consumption, reduces router energy consumption and balances load distribution while satisfying the requirements of latency and bandwidth constraints. Obviously, this is a multi-objective optimization problem, since these three performances cannot be optimal at the same time, the given weights $\alpha+\beta+\gamma = 1$.

TABLE 5. Quantum bit string.

w_1	w_2	w_3	w_4	...	w_n

$$min\ \ Target = \alpha\cdot\frac{R(P)}{MAX(R(P))} + \beta\cdot\frac{\delta^2}{\delta_g^2} + \gamma\cdot\frac{P^{total}}{P^{total'}} \tag{33}$$

This multi-objective optimization problem satisfies the constraints of (14), (15) and (20).

3) MULTI-OBJECTIVE OPTIMIZATION PROBLEM SOLVING

In this section, we solve this problem by using QGA, we represent optional underlying physical paths for each segment of the virtual link in quantum bit (qubit) and the qubit string represents the set of mappable physical links for each small segment of the virtual link.

This paper uses QGA [51] to solve the multi-objective optimization problem. QGA is different from the binary representation of the chromosome of the GA. QGA is based on the principle of qubit and quantum superposition state in quantum science. We represent the chromosome with qubit encoding, which means that a gene in the chromosome is stored and expressed with qubit, the possible states of the gene are '0', '1', or superposition state of both. '0' and '1' denote($|0>$ and $|1>$), superposition state $|\varphi>= \alpha\cdot|0> +\beta\cdot|1>$, where α and β are satisfied a bunch of complex numbers satisfying $|\alpha|^2 + |\beta|^2 = 1$. $|\alpha|^2$ denotes the probability of collapse of the quantum state $|\varphi>$ to $|0>$. $|\beta|^2$ denotes the probability of collapse of the quantum state $|\varphi>$ to $|1>$. The probability amplitude of a qubit can be defined as $[\alpha\ \ \beta]^T$, in addition, a quantum chromosome consisting of m genes can be expressed as:

$$q = \begin{bmatrix}\alpha_1\ \alpha_2\ldots\alpha_m \\ \beta_1\ \beta_2\ldots\beta_m\end{bmatrix} \tag{34}$$

A system containing m qubits can represent 2^m states simultaneously. Because traditional GA usually utilizes numerical or symbolic representation of chromosomes, the relative evolutionary steps correspond to specific chromosomes. Therefore, the diversity of populations and the guaranteed convergence speed of the algorithm cannot be carried out simultaneously. However, under the QGA, chromosomes are represented using qubit, which has a random nature, the corresponding evolutionary operations are performed on the probability magnitude of gene states. Therefore, the convergence speed of the algorithm is guaranteed. The diversity of the population is also guaranteed when searching for the optimal solution. The quantum evolutionary operation can be operated by quantum rotation gate, the purpose of the evolutionary operation is to avoid premature convergence

TABLE 6. Quantum revolving door selection table.

x_i	b_i	$f(x_i) > f(b_i)$	$\triangle\theta_{ki}$	S(α_i,β_i)			
				$\alpha_i\beta_i > 0$	$\alpha_i\beta_i < 0$	$\alpha_i = 0$	$\beta_i = 0$
0	0	False	0	-	-	-	-
0	0	True	0	-	-	-	-
0	1	False	0	-	-	-	-
0	1	True	0.050π	-1	+1	0	±1
1	0	False	0.010π	-1	+1	0	±1
1	0	True	0.025π	+1	-1	±1	0
1	1	False	0.005π	+1	-1	±1	0
1	1	True	0.025π	+1	-1	±1	0

due to local search. The quantum evolutionary operation can change the superposition state of chromosomes to the optimal solution. QGA can perform individual mutation by quantum crossover between individuals, which is good to avoid local convergence.

Let the initial population be $Q(t) = \left\{q_1^t, q_2^t, \ldots, q_k^t\right\}$, t is the number of generations of population evolution, k is the number of population individuals. In this case, the number of total SFC sub-flows, i.e., the number of chromosomes, m is the number of genes in the chromosome. In this case, the number of underlying physical links that can be deployed per virtual link segment, each qubit string represents the set of physical links that can be deployed per small virtual link segment. The k-th chromosome of the t-th generation population can be expressed as:

$$q_k^t = \begin{bmatrix} \alpha_{k1}^t & \alpha_{k2}^t \ldots & \alpha_{km}^t \\ \beta_{k1}^t & \beta_{k2}^t \ldots & \beta_{km}^t \end{bmatrix} \tag{35}$$

Initialize all genes $[\alpha \;\; \beta]^T$ are set to $[\frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}]^T$.

QGA uses quantum rotation gates to maintain the diversity of chromosomal individual populations by changing the probability amplitude of quantum states. The key to the QGA method is the quantum rotation gate (Q-gate), which can usually be expressed as $U(\theta_i)$:

$$U\left(\theta_i\right) = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix} \tag{36}$$

The specific use of the Q-gate is as follows:

$$\begin{bmatrix} \alpha_{ki}^{t+1} \\ \beta_{ki}^{t+1} \end{bmatrix} = U\left(\theta_{ki}\right) \cdot \begin{bmatrix} \alpha_{ki}^t \\ \beta_{ki}^t \end{bmatrix} = \begin{bmatrix} \cos\theta_{ki} & -\sin\theta_{ki} \\ \sin\theta_{ki} & \cos\theta_{ki} \end{bmatrix} \begin{bmatrix} \alpha_{ki}^t \\ \beta_{ki}^t \end{bmatrix} \tag{37}$$

$[\alpha_i^t \;\; \beta_i^t]^T$ and $[\alpha_i^{t+1} \;\; \beta_i^{t+1}]^T$ denote the i-th quantum position of the chromosome numbered k at generation t and the i-th quantum position of the chromosome numbered k at generation $(t+1)$, respectively. θ_{ki} is defined as the rotation angle and takes the following values: $\theta_{ki} = S\left(\alpha_{ki}, \beta_{ki}\right) \cdot \triangle\theta_{ki}$, where $S\left(\alpha_{ki}, \beta_{ki}\right)$ and $\triangle\theta_{ki}$ denote the direction of rotation and the rotation angle, respectively. The direction of rotation determines the direction of convergence to the global optimum. The rotation angle affects the speed of convergence to avoid local convergence caused by premature convergence. Q-gate can be used depending on the problem itself selecting the appropriate value, this paper adopts a variant of the general selection strategy, as shown in Table 6.

As shown in the Table 6, x_i and b_i denote the $i-th$ position in this chromosome measurement and the $i-th$ position of the currently measured optimal chromosome, respectively. $f(x_i)$ and $f(b_i)$ respectively, denote the corresponding fitnesses. The algorithm starts by defining the chromosome population Q(t) with each q_k^t as a chromosome. The search begins with an initialized probability amplitude of $[\frac{1}{\sqrt{2}}\frac{1}{\sqrt{2}}]^T$. The adjustment strategy of Q-gate is to compare $f(x_i)$ measured by individual q_k^t in chromosome with $f(b_i)$, if $f(x_i) > f(b_i)$, the genes in q_k^t are adjusted so that the probability amplitude evolves in the direction favoring x_i. In this paper, we choose the adaptation degree $f = \frac{1}{Target}$, the larger the adaptation degree, the smaller the target function, the smaller the adaptation degree, then the larger the target function.

Define the state solution of the algorithm as $P(t) = \left\{p_1^t, p_2^t, \ldots, p_k^t\right\}$, P(t) is the state solution obtained by performing a random number determination for each individual in Q(t) measured in real-time. P(t) is obtained from the quantum collapse of Q(t). P(t) is represented as a binary string consisting of $\{0, 1\}$, randomly generated a random number $\{0, 1\}$. If it is less than the probability amplitude $|\alpha_i^t|^2$, the measurement is taken as 0, otherwise, it is taken as 1. The process of determining the optimal individual obtained by iterative calculation above are calculated using the value in P(t).

The QGA for the link mapping process is shown in Algorithm 2.

4) ROUTING POLICY

The routing model in this paper adopts the SRv6-based model, as can be seen from Figure 2. The sender gives the demand characteristics of the service flow (e.g., whether the service is a latency-sensitive class of service, which physical servers with VNF deployment need to be passed). The SDN controller collects the network state and hardware resources in the physical network. According to the service demand, the SDN controller gives the number of SFC shunts and goes for the mapping of SFCs according to the previous method. The optimal path assignment of the service flow through these physical servers is carried out according to the routing state and the distribution of routers in the physical network. We aim to find a reliable, low-blocking physical link under the premise of satisfying link bandwidth constraints and delay constraints. Access device 1 encapsulates the entire path into F-SL via SRv6, which based on the demand characteristics of the service flow at the sender and the path result from the SDN controller. The relay device performs forwarding according to F-SL. The access device 2 decapsulates the SRv6, then forwards the SFC service flow to the terminal device. Total calculation method reference Figure 6.

SRv6 encapsulates the SFC requirements with the number of SFC diversions and service flow forwarding paths, labels them with SRH, then sends them down to the physical network access side for SFC orchestration and data flow processing. Set the number of sub-flows, up to three, to avoid the high cost and low reliability of SFC service flows caused by shunting all the time. In addition, the number of sub-flows

Algorithm 2 Link Mapping Process Algorithm - QGA

Input: physical network resources G, routing resources in the physical network, request flow subflow Q(t) of SFC, define T as the number of iterations.

Output: Link Mapping Solution for SFC.

```
 1: Let $t = 0$, initialize the population $Q(t) = \{q_1^0, q_2^0, \ldots, q_k^0\}$ and initialize all chromosomal gene
    probability magnitudes to $\frac{1}{\sqrt{2}}$
 2: Quantum collapse of Q(t) generates the state solution $P(t) = \{p_1^t, p_2^t, \ldots, p_k^t\}$
 3: Modified state solution $P(t)$
 4: The corresponding adaptation value $f(P)$ is calculated for each state solution
 5: Cache records of the calculated optimal individuals and their corresponding adaptation values
 6: Iterative Evolution
 7: for $t = 0 : T$ do
 8:     Quantum collapse of $Q(t-1)$ to generate state solutions $P(t-1) = \left\{p_1^{t-1}, p_2^{t-1}, \ldots, p_k^{t-1}\right\}$
 9:     Modified state solution $P(t-1)$
10:     The corresponding adaptation value $f(P)$ is calculated for each state solution
11:     Determine the quantum gate rotation angle and rotation direction according to the rotation angle
        selection strategy in the Table 6
12:     Using the formula
        $\begin{bmatrix} \alpha_{ki}^{t+1} \\ \beta_{ki}^{t+1} \end{bmatrix} = \begin{bmatrix} \cos\theta_{ki} & -\sin\theta_{ki} \\ \sin\theta_{ki} & \cos\theta_{ki} \end{bmatrix} \begin{bmatrix} \alpha_{ki}^{t} \\ \beta_{ki}^{t} \end{bmatrix}$
        for iteration, the population is updated to obtain the evolved population Q(t) and the corresponding
        state solution P(t), then the corresponding fitness is found Compare the evolved optimal individuals
        and their corresponding adaptation values with the previous records, then cache the records if they are better
13:     if $t > T$ then
14:         exit the loop and output the result
15:     end if
16:     if the optimal result of the solution meets the constraints of the objective function then
17:         output the result
18:     else
19:         Request if the service request can be reduced, if so, return to step 6 until the end of
            the run, if not, output the request failed
20:     end if
21: end for
```

is set to avoid performance degradation and energy consumption increase caused by too many sub-flows. Excessive number of sub-flows will lead to a large increase in cost and energy consumption. When the SFC service flow has passed, the physical equipment bearing the VNF will automatically fall into a dormant state if it does not need to undertake other business services. The physical equipment will be turned on when the next service arrives.

V. SIMULATION

A. SIMULATION PARAMETERS

In this paper, in order to evaluate the performance of we proposed orchestration policies, we refer to the research of [16] that uses Mininet, a temporary simulation platform that allows us to deploy the required network environment topology and simulated NFV architectures, on a laptop configured with dual Intel®Xeon CPU 2.40GHz 4 cores and 16GB RAM. The results are analyzed by Origin. Mininet can create a virtual network with hosts, switches, controllers and links. Mininet also can provide an extensible python API for user network creation and experimentation, which we implement through the python programming language. In the experiment, we set parameters for the simulated NFV infrastructure, the physical network simulated in this paper consists of 32 physical servers with a spatial capacity of [2400,4800] Mbps and 20 switches, three VMs in each server to place VNF instances, a link bandwidth of 2000 Mbps between the physical servers, a virtual link bandwidth from a uniform distribution of U(3,7) Mbps is generated. In this paper, those flows from the same source node to the target node and fall into time slot t are considered as flows. The traffic within time slot t is the total traffic divided by the time slot interval. The packet length is 1024 bytes on average. The link delay is set to [0.1, 0.2] ms. The VNF availability, VM availability and physical server availability takes values in the range [0.9, 0.99]. The computational resource size of each physical server is 8000 MIPS. The computational cost required for each type of VNF deployment is 10 MIPS. For each type of VNF, the resource processing capacity of 1 MIPS is evenly distributed between [0.4,0.5] Mbps. The SFC request is randomly generated, the number of VNF instances in each SFC is randomly taken as [3, 6]. The data flow delay requirement of each SFC is randomly generated in [50ms, 200ms]. The bandwidth requirement of each SFC is randomly generated in [0.1, 1] Gbps. The service flow of SFC is randomly generated between [4-20]. The average load of the links in the physical network is 30%. The link transmission rate is 100Mbps. The basic power consumption of the router and the port power consumption are 250W and 15.4W, respectively. We set α, β, γ, are the weighting factor, $\alpha + \beta + \gamma = 1$ (α is defined as the maximum, the most important purpose is to minimize the resource consumption).

The relevant parameters of the QGA are shown in Table 7. The algorithm implementation process in this work involves the following steps, initializing the values, then performing the process of finding, iterating, storing and comparing, without a huge amount of data, then here we have mainly used our own algorithm to go ahead.

B. ALGORITHM SIMULATION

This paper extends the existing heuristic-based single feature single request (HSFHR) algorithm proposed in [10] and

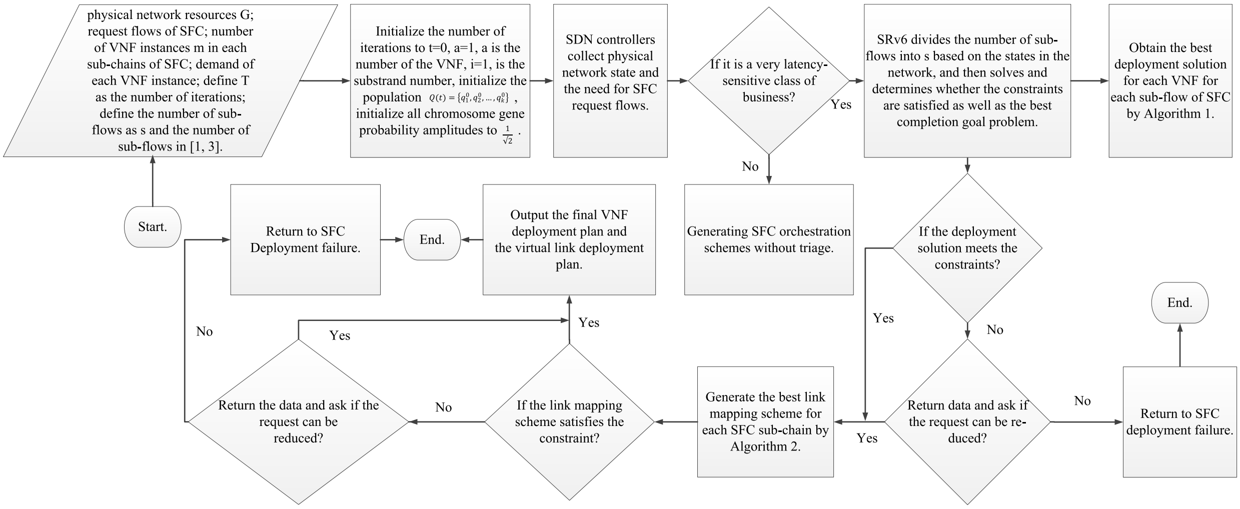

physical network resources G; request flows of SFC; number of VNF instances m in each sub-chains of SFC; demand of each VNF instance; define T as the number of iterations; define the number of sub-flows as s and the number of sub-flows in [1, 3].
Initialize the number of iterations to t=0, a=1, a is the number of the VNF, i=1, is the substrand number, initialize the population $Q(t) = \{q_1^0, q_2^0, \dots, q_k^0\}$, initialize all chromosome gene probability amplitudes to $\frac{1}{\sqrt{2}}$.
SDN controllers collect physical network state and the need for SFC request flows.
If it is a very latency-sensitive class of business?
Yes
SRv6 divides the number of sub-flows into s based on the states in the network, and then solves and determines whether the constraints are satisfied as well as the best completion goal problem.
Obtain the best deployment solution for each VNF for each sub-flow of SFC by Algorithm 1.
No
Start.
Return to SFC Deployment failure.
End.
Output the final VNF deployment plan and the virtual link deployment plan.
Generating SFC orchestration schemes without triage.
If the deployment solution meets the constraints?
End.
No
Yes
Yes
Yes
No
Return the data and ask if the request can be reduced?
No
If the link mapping scheme satisfies the constraint?
Generate the best link mapping scheme for each SFC sub-chain by Algorithm 2.
Yes
Return data and ask if the request can be re-duced?
No
Return to SFC deployment failure.

FIGURE 6. Deploy VNF with minimal resource utilization - MRU-VNF algorithm.

TABLE 7. QGA parameter.

Parameter	Value
Group size K	$0 \sim 100$
The number of iterations to terminate the evolution T	$30 \sim 50$
Initial value of $[\alpha_i^t \quad \beta_i^t]^T$	$[\frac{1}{\sqrt{2}} \; \frac{1}{\sqrt{2}}]^T$
Initial value of x_i	0
Initial value of b_i	0

compares it with the MRU-VNF algorithm proposed in this paper. The HSFHR algorithm relies on a single-path routing-based approach. This approach aims to search for the shortest path first, then VNFs that carry service flow requests along that path are deployed on the physical node with the highest availability, which minimizes the overall service latency. For better comparison with the algorithm in this paper, we have changed it. Changing this algorithm allows the service flow to be divided into sub-flow flows of equal size in an incremental manner until the delay is judged to have met the service demand or the delay is not further improved. We denote the changed version of the HSFHR algorithm as C-HSFHR. We also extend the Low-Latency algorithm to compare with our algorithm. The Low-Latency algorithm [52]: the algorithm works by setting up a new VM, then hosting all the VNF instances for each demand. To better compare with the algorithm in this paper, we extend the algorithm to D-Low-Latency that can be divided into sub-flow of equal size in an incremental manner until it is judged that the latency has met the service demand or the latency has not improved further and each sub-flow replicated VNF instances are placed in a new VM separately. Finally, we extend the Hill-Climbing algorithm as D-Hill Climbing to compare with the algorithm in this paper. The Hill-Climbing algorithm makes a judgment by searching the surrounding solutions with the current solution. If the new solution is better, the new solution is replaced with the current solution and vice versa. In order to better compare with the algorithm in this paper, so the extension of this algorithm can be divided into equal sizes in increments of sub-flow traffic until it is judged that the delay has met the service demand or the delay is not further improved.

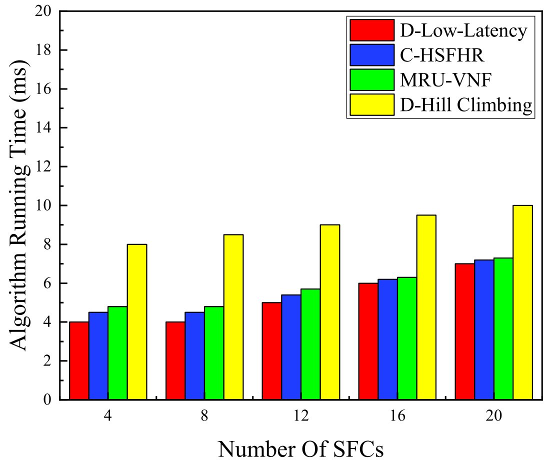


FIGURE 7. Running time of the algorithm with different number of SFCs.

As can be seen from Figure 7, the running time of the algorithm increases as the number of SFCs increases. Because the extended Low-Latency algorithm is to put all VNFs of each demand in a new VM, the VNF deployment is simpler, so the running time of this algorithm is lower. The extended C-HSFHR algorithm is the first to search for the shortest path, then find the highest resource available physical node on that path. The deployment of VNF instances is performed, but as the number of SFCs increases, the running time is close to that of the MRU-VNF algorithm proposed in this paper. The algorithm running time is slightly higher than the others, because the algorithm complexity of the MRU-VNF algorithm is slightly higher than the others. The SRv6 collection network situation with SA and QGA iterations are also considered in the algorithm. The extended hill-climbing algorithm can deploy SFC by finding the exact solution, but the running time is the longest.

As can be seen from Figure 8, the average packet delivery rate can reflect the goodness of SFC mapping and the reliability of deployment. In this paper, the average value is obtained through multiple simulations, the graph shows

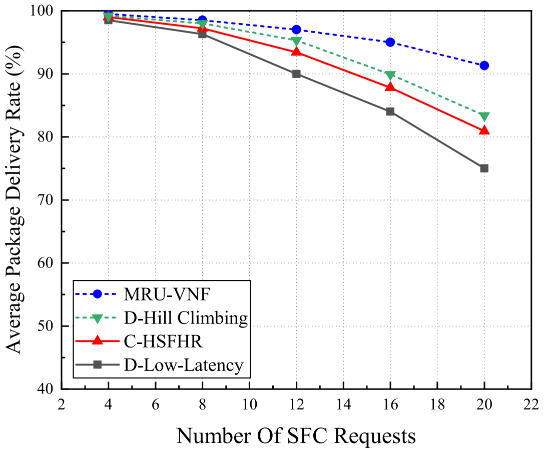


FIGURE 8. Average packet delivery rate with different number of SFC requests.

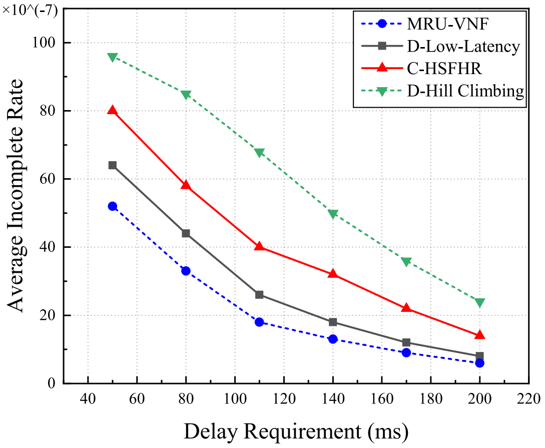


FIGURE 9. Average incomplete rate with different delay requirements.

that each algorithm has a decreasing packet delivery rate as the number of SFC requests increases, but the MRU-VNF algorithm has a smaller decrease in packet delivery rate and a higher packet delivery rate. The extended hill-climbing algorithm deploys SFCs by obtaining the exact solution, so the deployment nodes found are physical nodes with high reliability according to the problem formulation proposed in this paper. So the reliability is higher, but not as reliable as the MRU-VNF algorithm. The extended C-HSFHR algorithm is deployed on the shortest path, which tends to overload some of the links, resulting in low packet delivery rates. The extended Low-latency algorithm deploys all VNFs in the SFC sub-flow in a single VM, which takes longer to repair and has the worst reliability if a VM or server fails.

As seen in Figure 9, although the decrease is small, the average service incomplete rate decreases as the delay requirement continues to decrease. Specifically, the MRU-VNF algorithm combines the QGA mapping policy with the routing and forwarding advantages of SRv6 to perform better in this area of packet delivery rate, so the service has a lower uncompleted rate and higher reliability. The extended Low-Latency algorithm deploys VNFs in a physical server, but VM failure affects all VNFs redeployment, which is less reliable. However, for latency-sensitive services performance it is better in terms of latency, so the uncompleted rate is lower compared to the remaining two algorithms as the latency requirement increases. The C-HSFHR algorithm deploys through the shortest path, which tends to overburden the path, resulting in the packet non-completion rate will increase. The extended hill-climbing algorithm is based on greedy algorithm by step-by-step testing, so it is difficult to complete the packet delivery rate under high latency conditions.

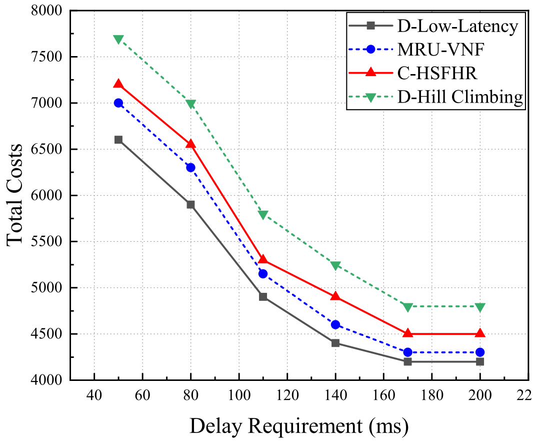


FIGURE 10. Total cost of ownership with different latency requirements SFC = 20.

With the increase of delay requirements, each algorithm for the selection of routing paths and routing nodes are more refined, because we need to find low latency, highly reliable routing paths. Because we consider the SFC parallel transmission, so the remaining three algorithms are extended. As can be seen from Figure 10, when the number of SFC request flows is 20, with the decrease of delay requirements, the total cost under each algorithm is decreasing. Especially at 110ms, the cost decreases to the greatest extent, it can be judged that, under the reduction of this delay requirement, the number of diversions is reduced, so it leads to a significant reduction in cost. Because of the remaining three algorithms to reach the goal of high latency requirements, so they also carry out shunting. Although they can achieve the latency target requirements, it is obvious to see that it leads to the cost is also increasing. For the extended Low-Latency algorithm, it puts the VNFs all on one physical server, so they do not occupy a large number of physical servers and routing devices. But if in the case of high delay, the MRU-VNF algorithm we propose in this paper is slightly inferior to the extended Low-Latency algorithm in terms of reducing operating costs. But in general, the difference is not very large and is acceptable, the gap between the two will continue to narrow as the latency requirement decreases. Therefore, MRU-VNF is able to meet the QoS of users while ensuring the profitability of operators. We will analyze the total cost under different numbers of diversions separately.

We define the average link load distribution rate in this paper as a value between 0 and 1. A smaller value indicates

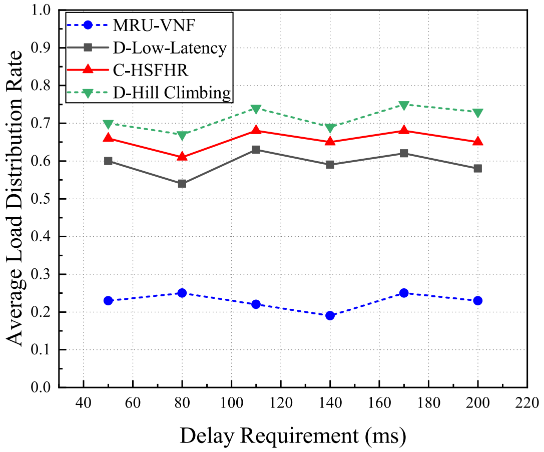


FIGURE 11. **Average link load distribution rate with different latency requirements.**

a more even load distribution on the average link. We simulated by randomly generating the number of SFC request flows under different delay conditions, and final averaged the obtained results in Figure 11. From Figure 11, we can see that the value of the average link load distribution rate is lower and does not fluctuate much as the delay requirement increases under the MRU-VNF algorithm. When constructing the multi-objective optimization problem, we take into account the load distribution, while combined with SRv6 which can collect the link load in the physical network. The algorithm can calculate the optimal routing path for the load distribution, so it performs better in the average link load distribution rate this performance. The C-HSFHR algorithm load is concentrated in the shortest path, which leads to the link load distribution in the whole physical network is not particularly balanced. The Low-Latency algorithm can also be effective in load distribution when solving the problem in this paper. The extended hill-climbing algorithm has the worst performance in load distribution. Figure 12 illustrates the relationship between link load distribution and time slot variation. It can be seen that the load distribution of we proposed MRU-VNF algorithm is more balanced compared to other algorithms. From the above analysis, it is clear that the performance of the MRU-VNF load distribution is better.

We map the virtual link by constructing the multi-objective optimization problem with the aim of minimizing the resource consumption in the physical link. In the simulation, the average value is obtained by randomly setting the different number of SFCs for 10 iterations to obtain the simulation Figure 13. By using different algorithms to solve the problem proposed in this paper, it can be seen that the MRU-VNF algorithm has the lowest average resource consumption and better performance. The algorithm can find more suitable nodes for VNF deployment by using multiple iterations. Through the global control capability of SRv6, the route forwarding path can be found faster, the resource consumption can be reduced compared to other algorithms in mapping SFCs. More detailed simulation of the energy performance is also available later.

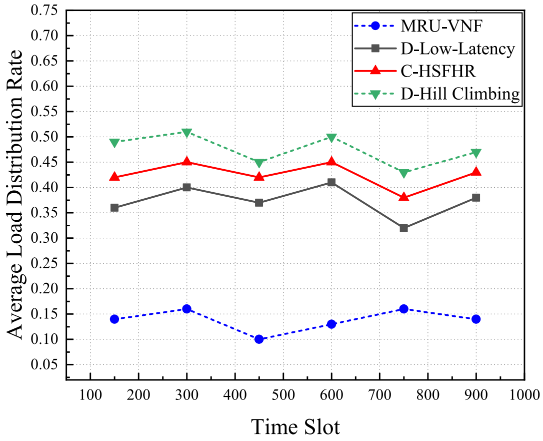


FIGURE 12. **Average link load distribution rate at different time slots.**

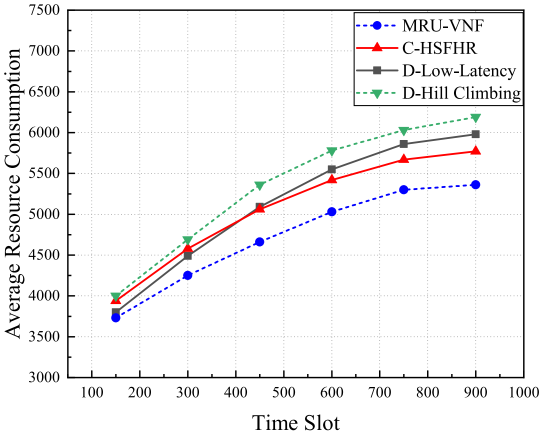


FIGURE 13. **Average resource consumption at different time slots.**

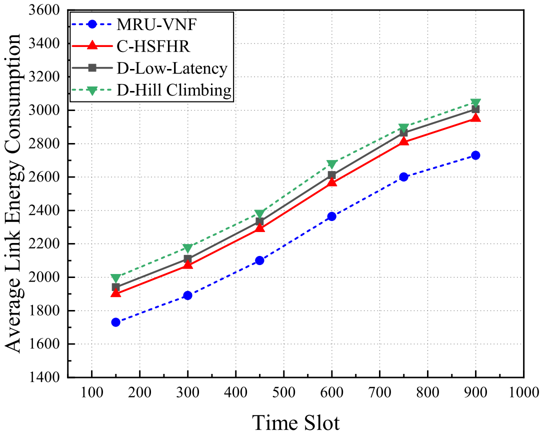
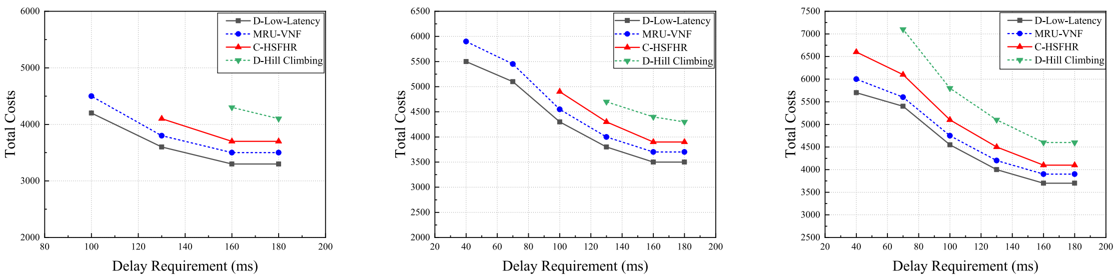


FIGURE 14. **Average link energy consumption at different time slots.**

From the above analysis, it is clear that an increase in the number of SFCs leads to an increase in the energy consumption of routers and switches. Next, we study the variation of average link energy consumption under the condition of the random number of SFCs. This is shown in Figure 14, We average the link energy consumption tested by ten times of randomly obtaining the number of SFCs to obtain this figure. We can see that the average link energy consumption to each algorithm increases as the time slot increases. As with the



(a) Number of sub-flows I=1 (b) Number of sub-flows I=2 (c) Number of sub-flows I=3

FIGURE 15. Total cost with different delay requirements.

previous resource consumption trend, the proposed MRU-VNF algorithm has lower routing energy consumption compared to the other algorithms. The C-HSFHR algorithm is second, we extend Low-Latency algorithm and hill-climbing algorithm, its routing energy consumption is more. The result computed by the MRU-VNF algorithm consumes less energy mainly because the MRU-VNF algorithm operates by collecting network state through SRv6. This algorithm considering the virtual link mapping and routing through the QGA, then deploying the most reasonable number of routing nodes through which the SFC request flows are routed to ensure the minimum energy consumption. Dealing with energy consumption is also reflected in our multi-objective optimization problem. We set reasonable weight values to consider routing energy consumption. A less energy-intensive approach to SFC deployment has a very important application in real-life, upholding the core concept of green network, also can well reduce the cost overhead of operators. In the above, we study the performance of different algorithms by analyzing the resource consumption and energy consumption. Next, we will simulate the total cost of consumption.

We did not fix the number of splits in our previous simulations, but the algorithm can be split according to the delay requirements. In Figure 15, we study the trend of the total cost with time delay for a number of SFC requests of 10 and a different fixed number of diversions. In this paper, the maximum number of flows is set to three. Because when the number of flows exceeds three, the service performance improvement decreases as the number of flows increases. An excessive number of flows can lead to significant energy consumption and cost, so the case of more than three flows is not considered in this paper. It can be seen that, as in Figure 10, the total cost decreases with decreasing delay requirements, the extended Low-Latency algorithm has the best performance in terms of total cost consumption. From Figure 15(a), which is case of no shunting. The C-HSFHR algorithm is a single-path routing method based on the shortest path, which finds the physical node with the highest resources in the shortest path for deployment. But it is not sufficiently considered in routing, which can easily cause excessive path loading around the node with high node centrality, which results in reduced transmission reliability and the need for redeployment in case of link failure, so it leads to a higher cost. The algorithm is difficult to accomplish the task without shunting when services with higher requirements than 110ms delay are generated. We extend the hill-climbing algorithm, which takes longer and is more extensive, is difficult to accomplish tasks with higher latency requirements than 160ms without shunting. From the latter two figures, it can be seen that different algorithms have advantages in solving delay-sensitive services under the shunt case. The other algorithms have certain advantages when they are extended to process the shunt case algorithm under high delay requirements. When I = 2, the C-HSFHR algorithm cannot meet services requiring delay within 100ms, but at I = 3, it can do well for high delay tasks, but it causes an increase in cost. It can be seen that the trend of the total cost incurred by each algorithm in the latter two figures is increasing, with the increase in the number of diversions, different algorithms can gradually meet the requirements at higher delay requirements. It can be seen that the transmission method proposed in this paper by dividing the data flow into multiple divisions and SFC parallel transmission is effective, which has practical significance in real-life scenarios.

VI. CONCLUSION

In this article, we consider the problem of multipath SFC orchestration in services with different sensitivity to latency. We have considered here splitting the SFC request flow, then multi-threading the transmission. Compared with other approaches that do not provide parallel SFC, the approach provided in this paper is more suitable for SFCs orchestration for services with high latency requirements. Even for latency-insensitive services, the method proposed in this paper is a good approach in solving the energy consumption problem and load balancing in the SFC mapping problem. In the part of VNF mapping to physical server nodes, we use a SA algorithm with adaptation degree values, then deploy the physical nodes with high availability by solving for multiple iterations to satisfy the constraints. The simulation part indicates that the deployment method can achieve good reliability requirements. In the part of virtual link to physical link mapping, we differ compared to other literature in that we consider the mapping problem in terms of the number of node hops passed by the SFC request flow. We construct the problem as a multi-objective optimization problem by considering the energy consumption of physical links and routers or switches. We use QGA to solve the problem, which can better achieve the parallel SFC scheduling problem with multiple paths through SRv6 global network control capability and network state collection capability as well as the unique SR segmentation capability. Finally, through experimental simulation analysis, it can be seen that the method proposed in this paper can deploy SFC with minimum resource consumption. Through comparison with other algorithms, it can be seen that the method proposed in this paper can better reduce the delay and cost when mapping SFC. In addition, our algorithm can reduce the routing energy consumption during routing while ensuring service reliability. This algorithm can not only reduce the cost and increase the profit of network operators but also has high practical application significance in real-life large-scale scenarios. The method proposed in this paper provides a solution for the future scheduling of SFC delay-sensitive class services.

Finally, we found in the related literature on SFC orchestration that the problem of parallel SFC orchestration can more effectively meet the needs of delay-sensitive services, this method can better improve service quality. Through the final simulation, we found that proposed algorithm will get a better solution than other similar algorithms, which will make the solution more accurate. For specific analysis, please refer to the detailed analysis of the simulation in Chapter 5. The advantage of this paper is that it can better reduce the transmission delay of the service, improve the reliability of the service, ensure the balanced distribution of the link load and reduce the cost of operators, which leads to improving the income. But this paper also has certain shortcomings. Service availability is regarded as reliability for VNF deployment. Although reliability will be improved to a certain extent. We can deploy more backup VNFs to ensure reliability, but it will lead to increased deployment costs and transmission delays. The increase is inconsistent with our main purpose. In the future, we will study how to add the deployment of virtual monitoring functions to the parallel SFC orchestration to increase reliability. Finally, our results show that our work has certain practical significance in delay-sensitive businesses, such as telemedicine, vehicle network and other fields. In our future work, we hope to consider the security of nodes in parallel SFC orchestration and cross-domain SFC orchestration by designing new architectures in conjunction with blockchain technology. We also hope to consider the VNF deployment location problem by constructing a more accurate payoff function in conjunction with game theory in the cloud and edge deployment VNF problem.

ACKNOWLEDGMENT

The authors would like to thank all referees for their invaluable insights and suggestions.

REFERENCES

[1] J. Zhou, W. Zhao, and S. Chen, "Dynamic network slice scaling assisted by prediction in 5G network," *IEEE Access*, vol. 8, pp. 133700–133712, 2020.

[2] J. García-Morales, M. C. Lucas-Estañ, and J. Gozalvez, "Latency-sensitive 5G RAN slicing for industry 4.0," *IEEE Access*, vol. 7, pp. 143139–143159, 2019.

[3] A. Mukherjee, P. Goswami, M. A. Khan, L. Manman, L. Yang, and P. Pillai, "Energy-efficient resource allocation strategy in massive IoT for industrial 6G applications," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5194–5201, Apr. 2021.

[4] Z. Zhang, C.-H. Lung, M. St-Hilaire, and I. Lambadaris, "An SDN-based caching decision policy for video caching in information-centric networking," *IEEE Trans. Multimedia*, vol. 22, no. 4, pp. 1069–1083, Apr. 2020.

[5] W. Qiao, Y. Liu, Y. Lu, X. Li, J. Yan, and Z. Yao, "A novel approach for service function chain embedding in cloud datacenter networks," *IEEE Commun. Lett.*, vol. 25, no. 4, pp. 1134–1138, Apr. 2021.

[6] P. L. Ventre, S. Salsano, M. Polverini, A. Cianfrani, A. Abdelsalam, C. Filsfils, P. Camarillo, and F. Clad, "Segment routing: A comprehensive survey of research activities, standardization efforts, and implementation results," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 182–221, 1st Quart., 2021.

[7] P. Loreti, A. Mayer, P. Lungaroni, F. Lombardo, C. Scarpitta, G. Sidoretti, L. Bracciale, M. Ferrari, S. Salsano, A. Abdelsalam, R. Gandhi, and C. Filsfils, "SRv6-PM: A cloud-native architecture for performance monitoring of SRv6 networks," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 1, pp. 611–626, Mar. 2021.

[8] A. Varasteh, B. Madiwalar, A. Van Bemten, W. Kellerer, and C. Mas-Machuca, "Holu: Power-aware and delay-constrained VNF placement and chaining," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 2, pp. 1524–1539, Jun. 2021.

[9] S. I. Kim and H. S. Kim, "A VNF placement method considering load and hop count in NFV environment," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Barcelona, Spain, Jan. 2020, pp. 707–712.

[10] N. Promwongsa, M. Abu-Lebdeh, S. Kianpisheh, F. Belqasmi, R. H. Glitho, H. Elbiaze, N. Crespi, and O. Alfandi, "Ensuring reliability and low cost when using a parallel VNF processing approach to embed delay-constrained slices," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 4, pp. 2226–2241, Dec. 2020.

[11] A. Baumgartner, V. S. Reddy, and T. Bauschert, "Combined virtual mobile core network function placement and topology optimization with latency bounds," in *Proc. 4th Eur. Workshop Softw. Defined Netw.*, Bilbao, Spain, Sep. 2015, pp. 97–102.

[12] R. Kang, F. He, and E. Oki, "Virtual network function allocation in service function chains using backups with availability schedule," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 4, pp. 4294–4310, Dec. 2021.

[13] Y. Chen and J. Wu, "Latency-efficient VNF deployment and path routing for reliable service chain," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 1, pp. 651–661, Jan. 2021.

[14] Y. Yuan, X. Mu, X. Shao, J. Ren, Y. Zhao, and Z. Wang, "Optimization of an auto drum fashioned brake using the elite opposition-based learning and chaotic K-best gravitational search strategy based grey wolf optimizer algorithm," *Appl. Soft Comput.*, vol. 123, Jul. 2022, Art. no. 108947.

[15] Y. Yongliang, R. Jianji, W. Shuo, W. Zhenxi, M. Xiaokai, and Z. Wu, "Alpine skiing optimization: A new bio-inspired optimization algorithm," *Adv. Eng. Softw.*, vol. 170, Aug. 2022, Art. no. 103158.

[16] M. Gharbaoui, S. Fichera, P. Castoldi, and B. Martini, "Network orchestrator for QoS-enabled service function chaining in reliable NFV/SDN infrastructure," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Bologna, Italy, Jul. 2017, pp. 1–5.

[17] M. Rost and S. Schmid, "Virtual network embedding approximations: Leveraging randomized rounding," *IEEE/ACM Trans. Netw.*, vol. 27, no. 5, pp. 2071–2084, Oct. 2019.

[18] L. Ruiz, R. J. Duran, I. de Miguel, N. Merayo, J. C. Aguado, P. Fernandez, R. M. Lorenzo, and E. J. Abril, "Joint VNF-provisioning and virtual topology design in 5G optical metro networks," in *Proc. 21st Int. Conf. Transparent Opt. Netw. (ICTON)*, Angers, France, Jul. 2019, pp. 1–4.

[19] L. Yala, P. A. Frangoudis, G. Lucarelli, and A. Ksentini, "Cost and availability aware resource allocation and virtual function placement for CDNaaS provision," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 4, pp. 1334–1348, Dec. 2018.

[20] T. Gao, X. Li, Y. Wu, W. Zou, S. Huang, M. Tornatore, and B. Mukherjee, "Cost-efficient VNF placement and scheduling in public cloud networks," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 4946–4959, Aug. 2020.

[21] Y. Zhang, B. Anwer, V. Gopalakrishnan, B. Han, J. Reich, A. Shaikh, and Z. Zhang, "ParaBox: Exploiting parallelism for virtual network functions in service chaining," in *Proc. Symp. SDN Res. (SOSR)*, Santa Clara, CA, USA, Apr. 2017, pp. 143–149.

[22] R. Gouareb, V. Friderikos, and A.-H. Aghvami, "Virtual network functions routing and placement for edge cloud latency minimization," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2346–2357, Oct. 2018.

[23] R. Cziva, C. Anagnostopoulos, and D. P. Pezaros, "Dynamic, latency-optimal vNF placement at the network edge," in *Proc. INFOCOM IEEE Conf. Comput. Commun.*, Honolulu, HI, USA, Apr. 2018, pp. 693–701.

[24] A. Gupta, B. Jaumard, M. Tornatore, and B. Mukherjee, "A scalable approach for service chain mapping with multiple SC instances in a wide-area network," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 529–541, Mar. 2018.

[25] L. Qu, C. Assi, M. J. Khabbaz, and Y. Ye, "Reliability-aware service function chaining with function decomposition and multipath routing," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 2, pp. 835–848, Jun. 2020.

[26] D. Li, P. Hong, K. Xue, and J. Pei, "Availability aware VNF deployment in datacenter through shared redundancy and multi-tenancy," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 4, pp. 1651–1664, Dec. 2019.

[27] M. Karimzadeh-Farshbafan, V. Shah-Mansouri, and D. Niyato, "A dynamic reliability-aware service placement for network function virtualization (NFV)," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 318–333, Feb. 2020.

[28] M. Huang, W. Liang, X. Shen, Y. Ma, and H. Kan, "Reliability-aware virtualized network function services provisioning in mobile edge computing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 11, pp. 2699–2713, Nov. 2020.

[29] S. Vassilaras, L. Gkatzikis, N. Liakopoulos, I. N. Stiakogiannakis, M. Qi, L. Shi, L. Liu, M. Debbah, and G. S. Paschos, "The algorithmic aspects of network slicing," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 112–119, Aug. 2017.

[30] J. G. Herrera and J. F. Botero, "Resource allocation in NFV: A comprehensive survey," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 3, pp. 518–532, Sep. 2016.

[31] A. Alleg, T. Ahmed, M. Mosbah, R. Riggio, and R. Boutaba, "Delay-aware VNF placement and chaining based on a flexible resource allocation approach," in *Proc. 13th Int. Conf. Netw. Service Manag. (CNSM)*, Tokyo, Japan, Nov. 2017, pp. 1–7.

[32] W. Guan, X. Wen, L. Wang, Z. Lu, and Y. Shen, "A service-oriented deployment policy of end-to-end network slicing based on complex network theory," *IEEE Access*, vol. 6, pp. 19691–19701, 2018.

[33] F. B. Jemaa, G. Pujolle, and M. Pariente, "QoS-aware VNF placement optimization in edge-central carrier cloud architecture," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1–7.

[34] D. B. Oljira, K.-J. Grinnemo, J. Taheri, and A. Brunstrom, "A model for QoS-aware VNF placement and provisioning," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Berlin, Germany, Nov. 2017, pp. 1–7.

[35] Y. T. Woldeyohannes, A. Mohammadkhan, K. K. Ramakrishnan, and Y. M. Jiang, "ClusPR: Balancing multiple objectives at scale for NFV resource allocation," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 4, pp. 1307–1321, Dec. 2018.

[36] L. Qu, C. Assi, K. Shaban, and M. J. Khabbaz, "A reliability-aware network service chain provisioning with delay guarantees in NFV-enabled enterprise datacenter networks," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 3, pp. 554–568, Sep. 2017.

[37] R. Guerzoni, Z. Despotovic, R. Trivisonno, and I. Vaishnavi, "Modeling reliability requirements in coordinated node and link mapping," in *Proc. IEEE 33rd Int. Symp. Reliable Distrib. Syst.*, Nara, Japan, Oct. 2014, pp. 321–330.

[38] T. Guo, N. Wang, K. Moessner, and R. Tafazolli, "Shared backup network provision for virtual network embedding," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kyoto, Japan, Jun. 2011, pp. 1–5.

[39] W. Yeow, C. Westphal, and U. C. Kozat, "Designing and embedding reliable virtual infrastructures," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 2, pp. 57–64, Apr. 2011.

[40] F. C. Chua, J. Ward, Y. Zhang, P. Sharma, and B. A. Huberman, "Stringer: Balancing latency and resource usage in service function chain provisioning," *IEEE Internet Comput.*, vol. 20, no. 6, pp. 22–31, Nov. 2016.

[41] N. Zhang, Y.-F. Liu, H. Farmanbar, T.-H. Chang, M. Hong, and Z.-Q. Luo, "Network slicing for service-oriented networks under resource constraints," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2512–2521, Nov. 2017.

[42] A. Baumgartner, T. Bauschert, A. M. Koster, and V. S. Reddy, "Optimisation models for robust and survivable network slice design: A comparative analysis," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Singapore, Dec. 2017, pp. 1–7.

[43] F. Carpio, S. Dhahri, and A. Jukan, "VNF placement with replication for Loac balancing in NFV networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, May 2017, pp. 1–6.

[44] A. Leivadeas, M. Falkner, I. Lambadaris, M. Ibnkahla, and G. Kesidis, "Balancing delay and cost in virtual network function placement and chaining," in *Proc. 4th IEEE Conf. Netw. Softwarization Workshops (NetSoft)*, Montreal, QC, Canada, Jun. 2018, pp. 237–241.

[45] G. Wang, G. Feng, W. Tan, S. Qin, R. Wen, and S. Sun, "Resource allocation for network slices in 5G with network resource pricing," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Singapore, Dec. 2017, pp. 1–6.

[46] L. Wang, Z. Lu, X. Wen, R. Knopp, and R. Gupta, "Joint optimization of service function chaining and resource allocation in network function virtualization," *IEEE Access*, vol. 4, pp. 8084–8094, 2016.

[47] C. Mouradian, S. Kianpisheh, M. Abu-Lebdeh, F. Ebrahimnezhad, N. T. Jahromi, and R. H. Glitho, "Application component placement in NFV-based hybrid cloud/fog systems with mobile fog nodes," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1130–1143, May 2019.

[48] M. Rost and S. Schmid, "Charting the complexity landscape of virtual network embeddings," in *Proc. IFIP Netw. Conf. (IFIP Networking) Workshops*, Zurich, Switzerland, May 2018, pp. 55–63.

[49] Y. Liu, X. Shang, and Y. Yang, "Joint SFC deployment and resource management in heterogeneous edge for latency minimization," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 8, pp. 2131–2143, Aug. 2021.

[50] S. Hua, G. Liangxian, and G. Chunlin, "Hybrid simulated annealing algorithm based on the parallel strategy," in *Proc. Int. Symp. Comput. Intell. Design*, Hangzhou, China, Oct. 2010, pp. 102–105.

[51] C. Niansheng, L. Layuan, and K. Zongwu, "QoS multicast routing algorithm based on QGA," in *Proc. IFIP Int. Conf. Netw. Parallel Comput. Workshops (NPC)*, Dalian, China, Sep. 2007, pp. 683–688.

[52] K. Yang, H. Zhang, and P. Hong, "Energy-aware service function placement for service function chaining in data centers," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1–6.

SONGLIN WEI received the B.S. degree in communication engineering from the North China Institute of Science and Technology, Hebei, China, in 2021. He is currently pursuing the M.S. degree in communications engineering with the Beijing Information Science and Technology University, Beijing, China. His research interests include resource allocation, SFC orchestration, and routing.

JINHE ZHOU received the B.S. and M.S. degrees in radio physics from Wuhan University, Hubei, China, in 1988 and 1991, respectively. He is currently a Professor with the School of Information and Communication Engineering, Beijing Information Science and Technology University. He has authored more than 50 articles. He hosted and participated in several scientific research projects, including the National Key Project of Hi-Tech Research and Development Program of China (973 Program) and the National Natural Science Foundation of China. His research interests include 5G networks, edge computing, game theory, and green information-centric networks. He received the Beijing Famous Teacher Award.

SHUO CHEN (Member, IEEE) received the Ph.D. degree in information and communication engineering from the Beijing University of Posts and Telecommunication (BUPT), in 2018. She is currently an Associate Professor with the School of Information and Communication Engineering, Information and Communication Engineering (BISTU). Her current research interests include wireless communications and networks, with an emphasis on resource management.