Routing optimization algorithm for logistics virtual monitoring based on VNF dynamic deployment

Qiujuan Qiao*
Shanxi Engineering Vocational College
Taiyuan, Shanxi 030032, China
[E-mail: jiuyao489915947@163.com]
*Corresponding author: Qiujuan Qiao

Received November 5, 2021; revised March 11, 2022; revised March 28, 2022; accepted April 23, 2022; published May 31, 2022

Abstract

In the development of logistics system, the breakthrough of important technologies such as technology platform for logistics information management and control is the key content of the study. Based on Javascript and JQuery, the logistics system realizes real-time monitoring, collection of historical status data, statistical analysis and display, intelligent recommendation and other functions. In order to strengthen the cooperation of warehouse storage, enhance the utilization rate of resources, and achieve the purpose of real-time and visual supervision of transportation equipment and cargo tracking, this paper studies the VNF dynamic deployment and SFC routing problem in the network load change scenario based on the logistics system. The BIP model is used to model the VNF dynamic deployment and routing problem. The optimization objective is to minimize the total cost overhead generated by each SFCR. Furthermore, the application of the SFC mapping algorithm in the routing topology solving problem is proposed. Based on the concept of relative cost and the idea of topology transformation, the SFC-map algorithm can efficiently complete the dynamic deployment of VNF and the routing calculation of SFC by using multi-layer graph. In the simulation platform based on the logistics system, the proposed algorithm is compared with VNF-DRA algorithm and Provision Traffic algorithm in the network receiving rate, throughput, path end-to-end delay, deployment number, running time and utilization rate. According to the test results, it is verified that the test results of the optimization algorithm in this paper are obviously improved compared with the comparison method, and it has higher practical application and promotion value.

Keywords: Logistics system, network node, routing topology, data stream coding, SFC-MAP algorithm

1. Introduction

Driven by the popularization of Internet technology and the technology represented by the Internet of Things technology, the annual growth rate of global Internet devices and connections is as high as 10%. The huge number of Internet users and network devices makes network services and applications more and more diversified. It is estimated that by 2023, the global mobile application downloads will reach nearly 300 billion times. Therefore, how to use more efficient network management, traffic engineering and resource allocation technology to cope with the rapid growth of the Internet community and diversified network services and application needs in the future has become an urgent problem for network operators. In view of the real-time geographical and equipment status data returned by logistics equipment, a fully distributed real-time tracking and data analysis and mining application system is developed. The system has the following functions: real-time monitoring, historical data statistics, mining recommendation, and user management. It also supports real-time processing and offline processing, which can meet the needs of the logistics industry for different business scenarios of massive logistics data [1].

In traditional networks, network operators will deploy a large number of middleware devices to provide users with various network services. Literature [2] pointed out that a small network with less than 1K hosts roughly needs to deploy 10 network functions, while for a super-large-scale network with more than 100K hosts, the number of network functions it contains will reach 2000 [3]. However, the high purchase price and complicated management configuration process of middleware realized by dedicated hardware equipment make it difficult to apply to flexible and large-scale network deployment scenarios. Network services implemented by hardware middleware often require management personnel with specialized business skills to operate and configure, and the function update of these middleware requires equipment manufacturers to provide support from the hardware level, which greatly extends the update of network services and the deployment period of new network services. To make matters worse, for middleware devices of the same category and produced by different manufacturers, there are often problems such as device interfaces, protocols, and operating platforms that are not uniform and universal, which make network operators need to deploy network services at the beginning determine the model and manufacturer of the middleware [4]. Therefore, the network service deployment model based on hardware middleware will cause serious equipment bundling problems for operators, which greatly increases the purchase cost of operators' equipment and the difficulty of later management and maintenance.

To solve the above problems, Network Function Virtualization (NFV) is born. Applying virtualization technology, NFV decouples network functions from dedicated hardware devices, which can allow users to provide diversified network services through software-based virtual network functions (VNF) [5]. Different from middleware implemented by dedicated hardware, virtual network functions are a series of software and applications that can be run on commercial servers, so VNF has the advantages of good flexibility and high scalability [6].

The paper uses virtualization technology, and NFV implements flexible deployment of routing topologies based on logistics systems in general hardware devices and different network locations in a software-based manner. With the flexible and easy-to-deploy feature of VNF, customized network services for users are provided by using SFC technology in SDN/NFV networks. However, in the actual network environment, the deployment of VNF directly affects the service quality of the network and SFC applications. An excellent VNF deployment strategy can achieve the purpose of balancing network load and improving network service quality through reasonable VNF deployment according to network load and resource conditions. Moreover, the deployment of VNF will also affect the routing calculation of SFC. Because SFC requires its path to pass through a series of specific types of VNFs in sequence. Therefore, the centralized deployment of various VNFs in the functional nodes of the transmission path can reduce the number of hops in the SFC path, reduce the bandwidth and cache resource overhead of the network stream in the transmission process, and optimize the end-to-end delay of the path.

The main contributions of the paper are:

(1) The main innovative work of this paper is to develop a real-time tracking and data analysis mining application system with fully distributed architecture for the real-time geographic and equipment status data returned by logistics equipment. The module has a fully distributed architecture that enables rapid scale-out.

(2) According to the characteristics of the data flow resource preference of the logistics system, a SFC routing algorithm matching network flow characteristic is proposed, which improves the network performance and realizes the network load balance.

(3) In view of the characteristics of dynamic changes in network load and the characteristics of flexible deployment of VNFs, SFC routing and VNF deployment scenarios are jointly considered, and the problem is mathematically modeled. Combined with relative costs, SFC mapping algorithms and VNF dynamic release algorithms are proposed, which can minimize the cost of both and improve the resource utilization rate of the network.

The content structure of the paper: Section 1 mainly elaborates the research background and significance of this thesis. Section 2 gives a brief introduction to the technology involved. Section 3 designs the routing topology based on the logistics data distribution system, describes the system based on the logistics network topology, and then models it based on the VNF design. The SFC mapping and VNF dynamic release algorithm are presented in Section 4. First, introduce the multi-layer map of the logistics network, and then use the SFC-MAP algorithm to optimize VNF deployment resources. Section 5 adopts the method of building a simulation experiment platform, and compares and tests the proposed method with other algorithms to verify the validity and optimization effect of the research content in the paper. Section 6 summarizes and looks forward to future work.

2 Related technologies

This section introduces the logistics network system based on the front-end framework Bootstrap, the VNF network topology and dynamic deployment, VNF-DRA algorithm and other related theoretical contents.

2.1 The front-end framework Bootstrap

Bootstrap, as a framework applied to front-end systems, aims to optimize the convenience of interface display while coding [21]. It can also provide a high-quality HTML and CSS usage rules, which respond positively to the overall layout, making the computer and mobile phone ports perfectly and error-free compatible. The highlight of this technology is that the mobile phone and PC can share a production version. We do not need to make efforts to create two versions to complete the practical significance of improving work efficiency. Bootstrap provides users with an extensible library. The document architecture performance of this library is good and the reading performance is simple. Other users can build or extend their own project library for this library. So far, it has included many components, compatible with responsive and non-responsive WEB design forms, has strong compatibility, and has been stable and accurate operation in many facilities. It can greatly improve the R & D efficiency of WEB developers and reduce the expenditure on application development. The number of employees using Bootstrap to develop software is increasing day by day.

Bootstrap can provide many reusable components for developers, including navigation, icons, click drop-down menu options, progress bars and other diverse needs. The components mentioned above all have interactive performance under the corresponding JQuery plug-ins. Using the above components can greatly improve the user's interactive experience and enhance the attractiveness of the project to users. The steps to use these components are particularly easy, requiring only three steps to complete the design:

The first step is to apply the basic HTML structure;

The second step is to call the corresponding JQuery plug-in;

The third step is to give the elements the corresponding properties they deserve.

Bootstrap has the following advantages:

(1) Expand that storage device and span a plurality of browser;

(2) Responsive structure layout: for example, the website we created can be perfectly compatible with PC, Android and Apple mobile devices to provide a more smooth and perfect user experience interface for a variety of users;

(3) Supply more detailed components;

(4) Configure JQuery plug-ins. JQuery components are built in Bootstrap. The above components allow program developers to use JQuery more conveniently, which is conducive to the development of the system;

(5) Support HTML5, CSS3 and other software development;

(6) Support the dynamic form of LESS. As a programming language with dynamic structure,

LESSCSs is one of the CSS preprocessing programming languages. This language has the special features of dynamic languages, such as variables, integration, calculation and so on, which makes the coding and later maintena

nce of CSS more convenient.

To meet the increasingly diverse needs of users, it has created and provided a set of customized services with its own characteristics, including CSS form customization, plug-in customization and JQuery component customization [24]. Users can independently adopt the form and performance they need, and delete redundant repetitive components so that the project they are responsible for can be further refined and simplified, and the code coding is more efficient.

2.2 Network topology based on VNF

The network topology is shown in **Fig. 1**.

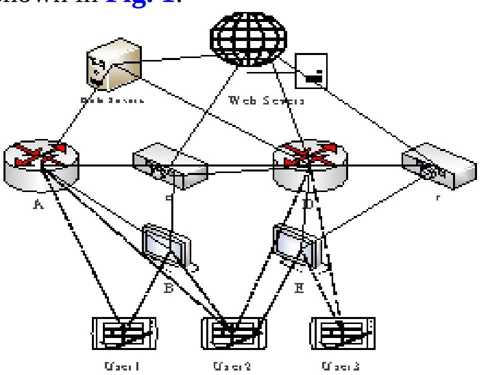

Fig. 1. Network Topology

Fig. 1 contains a total of switching nodes and functional nodes. In the functional nodes, VNFs are deployed, and each type of VNF is deployed in multiple instances in the network. Where VNF_a^1 and VNF_a^2 represent the first and second VNFI of VNF_a , respectively. The network flow generated by $SFCR_2$ needs to pass through the instances of ingress node C, VNF_a and VNF_b in sequence, and finally reach the egress node E. Both function nodes E and F deploy instances, and function nodes also allow deployment of these two types of VNFI [7]. To provide $SFCR_2$ users with the best quality network services, service providers need to design the best VNF deployment strategy to select or deploy VNF_a and VNF_b instances, and calculate the optimal transmission path of $SFCR_2$ [8].

In the SDN/NFV network environment with dynamic network load changes, there are two issues that need to be focused on:

(1) How to optimize the number of VNFI deployments and their deployment locations in the network.

(2) How to perform the optimal selection and routing calculation of VNFI in the SFCR routing process to achieve the goal of minimizing the total cost.

2.3 VNF dynamic deployment

VNF deployment and routing issues have become hot issues at the moment[9]. Most of the related work uses the Integer Linear Programming (ILP) model, Binary Integer Programming (BIL) model, Mixed Integer Programming (MILP) model and Mixed Integer Quadratically Constrained Program (MIQCP) model for problem modeling [10]. However, because the VNF deployment problem is a typical NP-hard problem, it is difficult to directly solve the optimal solution of the above model in large-scale network scenarios [11].

Aiming at the problem of VNF deployment for SFC service, literature [12] considers the energy consumption and data transmission cost of VNF and proposed a Markov approximation algorithm based on sampling. In this algorithm, the author uses the Markov transfer process to mathematically model the VNF deployment problem and makes the VNF deployment state in the network gradually converge to the optimal through each better action selection[13]. In order to improve the convergence speed of the algorithm, the author uses a many-to-one game model to model the VNF deployment problem and further proposes a matching mechanism. Literature [14] studies the VNF deployment problem of traffic awareness. According to the influence of VNF on the volume of traffic, VNF is divided into contraction VNF and expansion VNF. Then, under the premise that the network flow arrives one by one and the route is known, it is proved that the deployment method of deploying the shrinking VNF from the entrance of the path and deploying the expansion VNF at the exit of the path can obtain the optimal deployment results, and further the widest path routing algorithm is combined to realize the optimal deployment of VNFI in the network[15]. Literature [16] studies VNF deployment and optimization in multicast routing and spectrum allocation scenarios. The optimization goal is to minimize the use of optical fiber spectrum, VNF deployment costs, and IT resource overhead costs during the service process. The concepts of aggregation point and aggregation degree are introduced to measure the importance of nodes and optimize the use of spectrum. Then, the use of spectrum and 1T resources is further optimized based on the auxiliary frequency slot matrix. Finally, based on the auxiliary frequency slot matrix, a greedy search heuristic algorithm is proposed to solve the problem in polynomial time [17].

2.4 VNF-DRA algorithm

Network operators need to optimize the number of VNFI deployments to better adapt to dynamic network load changes. Although the SFC-MAP algorithm can dynamically deploy VNFI during the SFC routing process, the SFC-MAP algorithm does not consider the release of VNFI[18]. When the network changes from heavy load to light load, the over-deployed VNFI will increase the running time of VNFI, which will lead to a large amount of network resource occupation and energy consumption, and will also greatly reduce the utilization rate of VNFI [19]. Therefore, this paper proposes the VNF-DRA algorithm, which will efficiently release redundant VNFIs according to changes in network load.

The original intention of the VNF-DRA algorithm design is to reduce the total running time of the VNFI and achieve the purpose of reducing the total cost of the network operator by releasing the redundant VNFI with a low usage rate in the network according to the change of the network load [20].

The VNF-DRA algorithm is executed periodically, and its running period is: T. r_m^{cpu} is defined to represent the CPU resource availability rate of $\mathrm{VNFI_m}$ at time t, and $f\left(t\right)$ represents the VNF usage rate threshold at time f [21].When running the VNF-DRA algorithm, first, the CPU resource availability $r_m^{cpu}\left(t\right)$ of each VNFImeM is calculated in the network. Then, the VNF-DRA can redirect the SFCR on the VNFI, where the CPU usage rate is lower than the current threshold value of GFGF[22]. Finally, the VNF-DRA algorithm reduces its running time by closing the VNFI without SFC service .

3 Analysis of Intelligent logistics transportation system

3.1 Human-Computer Interaction

Human-Computer Interaction generally refers to the way users communicate with the operating system of a computer. At the same time, it is also the two-way information exchange of various symbols and actions between people and computer systems. The composition of human-computer interaction system requires software and hardware systems for communication between people and computers [23].

As the carrier of human-computer interaction based on Web, the important responsibility of creating a human-computer interface with perfect performance is to have the ability to provide users with sufficient information, simple and convenient interface, and finally a solid and reliable data operating system. To sum up, the selection of a Web-based human-computer interaction platform has the following infrastructure:

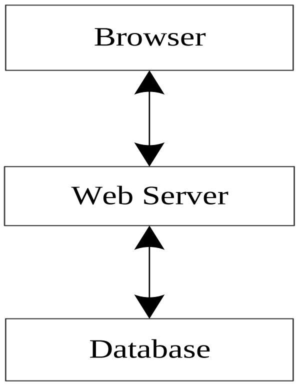

Fig. 2. Human-computer interaction platform based on Web

The information stored in the system by different users is not consistent, so the Web server must have the ability to obtain the information associated with the user from the database. At the same time, display the corresponding web interface in a dynamic form, and finally transmit it to the user's browser. Users also need to process the information of the Web server for the call application sent by the Web server, and then write it into the database.

3.2 Overall architecture of the system

This system provides a complete solution for logistics transportation, as shown in **Fig. 3**.

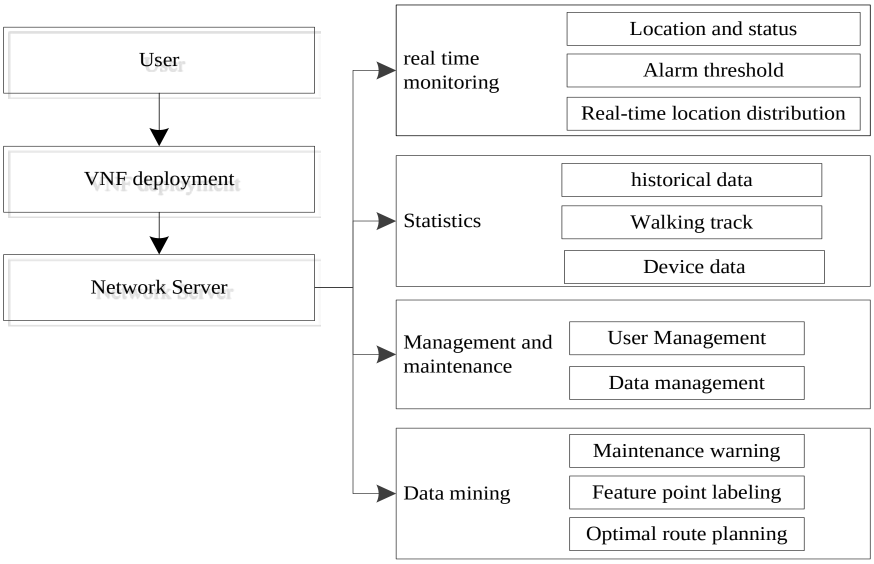

Fig. 3. Overall architecture of the system

The system has the following functions: real-time monitoring, data statistics, mining recommendation, management and maintenance. It also supports real-time processing and offline processing, which can meet the needs of the logistics industry for different business scenarios of massive logistics data. In view of the demand orientation of the three types of user groups, the final logistics big data platform has planned functional modules including real-time monitoring, data statistics, mining recommendation and user management.

4 Routing topology design based on logistics data distribution system

This paper uses the VNFI horizontal expansion mode, that is, a single function node that can deploy multiple instances of the same type of VNF. The available resources of each VNFI are fixed. The mathematical modeling of the VNF deployment and routing problem is expanded below[24].

4.1 System description

The logistics topology network is represented by the undirected graph GGG. In which V and f respectively represent the node collection and link collection of the physical network, and G represents the VNFI collection in the network.

4.2 System modeling

To provide network services for the newly arrived $SFCR_f$, formulas (1)-(3) ensure that the physical links, physical nodes and VNFI are included in the transmission path have enough bandwidth, cache and CPU resources [25].

$$\sum_{ur,ve} \phi_{ife}^{bw} z_{in}^{u_t} \le r_m^{bw} C_m^{bw} C_{ur}, E_{in} \tag{1}$$

$$\sum_{u_f v_f} \phi_f^{mem} f''^{u_f}_{z_m} \le r_m^{mem} C_m^{mem}, V_m e^{mem}, M \tag{2}$$

$$\sum_{u_{jf}} \phi_f^{cpu} f''^{u_f}_{z_m} \le r_m^{cpu} C_m^{cpu}, V_m e^{cpu}, M \tag{3}$$

Where, M is the VNFI set; P is the VNF category set; n_u is the number of VNFIs allowed to be deployed by the physical node; , $C_{uv}^{bw}, C_u^{mem}, C_m^{cpu}$ are the bandwidth capacity, cache capacity and CPU frequency of the physical link respectively; $r_{uv}^{bw}, r_u^{mem}, r_m^{cpu}$ are the bandwidth availability, cache and CPU availability of the physical link respectively; $C_{f,uv}^{bw}, C_{f,u}^{mem}, C_{f,m}^{cpu}$ are respectively that bandwidth, cache and CPU cost of the physical node; C_p^{place} is the deployment cost of the first type of VNF; θ_m is the set of VNFIs of the same type as $\mathrm{VNFI}_{m \in M}$; S_f, T_f are respectively the ingress node and the egress node; $\phi_f^{bw}, \phi_f^{mem}, \phi_f^{cpu}$ are respectively the bandwidth, cache and CPU resource requirements; and ϕ_f^{dekay} is the maximum tolerable delay.

Where $z_{uv}^{u_f, v_f}$, $z_m^{u_f}$ and $z_u^{u_f, v_f}$ are variables between 0-1, which are used to indicate whether the logical link $SFCR_f$ and logical node $u_f v_f \in q_f$ need to pass through the physical link $uv \in f$, physical node $u \in v$ and $\mathrm{VNFI}_{m \in M}$ if and only if they pass through corresponding to physical link, physical node and VNFI. If it passes, the value of the 0-1 interval variable is 1, otherwise the value is 0[26].

In the SDN/NFV network, the physical resources of each functional node are limited. Therefore, there is an upper limit on the number of VNFIs that can be deployed. Formula (4) ensures that the number of VNFI deployed on each functional node cannot exceed the maximum number of VNH deployments on that node:

$$\sum_{me,u} x_{tu} x_{f,m} \le n_u, ev \tag{4}$$

In formula (4), x_{tu} and $x_{f,m}$ are indicator variables, in which n_u indicates whether the $\mathrm{VNFI}_{m\in M}$ is allowed to be deployed on the physical node $u \in V$. When the physical node allows $\mathrm{VNFI}_{m\in M}$ to be deployed, the value of x_{tu} is 1, otherwise the value is 0. The indicator variable $x_{f,m}$ indicates whether it has been deployed after providing services to SFCR_f. When the $\mathrm{VNFI}_{m\in M}$ is already deployed, the $x_{f,m}$ value is 1, otherwise the value is 0. The product of x_{tu} and $x_{f,m}$ is used to indicate the deployment of $\mathrm{VNFI}_{m\in M}$ on the physical node $u \in V$, otherwise it means that it has not been successfully deployed on this node [27].

Different SFCR requires end-to-end path delays are often different. The end-to-end path delay is mainly composed of link delay, node delay and VNFI delay [28]. Formula (5) ensures that the total delay on the SF path cannot exceed the maximum tolerable delay of SJ:

$$\sum_{uv\in f}\sum_{u_f\in q_f} d_{uv} z_{uv}^{u_f v_f} + \sum_{u\in v}\sum_{u_f v_f\in q_f} d_u z_u^{u_f v_f} \le \phi_f^{delay} \tag{5}$$

The path guaranteed by formula (6) passes through the specific VNFI sequentially in accordance with its defined order constraint:

$$z_m^n y_n^n \le z_{uv}^n, \nabla v_f \in v_f \tag{6}$$

Formula (7) ensures that when a logical link passes through a physical link, the physical node $u_f v_f \in V_f$ connected to the physical link will also be passed through by the logical link:

$$z_f^{uf} z_v^{u_f v_f} = \begin{cases} 1, z_{uv}^{u_f v_f} = 1, \nabla u, v \in q \\ \quad 0, \text{otherwise} \end{cases} \tag{7}$$

Since a complete path is composed of several physical links, formula (8) guarantees that these physical links can be connected head-to-end to form a reasonable path in series[29], where s_f and t_f represent the entry node and exit node of SFCR_f, respectively:

$$\sum_{u\in v}\sum_{u_f\in v_f}\left(z_{uv}^{u_f v_f} - z_v^{uv}\right) = \begin{cases} 1, u = s_f \\ -1, u = t_f \\ 0, \text{otherwise} \end{cases} \tag{8}$$

Given that each VNF on SFCR can only be served by only one VNFI. Therefore, if the SFCR_f is successfully served, each VNF of the SFCR_f satisfies the above uniqueness constraint[30], which is expressed by formula (9):

$$\sum_{m\in M} z_m^{u_f} = 1, \nabla u_f \in v_f \tag{9}$$

Considering the dynamic change characteristics of network load, the number of VNFI deployed in the network also changes with the change of network load. Therefore, for new arrivals, it can not only use the VNFI that has been deployed in the current network, but also provide services for it by deploying a new VNFI [31]. Formula (10) guarantees the establishment of the constraint:

$$\sum_{u\in v}\sum_{m\in M} \frac{n_u y_u x_v y_v}{n}, y_{uv} = 1, \nabla y_f \{y, y_f\} \tag{10}$$

Formula (11) gives the unique constraint of the deployment location of VNFI, which guarantees that a VNFI can only be deployed on one functional node:

$$\sum_{u \in v} y_{uv} = 1, y_u e \in M \tag{11}$$

Formula (12) gives the uniqueness constraint of VNFI, where q_p^m is an indicator variable, which indicates whether $m \in M$ belongs to the $p \in P$ category of VNF, which is 1, otherwise it is 0[32]. This constraint ensures that a VNFI can only belong to one type of VNF:

$$\sum_{u \in v} q_p^m = 1, \nabla m \in M \tag{12}$$

The total cost of SFCR_f routing for successful service is composed of three parts: physical link bandwidth overhead, physical node cache overhead, and CPU resource overhead of VNFI. The total routing cost can be calculated by formula (13):

$$R = \sum_{uv \in f} \sum_{u_f \in y} c_{t,wal}^{v_n bw} + \sum_{u(v)} \sum_{ub} c_{v,th}^{mem_i t} + \sum_{u(v)} \sum_{ub} c_v^{cpuu_f} \tag{13}$$

In formulas (14)-(16), Ω_m represents the set of all VNFIs of the same category as $\mathrm{VNFI}_{m \in M}$:

$$c_{f,uv} = \frac{maxCC_w^{bw}}{r_{ub}^{bw} C_{uv}^{bw}} \tag{14}$$

$$c_{f,M}^{mem} = \frac{max_{u \in sn}}{r_u^{mem} c_u^{mem}}, V_{sn} \le V \tag{15}$$

$$c_{f_n}^{cpu} = \frac{\max_{n-\infty} c_u^{cpu}}{r_m^{cpu} c_m^{cpu}}, \Omega \le M \tag{16}$$

In the calculation of the cost of network resources (bandwidth, cache and CPU), the cost of logistics network resources is based on the reciprocal relationship of the available resources on its nodes, links or VNFI. The resource cost of this reciprocal relationship can be equivalently transformed into a monotonically increasing convex function form. Here, the resource cost is set as a monotonically increasing convex function, and its main purpose is to increase the resource cost of those devices with insufficient available resources quickly, so that the subsequent SFCR reduces the use of the device during the routing process. It prevents the device from forming a network bottleneck due to exhaustion of resources, which is beneficial to achieve load balancing. The amount of available cache resources of functional nodes (such as data centers, micro data centers, etc.) is generally much larger than that of switching nodes, which leads to a very large cost of switching nodes in the calculation of cache costs. To keep the cost of each part of formula (13) basically balanced, only the switching node V_{sn} is considered when calculating the node cache cost. For the functional node $u = V_{fn}$, the amount of available cache resources is approximately regarded as infinite, the cache cost value $c_{f,u}^{mem} = 1$ is defined, and the available resource rate is set to $r_u^{mem} = 1$.

The VNF deployment cost caused by SFCR_f can be calculated by formula (17), where c_p^{place} represents the deployment cost of the $p \in P$ for VNF:

$$D = \sum_{m \in M} \sum_{p \in P} c_p^{place} q_p^m max\{x_{j,m} - x_{f,m}, 0\} \tag{17}$$

To more clearly reflect the dynamic concept of deployment, a new indicator variable $\overline{x}_{f,m}$ is defined in formula (17), which represents the state of the $VNFI_{m\in M}$ before the SFCR_f is served.

In the paper, resource cost (including bandwidth, cache, and CPU) and VNF deployment cost are used to define the total cost of SFCR routing. When in a light load state, the available resources in the network are sufficient. So that in the total routing cost of SFCR, the resource overhead cost is much less than the deployment cost of VNF; SFCR reuses the VNFI deployed in the network during the routing process, thereby reducing the additional deployment cost caused by the deployment of new VNFIs. When under heavy load, the available resources in the network are reduced, which can lead to the resource overhead in the routing process of SFCR exceeding the cost of VNF deployment. It can be seen that the network urgently needs to deploy a new VNFI to increase the service capacity of the network. Therefore, deploying a new VNFI will bring lower costs and overhead than reusing the VNFI deployed in the network. In this way, under heavy load, the network will deploy many new VNFIs to balance the network load, thereby reducing the appearance of network bottlenecks. In summary, the optimization goal of the BIP model is to minimize the total cost of each SFCR routing process. The formula is:

$$\underset{Z_{uv}^{ufvf},Z_{u}^{ufvf},Z_{m}^{uf},x_f}{Minimize}\ R+D \tag{18}$$

As the network load increases, more and more VNFIs will be deployed in the network to provide more available resources for SFCR and balance the network load. However, when changing from a heavy load state to a light load state, the network will generate a lot of OPEX/CAPEX because of the need to maintain too much work and operation of the VNFI. Therefore, network operators need to focus on how to efficiently deploy and release VNFI during the SFCR routing process.

5 Proposed SFC mapping and VNF dynamic release algorithm

In the paper, the SFC-MAP algorithm is proposed to realize the optimal deployment of VNFI in the SFCR routing process, which can avoid the high calculation and time complexity problems caused by solving the mathematical analytical solution of the BIP model. SFC-MAP is an algorithm based on the idea of topology map conversion. It converts the physical network topology into a multi-layer graph according to SFCR, and then optimizes the deployment of VNF on the multi-layer graph and completes the routing calculation of SFCR. Besides, the changes in network load should be adapted. Meanwhile, the VNF-DRA algorithm releases redundant VNFI to reduce the total running time of VNFI.

5.1 Multi-layer diagram of logistics network

The multi-layer graph is formed by connecting several physical network topologies. In a multi-layer graph, each physical network topology is called a layer, and adjacent layers are connected by interlayer nodes and interlayer links. The layer corresponding to $SFCR_f : A \to VNF_a \to VNF_b \to H$ is given, as shown in **Fig. 4**.

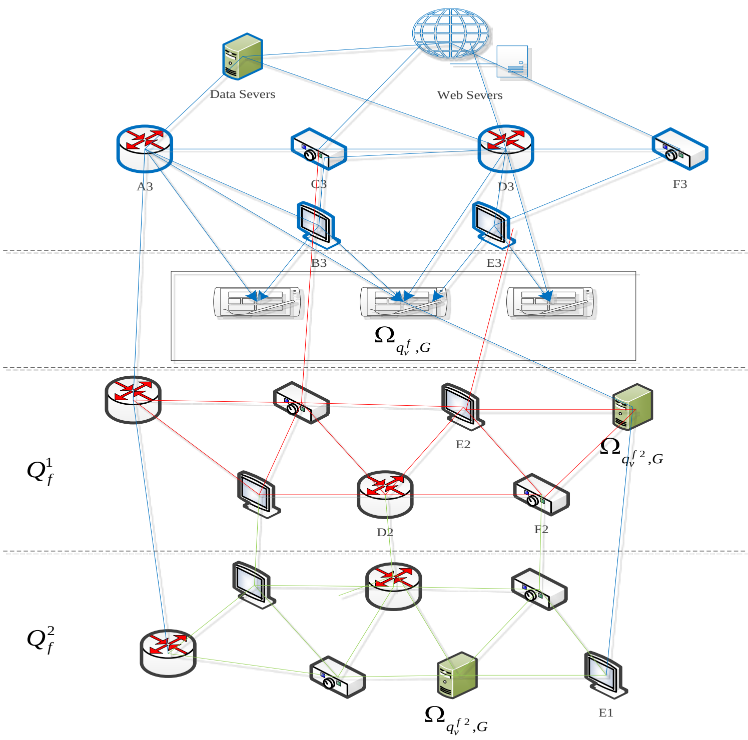

Fig. 4. Structure of Multi-layer graph

In the multi-layer graph, $Q^1_{j,u} \in M, L = 1, 2, \cdots + Np_f - 2$ is defined, which represents the set of all VNFIs deployed in the functional node $u \in V_{fn}$ and belonging to the $VNFQ^1_f$ category. Each set contains at most two elements. In which one element represents the VNFI with the lowest CPU cost belonging to the Q^1_f category on the functional node VNFI, and the other element represents the VNFI that belongs to Q^1_f but has not yet been deployed. According to formula (16), when the number of VNFIs deployed by the function node $u \in V_{fn}$ reaches the maximum value, $\Omega_{Q^1_{f,u}}$ only includes a VNFI with the lowest CPU cost that has been deployed; when the function node $u \in V_{fn}$ does not deploy VNFI, its corresponding $\Omega_{Q^1_{f,M}}$ set only includes an undeployed VNFI.

Containing multiple physical topologies are named layer 1, layer 2, ... in order. The subscript of each layer of physical nodes indicates the layer sequence number where the node is located. Node B in the physical network is represented as a pregnant node in the first layer of **Fig. 2** and a node in the second layer. Links in the same layer are called intra-layer links. The links and nodes used to connect functional nodes between two adjacent layers are called interlayer links and interlayer nodes, respectively.
The nodes between layers are selected and arranged in accordance with the order constraints of SFCR. In order to explain the inter-layer links more clearly, the set of VNF categories that need to be traversed is defined here as Q_f ($Q_f = Q^1_f, Q^2_f, \cdots, Q^l_f$), where Q^1_f represents the l type of VNF that $SFCR_j$ needs to traverse.

According to the definition, the QF set in **Fig. 2** contains two elements, where Q1F and Q2F are respectively corresponding to the first type of VNF and the second type of VNF that need to pass, namely, in the multi-layer graph, all the inter-layer nodes located between the l layer and the l +1 layer are represented by instances. For example, when l =1, the interlayer nodes between the 1st and 2nd layers in the multi-layer graph all represent 0, that is, the instance of the VN that the SF needs to pass through; when l =2, the inter-layer nodes between the 2nd and 3rd layers in the multi-layer graph all represent instances, that is, the instances that SF needs to pass next.
For the intra-layer nodes and intra-layer links in the multi-layer graph, the link bandwidth cost and node cache cost C are the same as their cost in the physical network topology, and both can be calculated by using formulas (14)-(15). The cost of inter-layer nodes includes VNF deployment cost and CPU cost, which can be calculated by formula (16). The inter-layer link in the multi-layer graph is mainly responsible for connecting the inter-layer node and the functional nodes of two adjacent layers, and the link cost is set to 0.

5.2 SFC-MAP algorithm

SFC-MAP is a VNF deployment and SFC routing algorithm based on multi-layer graphs. The entire SFC-MAP algorithm can be divided into 6 steps to execute, the process is shown in **Fig. 5**.

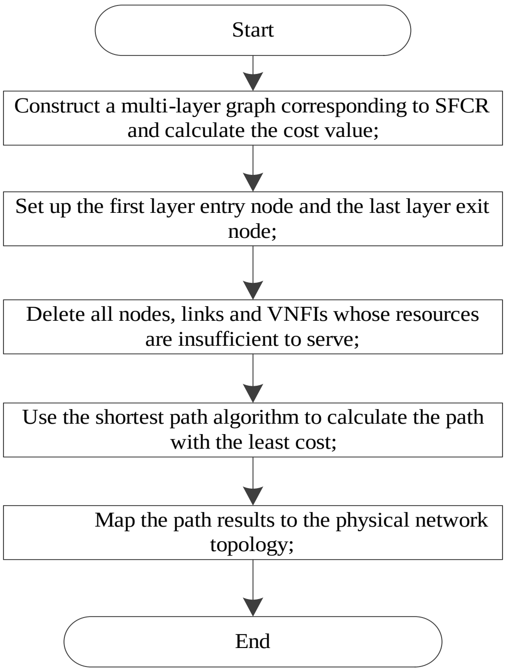

Fig. 5. Process of SFC-MAP algorithm

(1) The corresponding multi-layer graph is constructed according to SFCR and calculate the corresponding cost value;
(2) The entry node and exit node on the first and last layers are set, respectively;
(3) In each layer, all nodes, links, and VNFIs are deleted whose resources are insufficient to serve the SFCR;
(4) The shortest path algorithm is used to calculate the path with the least cost in the multi-layer graph;

(5) According to the mapping relationship between the physical network topology and the multilayer graph, the path results of the multilayer graph solution are mapped to the physical network topology;

(6) According to the path results of the physical network topology, the corresponding VNFI is deployed to complete the routing calculation.

For $SFCR_f : A \to VNF_a \to VNF_b \to Z$, the ingress node is A, which needs to be processed by VNF_a and VNF_b instances, and finally reaches the egress node Z. In **Fig. 3**, the entry node of $SFCR_f$ is denoted as A, and the exit node is denoted as Z. Supposed that the optimal path is $A_1 \to B_1 \to \Omega^1_{Q_f,b} \to B_2 \to \cdots \to Z_2$, which represents the VNFI corresponding to the first element in $\Omega^1_{Q_f,b}$, and the VNFI of the second element pair in $\Omega^2_{Q_f,b}$. $\Omega^1_{Q_f,b}$ has been deployed, but $\Omega^2_{Q_f,b}$ has not yet been deployed. Before serving, a new VNFI needs to be deployed on function node F. According to the mapping relationship between the physical network topology and the multilayer graph, the corresponding $SFCR_f$ path on the physical network topology: $A \to B \to \Omega^1_{Q_f,b} \to C \to \Omega^2_{Q_f,b} \to \cdots \to Z$.

When the SFC-MAP algorithm uses a multi-layer graph to solve the optimization goal, it may happen that the path obtained does not meet the resource constraints. Especially in the case of heavy load, the available resources of the calculated path node or link may not meet the resource requirements of SFCR, so it cannot successfully provide network services for it. The penalty factor δ is a value greater than 1. When the path calculated by the SFC-MAP algorithm does not meet the resource constraints of SFCR, the cost value corresponding to the node or link with insufficient resources on the path will be multiplied by the penalty factor δ . Through the use of the penalty factor δ , the cost of the bottleneck node or link in the network can be enlarged, and the possibility of the bottleneck being selected in the subsequent route calculation process is reduced. However, the introduction of the penalty factor δ will affect the calculation of the network resource cost. Therefore, the SFC-MAP algorithm cannot completely guarantee that the calculated path must be the least costly path in the network.

The pseudo code and description of the SFC-MAP algorithm are as follows:

Input: the dimension of $x_{s,k}$ is $|v| + \varepsilon + |k| + 2q + 3$.

The output $\tilde{y}_{s,k}$ is obtained according to the VNFI deployed in the network.

Number of initialization iterations is k = l;

Exclude the nodes, links and VNFIs whose resources obviously do not meet the conditions;

Cycle $\nabla uv \in \varepsilon, \nabla m \in m$: Execute:

Compute $c^{bw}_{f,uv}$, $c^{mem}_{f,u}$, $c^{cpu}_{f,m}$ in the physical network;

End;

Find the VNFI with the minimum CPU cost in each type of VNF in each function node;

Construct a multi-layer graph;

Cycle: The current iteration number k is less than the maximum iteration number limit K to execute: if: path exists Execute:

Convert that path from a multi-layer graph into a physical topology;

If: Path satisfies all constraints Execute:

Update $c_{f,uv}^{bw}$, $c_{f,u}^{mem}$, and $c_{f,m}^{cpu}$;
Deploy the VNFI on the corresponding function node of the path;
Use that path to provide network service for SFCRf;
Else:
Multiply the cost value of the resource in the path that does not satisfy the constraint by λ;
$k = k + 1$;
Jump to value line 8;
End.

The SFC-MAP algorithm excludes physical nodes, physical links, and VNFIs that obviously do not meet the resource requirements of SFCR based on the resource availability rate. Next, the cost of resources on physical nodes, physical links, and VNFI is calculated. After the SFC-MAP algorithm finds the interlayer nodes in the multi-layer graph, it further transforms the physical network topology into a multi-layer graph according to the cost of resources and the order constraints of SFCR. Then, SFC-MAP uses the shortest path algorithm to calculate the least costly path for SFCR on the multi-layer graph. Subsequently, SFC-MAP maps the path obtained on the multilayer graph to the physical network and checks whether the path meets the constraints. If the path meets all constraints, VNFI is deployed according to the path, and finally, the flow table is issued to provide services for SFCR[33]. If the path violates the resource constraint, the cost of resources of the bottleneck nodes and links on the path is multiplied by the penalty factor. *I* recalculates the path until it finds a path that satisfies the constraints or the number of iterations reaches the upper limit, then terminates the algorithm and refuses to accept the SFCR [34].

6 Performance evaluation of routing topology algorithm

The performance of the routing topology SFC-MAP algorithm is evaluated. First, this section will give the relevant parameter settings of the simulation environment. Then, through simulation experiments, the performance of the SFC-MAP algorithm and the existing algorithms are compared.

6.1 Simulation environment and parameter settings

The simulation test platform is used to evaluate the research method, and the platform configuration is shown in **Table 1**.

Table 1. Configuration of simulation test platform

Configuration	Parameter
CPU	Intel(R)Core(TM)i7-7900CPU4.50GHz
RAM	32GB RAM
Hard disk	1T SSD
System	Windows 10
Simulation platform	Matlab2018

The network topology adopts the network topology CORONETCONUS of the logistics distribution system, as shown in **Fig. 6**.

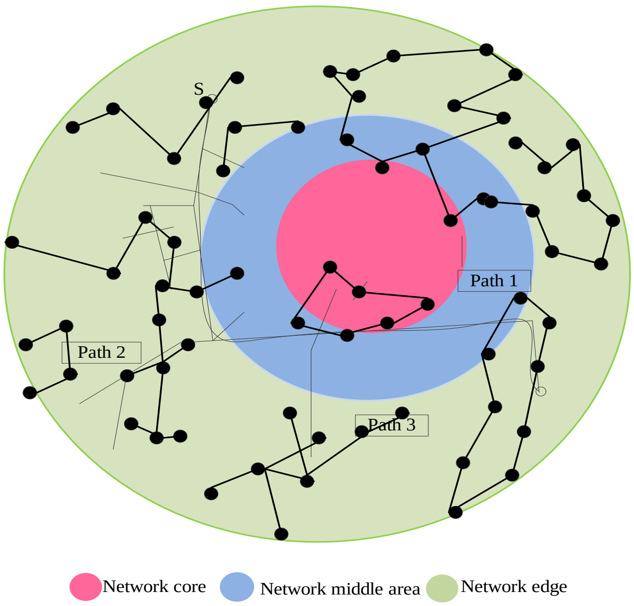

Fig. 6. Network topology of logistics area

In **Fig. 6**, there are 60 nodes and 110 links. In the network topology, 15 nodes with the largest degree are selected as functional nodes, and the rest are switching nodes. The model is widely used in 23 cities in China and has a certain universality. The network contains a total of 30 types of VNFs, each functional node is allowed to deploy 15 types of VNFs, and the total number of VNFIs that can be deployed by each functional node does not exceed 20. The deployment cost of each VNFI is $c_p^{place} = 50$. The bandwidth of each link is 500Mbps, the cache capacity of each switching node is 2000MB, and the CPU capacity of each VNFI is 100MIPS. The network delay setting formula is

$$d_{ax} = d_{bx}^{mp} + \frac{1}{dx} + \frac{1-r^{bx}}{r_{cyd}} dx \frac{dx}{dx_{cy}} \tag{19}$$

$$d_u = \frac{1-r^{n+m}}{r_n^m} - r_a^m, r_a, ue^n \cdots u_a e^n v_n \tag{20}$$

$$d_m = \frac{1-r_m^m}{r_m^m}, y_m e^{vec} x_n \tag{21}$$

In Formula19, d_{bx}^{mp} represents the propagation delay on the physical link $uv \in V$, which can be calculated based on the ratio of the link length to the transmission speed of the signal in the medium. d_x represents the transmission delay of the physical link $uv \in f$, which can be calculated by the ratio of the size of the data packet to the link bandwidth. r_{cyd} is used to calculate queuing delay, and its size is related to load and transmission delay. Formulas 20-21 are used to calculate the processing delay on the node and VNFI, where r_n^m and r_m^m represent the processing delay on the node $u \in V$ and $VNF\ \mathrm{I}_{m \in M}$, respectively. According to the queue model, in the case of low load, the queue delay and processing delay will increase approximately linearly as the load increases. However, when in a state of heavy network load, especially when the resource consumption is close to the resource capacity of the network device, they will be allocated a high price. It will help balance the network load and prevent these network devices from becoming a network bottleneck due to exhaustion of resources.

6.2 SFCR arrival rate test in simulation environment

In the simulation experiment, the arrival rate of SFCR is shown in **Fig. 7**.

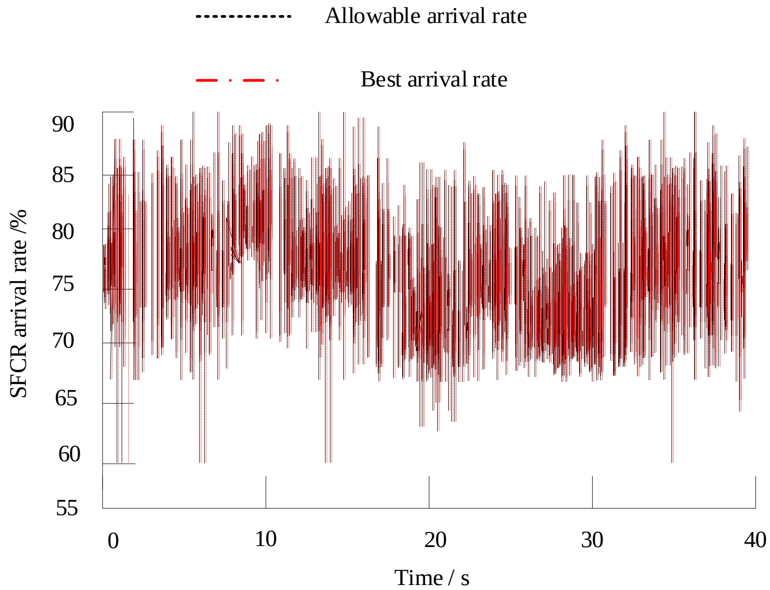

Fig. 7. The arrival rate of SFCR

The entrance and exit nodes of SFCR are randomly set. Each SFCR contains 3 types of VNF. The numerical size of bandwidth, cache and CPU resource demand of each SFCR meets the random distribution of (0,10). The maximum tolerable delay of each SFCR is between 50ms and 100ms. The lifetime of SFCR satisfies an exponential distribution with a mean value of 1000 time units (the time unit is a series of event trigger points set in the performance test experiment, and has nothing to do with actual time). Besides, in the SFC-MAP algorithm, the penalty factor δ >1 is set to 1.5. The algorithm is executed every 10s-time unit. The threshold p used to distinguish between long-lived and short-lived SFCR is set to 5-time units. In the threshold function f_t, the thresholds f_b and f_s are set to 50% and 20%, respectively. γ is used to judge whether the network load is fluctuating, and the threshold is set to $\gamma = 50Mbps$. Each group of simulation time is 15,000-time units, and each simulation result is subjected to 20 sets of simulation experiments.

6.3 Performance comparison of SFCR reception rate, network throughput and path end-to-end delay

In the comparison between the SFC-MAP algorithm and the VNF-DRA algorithm, the performance comparison of the SFCR reception rate is shown in **Fig. 8**.

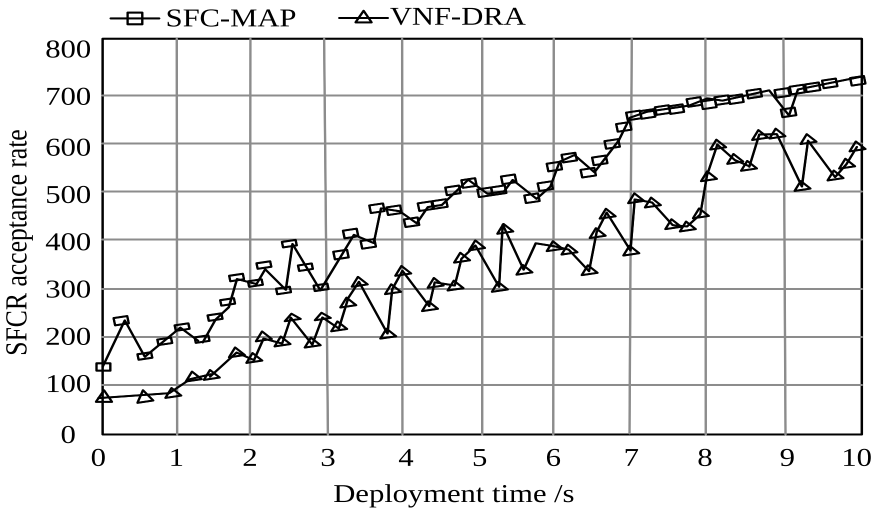

Fig. 8. Comparison of SFCR acceptance rate

According to the arrival rate of SFCR, a large amount of SFCR will be generated in a period of time, which causes the network load to be in a heavy load state. Under heavy load conditions, SFC-MAP obtains the best performance. In terms of SFCR reception rate, SFC-MAP is 16% higher than VNF-DRA algorithm.

The network throughput comparison is shown in **Fig. 9**.

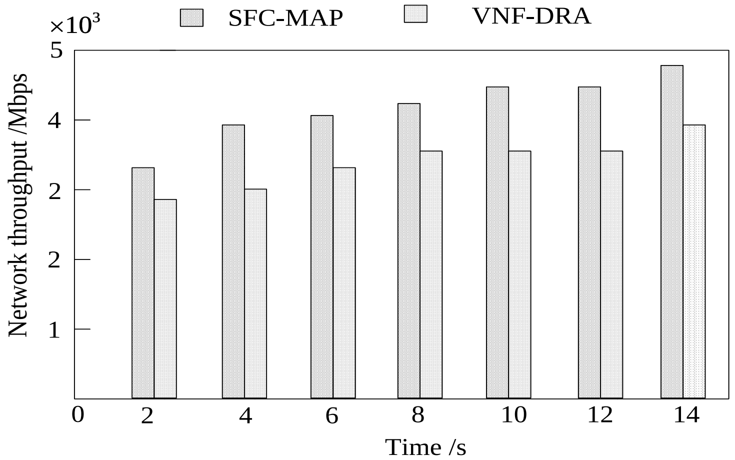

Fig. 9. Comparison of network throughput

SFC-MAP is 1000Mbps higher than VNF-DRA algorithm on average. The SFC-MAP algorithm comprehensively considers multiple types of network resources in the network, and can dynamically adjust and optimize the number of VNH deployments in the network according to changes in network load. The SFC-MAP algorithm achieves good network load balancing and improves network performance. The arrival rate of SFCR is low during the time period, and the network is under light load. Since the available resources in the network are relatively sufficient when the load is light, and network congestion is not prone to occur, the performance of the comparison algorithm in terms of SFCR reception rate has been improved.

The CDF curve of the end-to-end path delay distribution is shown in **Fig. 10**.

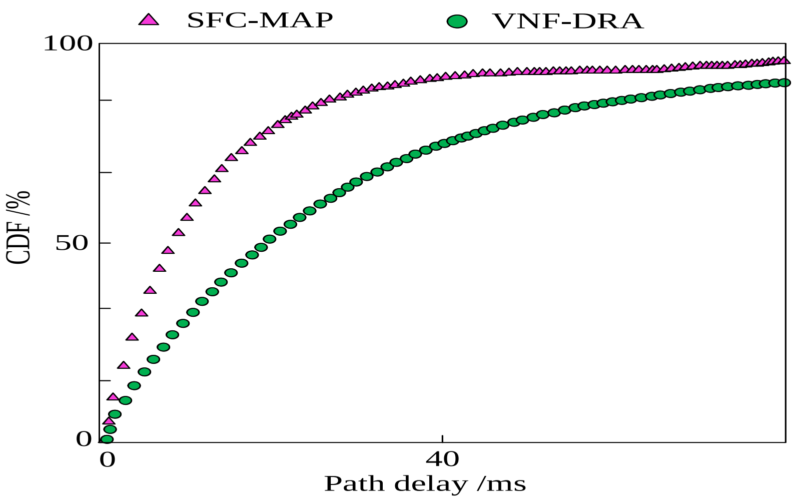

Fig. 10. Path end-to-end delay comparison

According to the simulation results, it can be seen that VNF-DRA has achieved the best performance, while the SFC-MAP algorithm is significantly better than the VNF-DRA algorithm. This is mainly because the path delay is optimized in the optimization objective of SFC-MAP, while the path delay is only considered as one of the constraint items in the VNF-DRA algorithm without optimization.

6.4 Performance comparison of VNFI deployment number, running time and utilization rate

The number of VNFI deployments is shown in **Fig. 11**.

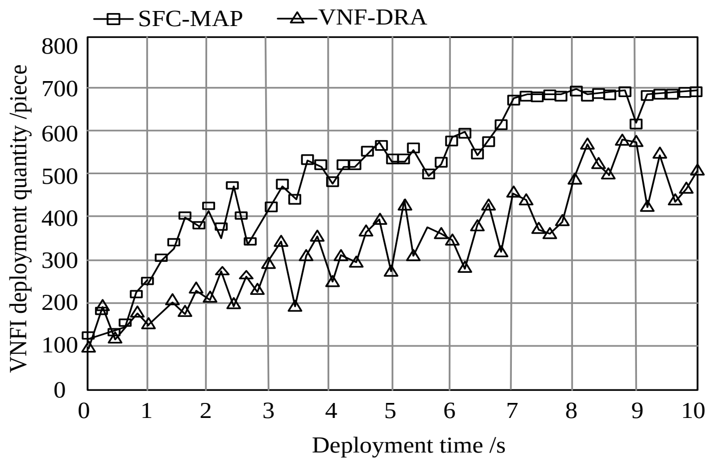

Fig. 11. The number of VNFI deployments

In **Fig. 9**, when in the $t \in [0,1]$ time period, the network load gradually decreases. Because the VNF-DRA algorithm can periodically perform VNFI release operations according to changes in network load, the number of VNFIs deployed in the network has shown a clear stepwise decline during the $t \in [2,3]$ time period. Therefore, the number of VNFI released by the comparison algorithm during the period of time when the network load drops is not large. As the network load increases during the $t \in [4,6]$ time period, the SFC-MAP algorithm will quickly deploy more VNFIs based on the network status to ensure that there are sufficient VNF resources in the network to provide network services for SFCR.

The total running time of VNFI in the period is shown in **Fig. 12**.

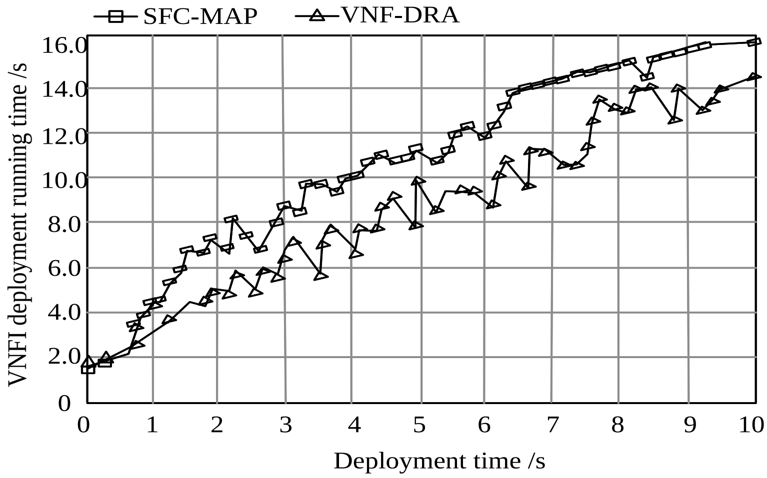

Fig. 12. Running time

Because the SFC-MAP algorithm can dynamically adjust the number of VNFI deployments following the network load. Therefore, the SFC-MAP algorithm has the shortest total running time of VNFI in one period, which reduces the total running time of VNFI by approximately 15% compared to the VNF-DRA algorithm.

The utilization rate of VNFI during the period is shown in **Fig. 13**.

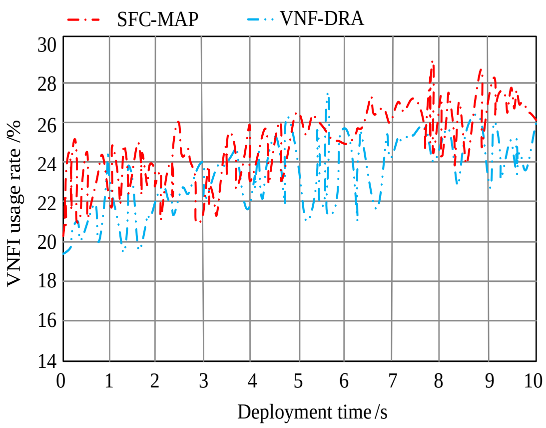

Fig. 13. Performance comparison of utilization

Through the optimization and adjustment of the number of VNFIs in the network, the SFC-MAP algorithm can maintain the utilization rate of VNFI above 60% regardless of the low load or high load, which is significantly better than the VNF-DRA algorithm.

6.5 Comparison of SFCR reception rate and the number of VNFI deployments in different scenarios

On the functional node, the comparison of the number of VNF categories allowed to be deployed is shown in **Fig. 14**.

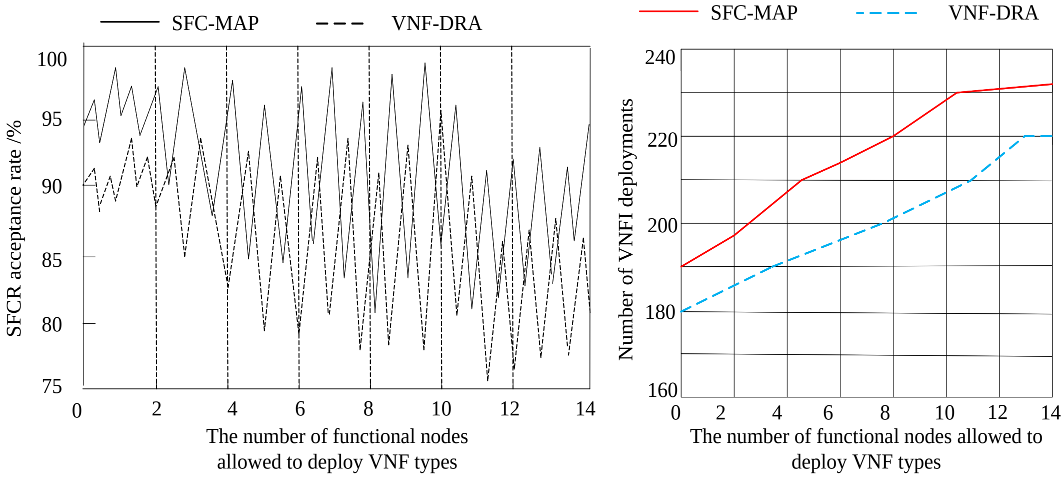

Fig. 14. The number of VNF categories allowed to be deployed on functional nodes

In **Fig. 14**, as the number of VNF categories allowed to be deployed in functional nodes increases, the SFCR reception rate will increase significantly. This is mainly due to the increase of VNF categories in the functional nodes, so that SFCR can meet the sequence requirements of VNFs through one or a few functional nodes, shorten the SFCR path, reduce network resource overhead, and enable more SFCR to be served. Besides, as the number of types of VNFs allowed to be deployed in functional nodes increases from 10 to 12, the number of VNFIs actually deployed in the network has a downward trend. This phenomenon shows that by increasing the number of VNF categories allowed to be deployed by functional nodes in the network, it is beneficial to improve network load balancing and increase network resource usage.

The algorithm performance of the CPU resource amount of a single VNFI is shown in **Fig. 15**.

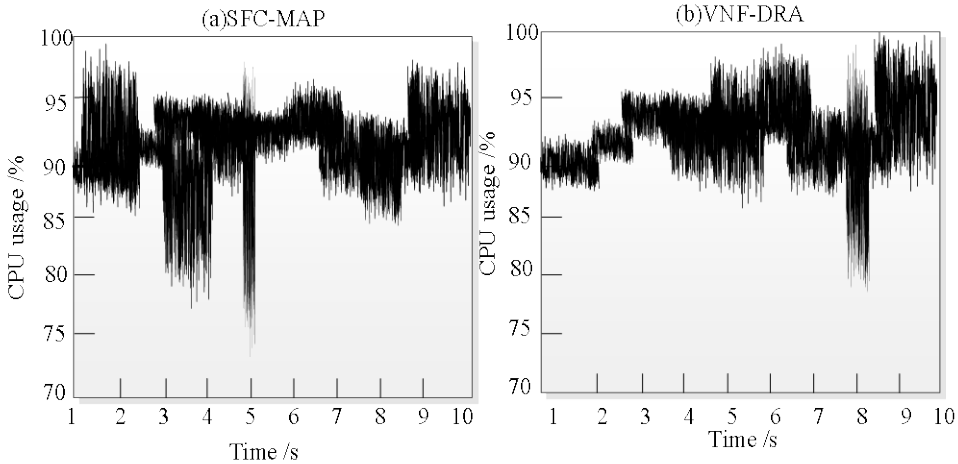

Fig. 15. The effect of CPU capacity on VNFI

In **Fig. 15**, as the amount of individual VNFI resources increases, the increase in SFCR reception rate gradually slows down, which shows that VNFI resources in the network are already sufficient, and link bandwidth and node cache resources have become the bottleneck restricting network performance.

The algorithm performance of the number of VNFs in SFCR is shown in **Fig. 16**.

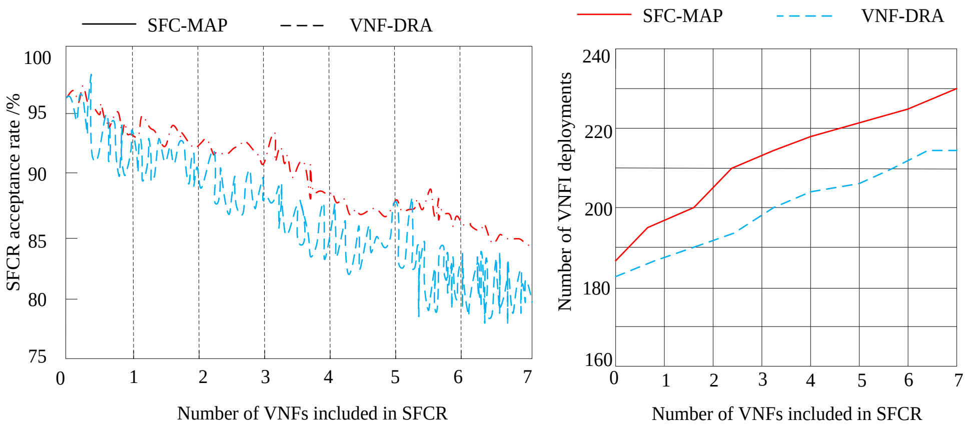

Fig. 16. The effect of the number of VNFs in SFCR

As the number of VNFs in SFCR increases, SFCR will consume more CPU resources on VNFI, so the number of VNFI deployments in the network will increase rapidly. Besides, as the number of VNFI deployments increases, the SFCR reception rate will continue to decline, which also shows that VNFI resources in the network have become a bottleneck restricting network performance.

On the functional node, the algorithm performance effect of the number of VNFIs allowed to be deployed is shown in **Fig. 17**.

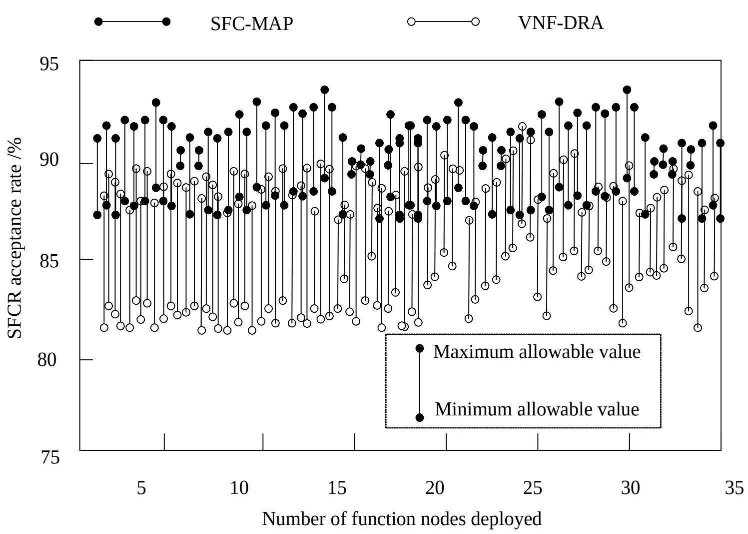

Fig. 17. The impact of the number of VNHs allowed to be deployed on a functional node

In **Fig. 17**, by increasing the number of allowed deployments of functional nodes VNFI, the SFCR reception rate is significantly improved. This is because the increase in the number of VNFIs on functional nodes makes the network have more resources for providing services. Moreover, the increase in the number of VNFIs on functional nodes enables SFCR to fulfill the requirements of VNF sequential processing in a single or a few functional nodes. This is beneficial to shorten the transmission path of SFCR, save network resource usage, and further improve the reception rate of SFCR.

6.6 Comparison of SFC-MAP algorithm and Provision Traffic algorithm

The Provision Traffic algorithm uses multi-stage graphs to implement VNFI deployment and SFC routing calculations. In order to realize the deployment of VNFI, Provision Traffic defines the concept of pseudo-VNFI. Pseudo VNFI actually represents some network resources, they can deploy any type of VNH as needed. For an SFCR, the Provision Traffic algorithm will select the corresponding VNFI or pseudo-VNFI from the network according to the VNF in the SFCR. Then, the selected VNFI or pseudo-VNFI will construct a multi-stage graph according to the order constraints of SFCR. In the multi-stage diagram, the cost of the logical link comprehensively considers the cost of VNF deployment, energy consumption, data transmission, service level agreement violation penalties, and resource fragmentation. Then, Provision Traffic uses the Viterbi algorithm to calculate the least costly path for SFCR in a multi-stage graph. Finally, the deployment of VNH and data transmission of SFCR are completed according to the obtained path.

As shown in **Fig. 18**, among the solutions calculated by SFC-MAP, the proportion of global solutions accounted for about 92.6%. Moreover, according to Formula 18, the average difference between the result obtained by the SFC-MAP algorithm and the result obtained by the Provision Traffic algorithm is about 1.2%.

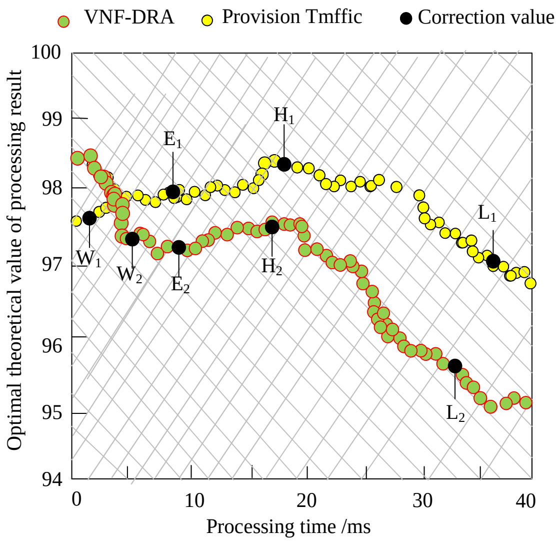

Fig. 18. Comparison of performance of different algorithms

The comparison result of the running speed of SFC-MAP algorithm and Provision Traffic algorithm is shown in **Fig. 19**.

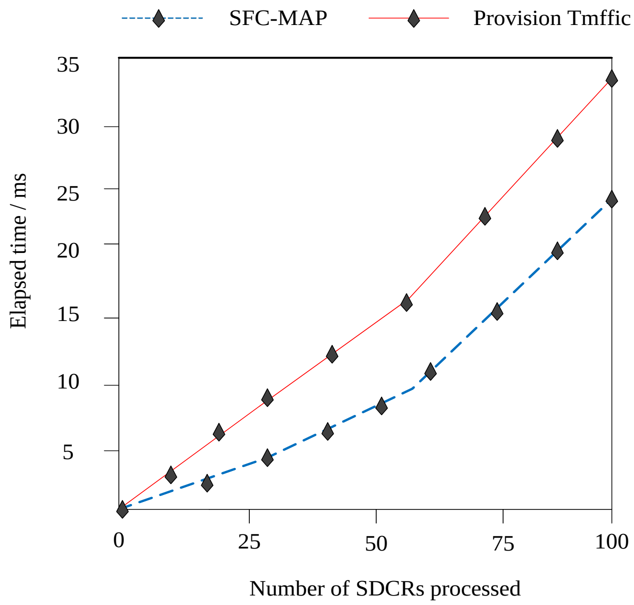

Fig. 19. Comparison of the speed of different algorithms

In **Fig. 19**, in the process of solving 100 SFCR VNF deployment and routing, the time used by the Provision Traffic algorithm is 2 orders of magnitude higher than that of the SFC-MAP algorithm. Therefore, in the regional logistics network, the SFC-MAP algorithm proposed in this paper can not only obtain near-optimal performance, but also its computational timeliness has been significantly improved, and it is also more suitable for dynamic deployment of VNFs in large-scale network scenarios. And routing problem solving.

7 Conclusion

In order to achieve fine-grained traffic management, according to the different resource preference characteristics of network flows, the paper starts from the perspective of network flow, the network flow is divided by bandwidth and computational overhead. Subsequently, in view of the characteristics of multi-category network resources, VNF multi-instance and flexible deployment in the SDN/NFV network, the SFC routing problem is mathematically modeled, and the optimization goal is to optimize the network resource cost of each SFCR. An SFC-MAP algorithm based on multi-layer graph is proposed, which can complete SFCR routing calculation and optimized deployment of VNFI at the same time. Considering that when the network changes from a heavy load to a light load, the over-deployed VNFI will generate high operation and cost overhead.

The SFC-MAP algorithm is compared with the VNF-DRA algorithm and the Provision Traffic algorithm. SFC-map achieves 5.6% improvement in arrival rate contrast and 16% improvement in SFCR acceptance rate. The utilization rate can be maintained above 26%; Resource utilization decreased by 3.4%, 2.3% and 4.2%, respectively. The evaluation results show that SFC-MAP can significantly improve network performance and resource utilization in the network, and can adjust and optimize the deployment of VNFI in the network according to load changes.

In the future, further improvement and research are needed:

(1) In the research of SFC routing based on network flow feature matching, this research

point assumes that the type of SFC network flow is known, and the characteristics of the network flow should be combined to achieve differentiated SFC routing calculations.

(2) In the research of VNF dynamic deployment based on network load adaptation, because the network load is dynamically changing, there is a certain fluctuation in network load within a short period of time.

Acknowledgement

This paper is supported by ″Internet plus background research and application of logistics professional virtual simulation training teaching of Shanxi Education Science 13th Five-Year plan 2020 special topic of Internet plus education, which number is HLW-20187″.

References

[1] A. Ghasempour, "Internet of Things in Smart Grid: Architecture, Applications, Services, Key Technologies, and Challenges," *Inventions*, vol. 4, no. 1, pp. 2–12, 2019. Article (CrossRef Link)

[2] F. Tseng, M. Tsai, C. Tseng, Y. Yang, C. Liu, and L. Chou, "A lightweight Autoscaling Mechanism for Fog Computing in Industrial Applications," *IEEE Transactions on Industrial Informatics, Inventions*, vol. 14, no. 10, pp. 4529–4537, 2018. Article (CrossRef Link)

[3] Y. Kuznietsov, Stückler, and B. Leibe, "Semi-Supervised Deep Learning for Monocular Depth Map Prediction," in *Proc. of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2215-2223, 2017. Article (CrossRef Link)

[4] L. Gu, D. Zeng, W. Li, S. Guo, A. Zomaya, and H. Jin, "Intelligent vnf Orchestration and Flow Scheduling via Model-assisted Deep Reinforcement Leaming," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 279-291, 2020. Article (CrossRef Link)

[5] C. Sears, R. Tandias, and J. Arroyo, "Deep Learning Algorithm," *Survey of Ophthalmology*, vol. 63, no. 3, pp. 0039625717303314, 2017. Article (CrossRef Link)

[6] Z. Zhao, L. Yuan-Xiang, and Y. Fei, "Improved Deep Learning Algorithm Based on Extreme Learning Machine," *Computer Engineering and Design*, vol. 15, no. 4, pp. 1022-1026, 2015. Article (CrossRef Link)

[7] M. Chen, E. Lughofer, and E. Egrioglu, "Deep Learning and Intelligent System Towards Smart Manufacturing," *Enterprise Information Systems*, vol. 16, no. 2, pp. 189-192, 2022. Article (CrossRef Link)

[8] N. Luong, T. Hoang, S. Gong, D. Niyato, W. Ping, Y. Liang, and I. Dong, "Applications of Deep Reinforcement Learning in Communications and Networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133-3174, 2018. Article (CrossRef Link)

[9] B. Li, W. Lu, S. Liu, and Z. Zhu, "Deep-leaming-assisted Network Orchestration for on-demand and Cost-efifective vnf Service Chaining in Inter-dc Elastic Optical Networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 10, pp. D29-D41, 2018. Article (CrossRef Link)

[10] W. Cao, C. Zheng, Z. Yan, and W. Xie, "Geometric Deep learning: progress, Applications and Challenges," *Science China Information Sciences*, vol. 65, no. 2, pp. 126101, 2022. Article (CrossRef Link)

[11] X. Fu, F. Yu, J. Wang, Q. Qi, and J. Liao, "Service Function Chain Embedding for nfv-enabled Iot Based on Deep Reinforcement Leaming," *IEEE Communications Magazine*, vol. 57, no. 11, pp. 102-108, 2019. Article (CrossRef Link)

[12] P. Baskerville, D. Granziol, and P. Keating, "Appearance of Random Matrix Theory in deep learning," *Physica A: Statistical Mechanics and its Applications*, vol. 590, no. 15, pp. 126742, 2022. Article (CrossRef Link)

[13] H. Zhang, "Deep Learning Applications in Simultaneous Localization and Mapping," *Journal of Physics: Conference Series*, vol. 2181, no. 1, pp. 012012, 2022. Article (CrossRef Link)
[14] J. Xie, F. Yu, and I. Huang, "A Survey of Machine Learning Techniques Applied to Software Defined Networicing (sdn): Research Issues and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 393-430, 2018. Article (CrossRef Link)
[15] J. Pereira, and T. Schwede, "Interactomes in the Era of Deep Learning," *Science (New York, N.Y.)*, vol. 374, no. 6573, pp. 1319-1320, 2021. Article (CrossRef Link)
[16] K.Qu, W. Zhuang, Q. Ye, X. shen, J. rao, "Dynamic flow Migration for Embedded Services in sdn/nfv- Enabled 5g Core Networks," *IEEE Transactions on Communications*, vol. 68, no. 4, pp. 2394-2408, 2020. Article (CrossRef Link)
[17] C. Xu, "Enabling Technologies for Future Mobile and Edge Networks," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 3, pp. 4481, 2022. Article (CrossRef Link)
[18] O. Soualah, M. Mechtri, C. Ghribi, and D. Zeghlache, "Online and Batch Algorithms for vnfs Placement and Chaining," *Computer Networks*, vol. 158, no. 3, pp. 41, 2019. Article (CrossRef Link)
[19] Y. Liu, B. Li, J. Wang, D. Li, and Y. Ma, "Multi-information Fusion based Few-shot Web service Classification," *Future Generation Computer Systems*, vol. 130, no. 5, pp. 231-240, 2012. Article (CrossRef Link)
[20] N. Hyodo, T. Sato, and R. Shinkuma, "Virtual Network Function Placement for Service Chaining by Relaxing Visit Order and Non-loop Constraints," *IEEE Access*, vol. 7, no. 7, pp. 165399-165410, 2019. Article (CrossRef Link)
[21] M. Chinthala, "An Editorial Note on Network Services and Applications," *Journal of Computer Science &Systems Biology*, vol. 14, no. 11, pp. 1-1, 2021. Article (CrossRef Link)
[22] M. Muzammal, Q. Qu, and B. Nasrulin, "Renovating Blockchain with Distributed Databases: An Open Source System," *Future Generation Computer Systems*, vol. 90, no. 1, pp. 105-117, 2019. Article (CrossRef Link)
[23] A. Javad, Y. Saeed, and Z. Fatemeh, "Dual-channel Supply chain Coordination Considering Targeted Capacity Allocation under Uncertainty," *Mathematics and Computers in Simulation*, vol. 187, no. 12, pp. 566-585, 2021. Article (CrossRef Link)
[24] W. Liu, S. Wang, and J. Wang, "Evaluation Method of Path Selection for Smart Supply Chain Innovation," *Annals of Operations Research*, vol. 20, no. 8, pp. 1-27, 2021. Article (CrossRef Link)
[25] O. Ramy, I. Adrian, A. Bogdan, and B. Vasile, "Blockchain Technology—Applicability in the Traceability of a Product Throughout the Supply Chain," *Macromolecular Symposia*, vol. 396, no. 1, pp. 202000270, 2021. Article (CrossRef Link)
[26] L. Wei, Z. Han, S. Song, W. He, and X. Li, "Coping with Loss Aversion and Risk Management in the Supply Chain Coordination," *Sustainability*, vol. 13, no. 8, pp. 4364, 2021. Article (CrossRef Link)
[27] C. Shi, and G. Yang, "Robust Consensus Control for a Class of Multi-agent Systems via Distributed PID Algorithm and Weighted Edge Dynamics," *Applied Mathematics and Computation*, vol. 316, no. C, pp. 73-88, 2018. Article (CrossRef Link)
[28] B. Siddiqui, and A. Mahdy, "Storage Virtualization: Towards an Efficient and Scalable Framework," *Computer Science Journals*, vol. 7, no. 1, pp. 12-17, 2015. Article (CrossRef Link)
[29] A. Toosi, J. Son, Q. Chi, and Rajkumar, "Elasticsfc: Auto-scaling Techniques for Elastic Service Function Chaining in Network Functions Virtualization-based clouds," *Journal of Systems and Software*, vol. 152, no. 6, pp. 108-119, 2019. Article (CrossRef Link)
[30] S. Bonfim, L. Dias, and E. Fernandes, "Integrated nfv/sdn architectures: A Systematic Literature Review," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, pp. 1-39, 2019. Article (CrossRef Link)
[31] H. Hantouti, N. Benamar, and T. Taleb, "Traffic Steering for Service Function Chaining," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 487-507, 2019. Article (CrossRef Link)
[32] B. Yi, X. Wang, and K. Li, "A Comprehensive Survey of Network Fimction Virtualization," *Computer Networks*, vol. 133, no. 14, pp. 212-262, 2018. Article (CrossRef Link)

[33] I. Vidal, B. Nogales, D. Lopez, J. Rodríguez, F. Valera, and A. Azcorra, "A Secure Link-Layer Connectivity Platform for Multi-Site NFV Services," *Electronics*, vol. 10, no. 15, pp. 1868, 2021. Article (CrossRef Link)

[34] A. Gonzalez, G. Nencioni, A. Kamisinski, B. Helvik, and P. Heegaard, "Dependability of the nfv orchestrator: State of the Art and Research Challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3307-3329, 2018. Article (CrossRef Link)

Qiujuan Qiao(1983.08.26-), who is a female, is of Han nationality. She was born in Yuanping, Shanxi Province, China. She has a master degree (information science). Now, she is a lecturer who is mainly engaged in modern logistics services and management, logistics information system planning and design.