IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 16, 2020, accepted April 15, 2020, date of publication April 20, 2020, date of current version May 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2988719

Network Function Parallelization for High Reliability and Low Latency Services

JIANHONG ZHOU[1,2], (Member, IEEE), GANG FENG[2], (Senior Member, IEEE), AND YI GAO[2], (Member, IEEE)

[1]School of Computer and Software Engineering, Xihua University, Chengdu 610000, China
[2]National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu 610000, China

Corresponding author: Gang Feng (fenggang@uestc.edu.cn)

This work was supported in part by the Chunhui Project of Ministry of Education, China, under Grant 192618.

ABSTRACT In 5G-and-beyond wireless communication systems, Network Function Virtualization (NFV) has been widely acknowledged as an important network architecture solution to meet diverse service requirements in various scenarios. However, with the increase of network functions, the introduction of NFV may significantly increase the delay of traffic flows, which is much undesired, especially for Ultra Reliable and Low Latency Communication (URLLC) service. Network Function Parallelism (NFP) architecture has been recently proposed as an effective technique to address the bottleneck of NFV technology. NFP can potentially improve the reliability and reduce the delay of service function chains (SFCs). In this paper, we propose a learning based SFC deployment strategy under NFP architecture with aim to improve the service reliability while reducing the end-to-end service delay. Specifically, service reliability is improved by deploying back-up virtual network function (VNF) nodes, while the flow delay is reduced via parallel network function processing. We formulate the VNF deployment as an integer-programming problem with objective of minimizing the reserved computing and bandwidth resources, while guaranteeing the service reliability and end-to-end delay. Considering the hardness and properties of the problem, we transform it as a Markov Decision Process (MDP), and employ a reinforcement-learning algorithm to solve it. We conduct simulations and the numerical results demonstrate that the proposed strategy can significantly improve the service reliability and delay performance, which are crucial for URLLC service.

INDEX TERMS URLLC, NFV, NFP, parallel network service function chain.

I. INTRODUCTION

As one of the three major application scenarios of 5G mobile communication networks, Ultra Reliable and Low Latency Communication (URLLC) service is essential for a wide range of delay-sensitive applications, such as autonomous or assisted driving, augmented reality (AR), virtual reality (VR), tactile Internet, and industrial control. Although network service operators try to support such applications by using existing mobile communication systems, they cannot meet more stringent requirements, such as lower latency, higher reliability and security, of emerging applications. The requirements of some applications in terms of end-to-end delay and reliability are even lower than 1ms and higher than 99.9% respectively [1]. For example, remote surgery requires extremely high sensitivity and accuracy for object manipulation, and the round-trip data transport time is required to be lower than 1ms. Autonomous vehicles need to coordinate with each other to queue and overtake, and thus the reliability of message exchange should be higher than 99.9%. In current deployed LTE networks, the end-to-end delay is approximately 50-100ms, which is about an order of magnitude higher than that of 5G [2]. Therefore, achieving low latency and high reliability requires improvement from the network architecture.

In recent years, Software Defined Network (SDN), Network Function Virtualization (NFV), and Mobile Edge Computing (MEC) have been recognized as three key architectural technologies for 5G. NFV is considered as the main entity of the 5G core network, in which the vendors implement network functions in Virtual Network Function (VNF) components, and VNFs are deployed on high-capacity servers

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

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

or basic cloud architecture instead of dedicated hardware, thereby reducing costs and improving system flexibility. The provisioning of a service requires the execution of a set of VNFs, which in turn form a virtual Service Function Chain (SFC). NFV eliminates the dependence on the hardware platform, and can improve throughput and cost efficiency by flexibly deploying network functions. Meanwhile, using mobile edge computing and deploying the VNFs on the edge network can effectively reduce service delay [3]. However, the latency requirements of URLLC are still very challenging in SDN/NFV based mobile networks. While research on parallel computing in computer programming and high-performance computing has become relatively mature, NF parallelism has just been recently proposed to improve NFV performance [4]. This kind of SFC supporting network function parallelism is called a parallel network service function chain [4]. Therefore, the rational deployment of parallel network function service chains in NFV can enable multiple independent network functions of service requests to work in parallel, thus shortening the effective length of SFC, and significantly reducing the delay.

On the other hand, URLLC service requires not only extremely low latency, but also high reliability. Under the NFV-based network architecture, SFC service requests usually involve multiple VNFs. Therefore, end-to-end service reliability is not only determined by a single node, but also by all VNFs of the serving SFC. Any failure of one VNF of the SFC will cause the service to be interrupted, resulting in wasted resources and service interruptions. Thus, ensuring end-to-end reliability of SFCs is critical for providing URLLC services. Redundancy is an effective method to improve reliability [5], [6]. When a VNF node malfunctions, the carried traffic may be re-routed to the backup node to achieve fault recovery by reserving backup VNF and bandwidth resources. Intuitively, the more redundant backups, the higher the service reliability. However, this mechanism may increase the routing length and thus the end-to-end delay accordingly. Therefore, it is imperative to develop efficient SFC backup scheme by considering delay and reliability simultaneously. Obviously, after introducing the network function parallelization, redundant backup design is different from that in traditional NFV based networks. When VNFs executed in parallel are deployed on different physical hosts, there are multiple parallel links. Thus, the failure of a physical host implementing VNFs may cause some links failure instead of all. This may cause a chain reaction as many packets cannot be merged. Therefore more attention should be paid to the reliability of the parallel service function chains.

At present, the research on reliability for URLLC service mainly focuses on sequential SFCs. Through the deployment of highly reliable nodes and redundant backups of SFCs, the reliability can be effectively improved [5]–[9]. In [7], in order to ensure the end-to-end reliability of SFC, a priority-based deployment scheme was proposed. VNFs with different priorities are deployed on hosts with different reliability. To improve the overall reliability, VNFs with higher priorities are deployed on physical hosts with higher reliability. The authors of [8] proposed a joint optimization framework called reconfigurable awareness and latency-limited service chain for sequential SFC. This framework combines iterative backup selection and routing processes, and allocates resources to the network serving host as much as possible to ensure high reliability and low latency. The authors of [5] proposed an efficient backup method, CERA, by selecting sufficient backup VNFs to meet the reliability requirement of services. In view of the shortage of traditional redundant backups, the authors of [6] proposed an efficient redundant backup algorithm GREP, which can guarantee the reliability of the service within the polynomial time complexity and reduce the backup cost. The authors of [9] proposed to model the procedure of determining the required number of VNF backups as an incremental problem, and proposed a heuristic algorithm to solve it. Unfortunately, all of the aforementioned research work is mainly focused on sequential SFC, which is far from adequate for meeting the delay and reliability requirement of URLLC service. Intuitively, recently emerging Network Function Parallelism (NFP) [4] can potentially improve both service reliability and delay performance. In [10], we propose a new joint Two-Tier NF Parallelization (TNP) framework, which can agilely and flexibly organize parallel NF processing to greatly improve SFC performance in terms of latency and throughput. But the reliability aspect is not taken into account.

In this paper, with aim to minimize reserved resources under the premise of meeting end-to-end reliability and delay constraints under NFP architecture, we propose an intelligent VNF backup node deployment strategy for parallel SFC. Based on the properties of NFP, an MDP model is formulated, and a backup scheme using Q-learning framework is proposed. We evaluate the performance of our proposed backup mechanism using simulation experiments. Numerical results show that the proposed learning based backup algorithm can achieve higher network throughput on the premise of meeting the reliability and delay requirements, compared with ''The parallel SFC based Q-learning (QL-P)'', ''lowest reliability first (LRF)'', ''minimum computing resource first (MCRF)'', ''Random backup'' and ''sequential SFC-based Q-learning (QL-S)'' algorithms. The remainder of the paper is organized as follows. Section II describes the system model. In Section III, we describe the high-reliability SFC deployment model under NFP architecture and formulate the optimal SFC deployment strategy problem. Next, we elaborate the reinforcement learning algorithm for solving the problem in Section IV. Section V presents numerical results as well as discussions. Finally, Section VI concludes this paper.

II. SYSTEM MODEL

In NFP architecture [4], network functions could be executed in parallel in NFV network architecture. Fig.1 shows a parallel SFC model. Each service request can be accomplished by a set of VNFs, and a virtual SFC composed of multiple VNFs is formed in accordance with the execution order. When a

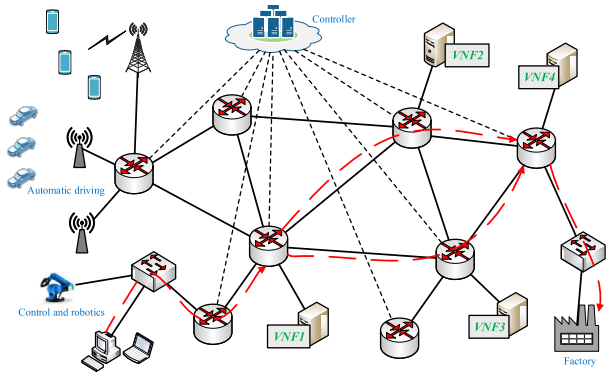

FIGURE 1. Parallel SFC model.

service request reaches the ingress network element of the core network, the corresponding SDN controller deploys corresponding VNFs on physical nodes and arrange the execution order for all VNFs of the SFC according to every VNF's requirements on computing resources, bandwidth resources, storage resources *etc.*. A physical link in the network with bandwidth resources is used for data packet transmission and a physical node with computing and storage resources performs corresponding VNF. In this paper, we assume that every physical node is able to execute every VNF. The physical node will process, forward, copy, and merge the data packets according to the instructions of the controller, and finally transmit the data packets to the destination node.

We assume that each SFC in the network corresponds to a service request. The network elements for all data packets entering and exiting the network are known and fixed. The VNFs required for all data packets execution of one request are the same [4]. NFP compilers are responsible for the parallelized deployment of each SFC, which is called parallel SFC. Specifically, the NFP compiler first identifies the VNFs that can be executed in parallel, and then constructs a parallel service function structure. In the example of Fig. 1, the dashed curve represents a deployed parallel SFC. As VNF_2 and VNF_3 can be executed in parallel, after executing VNF_1, the data packets are copied into two copies and forwarded to traverse different paths, and the merge is completed before VNF_4 is executed.

The physical network of Fig. 1 can be represented as an undirected graph $G(V, E)$, where V and E represent the physical nodes set and link set respectively. For any physical link $e \in E$, the bandwidth and transmission delay are denoted by $B_e > 0$ and $T_e > 0$ respectively. For physical node $v \in V$, the computing resource is denoted by $C_v > 0$. A set of VNFs deployed on any physical, *etc.*). Thus, each node has a reliability level, and all the VNFs running on the node inherits the reliability level of this node. Assume that the failure probability of each node is independent of each other, and only the failure of the physical node is considered here, and the failure of other network components, such as routers, switches, bridges, *etc.*, is not considered. To improve the reliability, some VNFs will be backed up. When a physical node fails, the backup VNFs running on other physical nodes are executed instead of the working VNFs deployed on the failure node. The node where the backup VNF is deployed should be connected to the predecessor and successor nodes of the failure node. We call the connections between backup node and the predecessor or successor node as backup links. Apparently, the backup links and backup node should reserve adequate bandwidth resources and computing resources for the backup VNF.

Let the set of all service requests be S. After NFP compilation of all parallel SFCs, the deployment scheme can be represented as a directed subgraph $G_i\left(V_i, E_i, \psi^i_{req}\right)$ of the physical network $G(V, E)$,where $V_i = \{v_i^1, \ldots, v_i^{|F_i|}\}$ represents the set of physical nodes used to deploy the VNFs of the parallel SFC i and F_i is the VNF set of parallel SFC i. E_i and ψ^i_{req} represent the set of links that link all the nodes and the reliability requirement for the parallel SFC i, respectively. For simplicity, we assume that the bandwidth requirement of the SFC remains unchanged, which is equal to the initial bandwidth b^i_{req}. Let $c^{i,j}_{req}$ represent the computing resources required for the backup VNF $j \in F_i$ of the parallel SFC i.

III. PROBLEM FORMULATION

A. HIGH-RELIABILITY DEDICATED BACKUP MODELING

Assume that the required reliability for any SFC i is ψ^i_{req} ($0 < \psi^i_{req} < 1$), and the required bandwidth is $b^i_{req} > 0$. The VNF set of the SFC i is $F_i \subseteq F$. The ingress and egress network elements are σ_i and δ_i respectively. The reliability of any node deploying VNF j, $\forall j \in F_i$ can be expressed as:

$$r_j = \frac{MTBF_j}{MTBF_j + MTTR_j}, \tag{1}$$

where $MTBF_j$ and $MTTR_j$ represent the mean time between failures and the mean time to repair the physical node deploying the VNF j, respectively. Obviously, $0 < r_j < 1$. It has mentioned in [11] that to avoid load imbalances and improve reliability, any non-parallel VNFs cannot be deployed on the same node. Meanwhile, for a service to work properly, all involved VNFs need to be executed appropriately. In other words, all the nodes deploying all VNFs of the same SFC should work properly.

Thus, for sequential SFCs, we assume that any two VNFs cannot be deployed at the same physical node, and the service reliability can be expressed as

$$\Psi = \prod_{j=1}^{F} r_j. \tag{2}$$

In the deployment of parallel SFC, multiple VNFs executed in parallel may be deployed on the same physical node. If a physical node with multiple VNFs fails, all VNFs deployed on this node will fail accordingly. Thus, the reliability calculation is different from that for the traditional sequential SFC. Specifically, the reliability of the parallel SFC is a product of the reliability of all mapped physical nodes. Some reliability

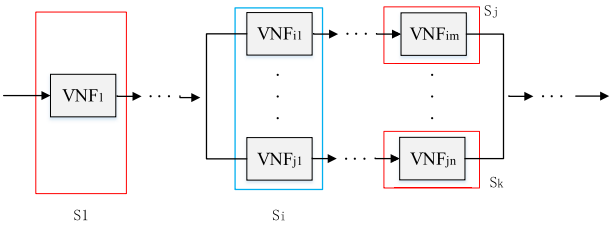

FIGURE 2. Illustration of the SFC parallel execution.

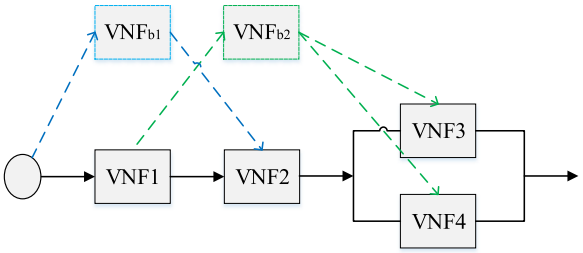

FIGURE 3. Dedicated backup model for parallel SFC.

calculation methods of parallel network functions are given in references [7] and [10]. As shown in Fig.2, a parallel SFC is divided into different subsystems according to different nodes. Each subsystem may deploy multiple VNFs. Let a parallel SFC consists of $K(K \geq 1)$subsystems $\tilde{S} = \{s_1, \ldots, s_k\}$, and each subsystem s_n consists of $k_n, (k_n \geq 1, \sum k_n = |F_n|)$VNFs. In general, the reliability of the parallel SFC can be expressed as:

$$\psi = \prod_{n=1}^{K} \mathrm{Re}(s_n), \tag{3}$$

where $\mathrm{Re}(s_n), 0 < \mathrm{Re}(s_n) < 1$ indicates the reliability of the physical node where the subsystem n is located. Obviously, for the same request, K and F represent the number of multiplication factors when calculating the reliability for parallel and sequential VNF deployment respectively, and $K < F$. It can be seen that when multiple VNFs deployed on the same physical node n can be executed in parallel, the reliability of the entire SFC can be improved to a certain extent.

There are generally two types of backup models, namely dedicated backup and shared backup [12]. In order to support highly reliable services, we adopt the dedicated backup model. Fig. 3 shows a dedicated backup model with parallel network functions. The backup VNFs VNF_{b1} and VNF_{b2} provide dedicated backup for working VNFs VNF_1 and VNF_2, respectively. Meanwhile, the required backup links are added between the backup nodes and their neighboring working nodes. Unlike traditional sequential SFC backup, a VNF mapping node may have multiple inbound and outbound flows. Therefore, when considering the selection of node for backup VNFs, it is necessary to choose the node with least backup links to reduce the reservation of backup bandwidth resources, and then lower the resource occupation ratio to the greatest extent. We define binary 0-1 variables $x_{i,j}^{v}(i \in S, j \in F_i, v \in V)$ to represent the backup deployment of VNF j in the SFC i, *i.e.*,

$$x_{i,j}^{v} = \begin{cases} 1, & \textit{VNF } j \textit{ of SFC } i \textit{ is deployed on node } v \\ 0, & \textit{otherwise} \end{cases}. \tag{4}$$

We define the binary 0-1 variable $y_{i,j}^{e}(i \in S, j \in F_i, e \in E)$ to represent the link occupied by VNF j in the SFC i, which means that the link e is occupied to execute the packet transmission which is served by the VNF j of SFC i.

$$y_{i,j}^{e} = \begin{cases} 1, & \textit{link } e \textit{ occupied by VNF } j \textit{ of SFC } i \\ 0, & \textit{otherwise} \end{cases}. \tag{5}$$

The binary 0-1 variable $h_{i,j}(i \in S, j \in F_i)$ is introduced to indicate whether the VNF j of the SFC i is backed up, and we define

$$h_{i,j} = \begin{cases} 1, & \textit{VNF } j \textit{ of SFC } i \textit{ is backed up} \\ 0, & \textit{otherwise} \end{cases}. \tag{6}$$

B. HIGH-RELIABILITY DEDICATED BACKUP MODELING

In our system, the highly reliable backup of services is considered in a resource limited environment. Thus, when selecting backup nodes, it is necessary to jointly consider node computing resource, reliability, link resource and link delay limits.

1) NODE CONSTRAINTS

As a failure of the physical node will cause all VNFs deployed on the node to fail, all the nodes deploying working VNFs are inappropriate to serve as backup nodes. Meanwhile, to save resources, we only consider the case of a maximum of one backup copy. The node constraints are given as follows:

$$x_{i,j}^{k} = 0, \quad \forall i \in S, \quad \forall j \in F_i, \ \forall k \in V_i, \tag{7}$$

$$\sum_{k \in V} x_{i,j}^{k} \leqslant 1, \quad \forall i \in S, \ \forall j \in F_i. \tag{8}$$

When a VNF is deployed on a physical node, the physical node needs to reserve computing resources for executing this VNF. Assume the actual computing resource of node k is C_k, another node constraint is

$$\sum_{i \in S} \sum_{j \in F_i} x_{i,j}^{k} \times c_{req}^{i,j} \leqslant C_k, \ \forall k \in V. \tag{9}$$

To improve the reliability of the SFC with parallel network functions, multiple backup VNFs may be backed up on the same node, and two VNFs that cannot be executed in parallel cannot be backed up on the same node. We define the 0-1 variable $z_{f,f'}^{i}$. If and only if $z_{f,f'}^{i} = 1$, the VNF f and f' of SFC i can be executed in parallel, and not vice versa. Thus, there're one more node constrain as

$$x_{i,f}^{k} + x_{i,f'}^{k} \leqslant 1 + z_{f,f'}^{i}, \quad \forall i \in S, \ \forall f, f' \in F_i, \ \forall k \in V. \tag{10}$$

Assume there are n VNFs of an SFC are deployed on the same physical node, of which $m(m < n)$ VNFs are

backed up VNFs. When the working node fails, the service cannot be provisioned normally. In this case, the backup only cause waste of resources rather than improving end-to-end reliability. Thus, all VNFs deployed on the same working node can only be backed up VNFs or working VNFs. For two different network functions deployed on the same node, the following constraint must be met:

$$x^u_{i,f} \oplus x^v_{i,f'} = 0, \quad \forall i \in S,\ \forall f, f' \in F_i,\ \forall u, v \in V. \quad (11)$$

2) TRAFFIC CONSTRAINTS

For $\forall e \in E$, the sum of the reserved bandwidth of all backup links mapping to e cannot exceed the maximum bandwidth provided. We thus have the following constraint:

$$\sum_{i \in S} \sum_{j \in F_i} y^e_{i,j} \cdot b^i_{req} \leqslant B_e, \quad \forall e \in E. \quad (12)$$

Due to the involvement of backup, the traffic of working VNF nodes will no longer be conserved, and the traffic conservation constraints cannot be used. We call the physical node where the previous VNF of current working VNF resides as the predecessor node, and the physical node where the next VNF of current working VNF resides as the successor node. A backup VNF node in G_i may have multiple predecessors and successors. Assume that the set of predecessor nodes of the backup VNF $j \in F_i$ of the SFC i is $V^F_{i,j}(V^F_{i,j} \in V_i)$ and the set of successor nodes is $V^L_{i,j}(V^L_{i,j} \in V_i)$. If the predecessor (successor) of the backed up VNF j is the ingress network element (egress network element), then $V^F_{i,j} = \sigma_i$, $(V^L_{i,j} = \delta_i)$. In this case, these nodes should meet the following constraints instead of the traffic conservation constraints:

$$\sum_{l.head = v_m} y^l_{i,j} = h_{i,j}, \quad \forall i \in S, \forall j \in F_i, \forall v_m \in V^F_{i,j}$$
$$\sum_{l'.tail = v_n} y^{l'}_{i,j} = h_{i,j}, \quad \forall i \in S, \forall j \in F_i, \forall v_n \in V^L_{i,j}. \quad (13)$$

The node where the backup VNF is located on no longer meets the traffic conservation, and the inflow and outflow are as the same as the working VNF node. The inflow and outflow of the backup node is $|V^L_{i,j}|$ and $|V^F_{i,j}|$, respectively. Thus following constraints should be satisfied:

$$x^v_{i,j} \times \sum_{l.head = v} y^l_{i,j} = x^v_{i,j} \times |V^L_{i,j}|, \quad \forall i \in S, j \in F_i,\ \forall v \in V$$
$$x^v_{i,j} \times \sum_{l.tail = v} y^l_{i,j} = x^v_{i,j} \times |V^F_{i,j}|, \quad \forall i \in S, j \in F_i,\ \forall v \in V. \quad (14)$$

If the working link is a serial link, that is $|V^L_{i,j}| = |V^F_{i,j}| = 1$, then $\sum_{l.head=v} y^l_{i,j} = \sum_{l.tail=v} y^l_{i,j}$, which satisfies the traffic conservation. For the other nodes meeting the traffic conservation should satisfy the following constraint:

$$\sum_{l.head = v} y^l_{i,j} = \sum_{l.tail = v} y^l_{i,j},$$
$$(\forall i \in S, j \in F_i, \quad \forall v \in V - V^F_{i,j} - V^L_{i,j}, x^v_{i,j} = 0). \quad (15)$$

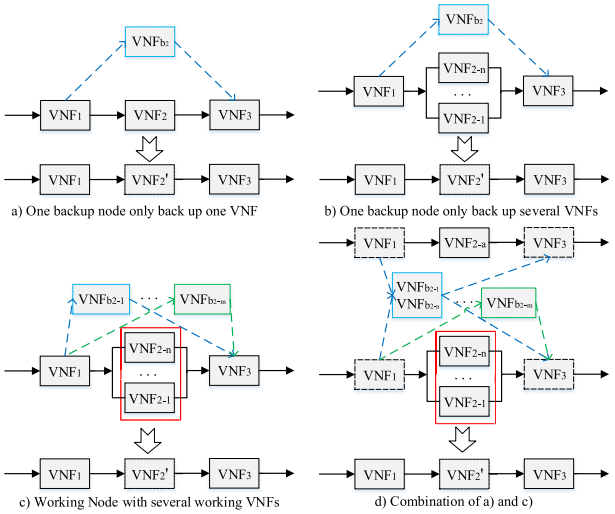

FIGURE 4. Backup SFC update model.

3) RELIABILITY CONSTRAINTS

In sequential SFC backup, it is only necessary to consider the constraint that the backup of different VNFs cannot be placed on the same node, and thus the backup design for reliability is easy. In parallel SFC backup, multiple VNFs can be deployed on the same physical node to execute in parallel. Meanwhile, multiple VNFs can also be divided into multiple branches to execute concurrently and then the results are merged. In this case, the backup design for reliability becomes much more complicated. To address this difficulty, we continuously update the parallel SFC structure by combining working VNFs and backup VNFs into new VNFs. As shown in Fig 4, several possible situations for backing up VNFs in the NFP architecture are listed.

As mentioned above, at most one copy is backed up for VNF j of SFC i. If only one backup VNF is deployed on the physical node, the model can be illustrated in Fig 4 (a). The reliability after backup is:

$$\psi_2 = 1 - (1 - r_2) \cdot (1 - r_{b2}), \quad (16)$$

where r_2, r_{b2} represent the reliability of the physical node where VNF_2 and VNF_{b2} are deployed, and ψ_2 is the reliability of the combined new VNF. As shown in Fig 4(b), if several backup VNFs deployed on one physical node are executed in parallel, while the corresponding working VNFs are deployed on several physical nodes, the reliability after backup is:

$$\psi_2 = 1 - (1 - r_{b2}) \cdot (1 - \prod_{i=1}^{n} r_{2-i}). \quad (17)$$

It is obvious that the reliability of the backed up VNFs is higher than that of the same VNFs without backup. The higher the reliability of the node with backup VNF is, the greater the reliability of the backed up VNF is. Meanwhile, backup on the same physical node can reduce operation and maintenance costs.

Next, we consider that n working VNFs are deployed on one node as shown in Fig 4(c). In this case, all the backup VNFs may be backed up on the same node or on different physical nodes. Let n working VNFs be backed up on $m(m \leqslant n)$ physical nodes and the reliability of the updated VNF_2 is:

$$\psi_2 = 1 - (1 - r_2) \cdot (1 - \prod_{i=1}^{m} r_{b2-i}), \tag{18}$$

where r_2 and r_{b2-i} represent the reliability of the physical node on which the n VNFs are deployed and the reliability of the i_{th} backup node, respectively. Obviously, the reliability of the updated VNF_2 after backup is enhanced.

At last, a more complicated situation is considered as shown in Fig 4(d). The working VNFs, VNF_{2-a} and $VNF_{2-1} - VNF_{2-n}$, are executed in parallel on different physical nodes, while the physical node with backup VNF VNF_{2-a} also backs up other VNFs. We assume that the backup node of VNF_{2-a} is as the same as the backup node of VNF_{2-1}. Let the event p' indicate that the physical node, where VNF_{2-1}-VNF_{2-n} are deployed, is working normally, the event p'' indicate that the physical node, where VNF_{2-a} are deployed, is working normally, and the event p_i indicate that the physical node, where VNF_{b2-i} are deployed, is working normally. Then the reliability of the updated VNF VNF_2 is

$$\begin{aligned}
\psi_2 &= \mathrm{P}(p'p'') + \mathrm{P}(\bar{p}'p'') \cdot \mathrm{P}(p_1 \ldots p_m) + \mathrm{P}(p'\bar{p}'') \cdot \mathrm{P}(p_1) \\
&\quad + \mathrm{P}(\bar{p}'\bar{p}'') \cdot \mathrm{P}(p_1 \ldots p_m) \\
&= \mathrm{P}(p'p'') + \mathrm{P}(\bar{p}') \cdot \mathrm{P}(p_1 \ldots p_m) + \mathrm{P}(p'\bar{p}'') \cdot \mathrm{P}(p_1) \\
&= r_a \cdot r_b + (1 - r_a) \cdot \prod_{i=1}^{m} r_i + r_a \cdot (1 - r_b) \cdot (1 - r_1) \\
&\geqslant r_a \cdot r_b,
\end{aligned} \tag{19}$$

where r_a and r_b represent the reliability of the physical node on which the VNF_{2-a} is deployed and on which the VNF_{2-1}-VNF_{2-n} are deployed, respectively. In this case, the overall reliability of the updated VNF decreases with the number of backup hosts.

For each SFC i, there is a reliability requirement ψ_{req}^{i}, and the links deployed on the physical network have a reliability ψ_i. The new reliability after backup can be denoted by ψ_i'. Then the reliability needs to meet the constraints:

$$\psi_i{}' \geqslant \psi_{req}^{i}, \ \psi_i{}' \geqslant \psi_i, \quad \forall i \in S. \tag{20}$$

4) DELAY CONSTRAINT

Dedicated redundant backup will not change the deployed structure of the parallel SFC. When a node fails, the working VNF will be transferred to the backup VNF node for execution. Assume that the execution time of the working VNF and the corresponding backup VNF are the same, the change in the delay for executing the backup VNF instead of the working VNF is only the change in the link delay. If the backup VNF node is far from the working VNF node, it will cause a sharp increase in link delay. Therefore, when considering the selection of backup nodes, the change of link delay also needs to be considered.

Different service function structures have different end-to-end delay calculation methods. For description simplicity, we define network function delay Δ as the delay required to execute a VNF, including the link delay of routing to the VNF node and the processing delay performed by the VNF, Then we have

$$\Delta = \max(\vec{t}_L) + \bar{t}_P + \max(\overleftarrow{t}_L{}'), \tag{21}$$

where $\vec{t}_L$ is the maximum link delay from the previous VNF mapping node to the current VNF mapping node; $\bar{t}_P$ is the processing time of current VNF; $\overleftarrow{t}_L{}'$ is the maximum link delay from the current VNF mapping node to next VNF mapping node. Thus, the actual requested network function latency is:

$$t_i = \sum_{j \in F_i} [\Delta_{i,j} \cdot h_{i,j} + \Delta_{i,j}{}'(1 - h_{i,j})], \quad \forall i \in S, \tag{22}$$

where $\Delta_{i,j}$ and $\Delta_{i,j}{}'$ are the network function delay for backup VNF j and working VNF j of SFC i, respectively. Meanwhile, the average requested network function latency $\bar{t}_i = \frac{t_i}{|F_i|}$, $\forall i \in S$, which will be used in the next Sec.

To quantify the delay incurred by the switching process from working VNF node to the backup VNF node, we define the constraint factor τ as the tolerable delay increment factor. Thus, to avoid the delay increased too much, the delay constrain t_i should satisfy

$$t_i \leqslant \tau \times \sum_{j} \Delta_{i,j}{}', \quad \forall i \in S, \tag{23}$$

where $\sum_{j} \Delta_{i,j}{}'$ is the service delay of SFC i before backing up.

C. PROBLEM FORMULATION

In the deployment of parallel SFC, there are a large number of VNFs that can be executed in parallel, and the reliability is supposed to be higher than that of traditional sequential SFC deployment. However, there are still some low-reliability physical nodes, which affect the reliability of the entire link seriously. Fortunately, we could improve the reliability of these nodes through backup. Considering parallel SFC backup in a known network, our goal is to obtain an optimized VNF backup scheme with the smallest resource occupation under a specific reliability constraint. To back up a VNF on a physical node, it is necessary to reserve computing resources on the node as well as reserve bandwidth resources for certain links. Different VNFs require different resources. Thus, the resources reserved for different backup VNFs are also different. Thus, the backup problem mentioned above can be formulated as a 0-1 integer-programming problem. As minimizing bandwidth and computing resources are considered simultaneously, we can formulate our problem as a

multi-objective constrained optimization problem as

$$\min(\sum_{i\in S}\sum_{j\in F_i}\sum_{k\in V} x_{i,j}^{k} \times c_{req}^{i,j},\ \sum_{i\in S}\sum_{j\in F_i}\sum_{m\in E} y_{i,j}^{e} \times b_{req}^{i}). \quad (24)$$

Note that we did not present the constraints in (23) for brevity. A common method for solving multi-objective optimization problems is to transform the multi-objective optimization problem into a single-objective optimization problem through mathematical transformations, *e.g.* the evaluation function method. Then, an approximate optimal solution of the original problem can be obtained by solving the single-objective optimization problem. In this paper, we choose the weighted summation of the two objective functions after normalization as the evaluation function. The weighting factors are α and β ($\alpha + \beta = 1$). Then the objective function of the original problem can be transformed into:

$$\begin{aligned}\min\ (&\frac{\alpha}{\sum_{k\in V} C_k} \times \sum_{i\in S}\sum_{j\in F_i}\sum_{k\in V} (x_{i,j}^{k} \times c_{req}^{i,j}) \\ &+ \frac{\beta}{\sum_{e\in E} B_e} \times \sum_{i\in S}\sum_{j\in F_i}\sum_{e\in E} (y_{i,j}^{e} \times b_{req}^{i}))\end{aligned}$$
$$\text{s.t. Constraints(6)} \sim (14), (19), (22). \quad (25)$$

where (6) $\sim$ (10) are the node constraints, (11) $\sim$ (14) are the traffic constraints, (19) is the reliability constraint, and (22) is the delay constraint. Obviously, this is a typical 0-1 integer-programming (0-1 IP) problem with multi-constraints, and some traditional continuous region solutions are infeasible for this problem as our 0-1 IP problem is discrete. Currently, there are three types of algorithms for solving such problems: precise algorithms (*i.e.* dynamic programming, recursive method, retrospective method, branch-and-bound method, *etc.*), approximation algorithms (*i.e.* greedy algorithm, Lagrange algorithm, *etc.*), and intelligent optimization algorithms [13]. Normally, precise algorithms could achieve the global optimal with surprisingly high computational complexity, while intelligent optimization algorithm could only achieve the suboptimal solution with high accuracy and small time complexity. The performance of approximation algorithm is between both. Meanwhile, if the static environment is considered, the common optimization solver can be used to solve the problem. However, in practical network environments, there are too many random attributes such as link congestion. As the network environment is constantly changing, we need to predictively obtain the backup strategy quickly. Thus, the traditional method is no longer applicable. Fortunately, Reinforcement learning (RL) algorithms can be exploited to make sequential decisions towards the long-term objective by continuously interacting with the environment.

IV. Q-LEARNING BASED VNF BACKUP ALGORITHM

We consider to back up some VNFs of an existing request in a network with limited resources. As the request arrives in sequential order, it is unnecessary to consider all requests simultaneously. Thus, we will back up each request one by one in this paper. In this case, the problem is transformed into a cost minimization problem about how to back up a parallel SFC with least bandwidth and computing resource occupation under some constraints in a dynamic network environment. RL is especially appropriate for solving decision problems in dynamic environments. This inspires us to use Q-learning algorithm to solve the problem of (24), and obtain a backup scheme for parallel SFC that meets reliability requirement.

A. Q-LEARNING BASED HIGH RELIABILITY BACKUP ALGORITHM

The backup procedure can be modeled as a Markov Decision Process (MDP), in which the transition probability is unknown. Thus, we intend to adopt Q-learning algorithm to solve our problem under the MDP framework. Q-Learning (QL) is a representative algorithm for reinforcement learning. It does not need to know the environment model and can be used for continuous tasks. The backup decision maker is modeled as an agent to choose the best backup candidate nodes for each service, thereby improving the reliability. When a node is selected or cancelled, the reliability of the service changes and the resource status in the network also changes. Therefore, the MDP can be represented as $M = \{S, A, P, R\}$, where S represents state, A represents action, P denotes the transition probability, which is unknown in our system, and R represents the reward. The details of the model are elaborated as follows.

Agent: The controller for making SFC deployment decisions, such as the Management and Orchestration (MANO) entity of NFV [14].

State S: We use the backup node selection status of each request, reliability of each VNF, and the network status to represent the system state. Let S be the entire possible state space, and $s(t) \in S$ is the state at time t. Let there be V physical nodes and E links in the network. If the deployed parallel SFC includes F VNFs, we have $s(t) = [(j,k), \theta_1, \theta_2, \ldots, \theta_F, c_1, c_2, \ldots, c_N, b_1, b_2, \ldots, b_E]$, where (j,k), $1 \leqslant j \leqslant F, 1 \leqslant k \leqslant N$ means the backup status at time. If $(j,k) = 1$, it means the that the VNF j is backed up in the node k, otherwise, $(j,k) = 0$. θ_i is the reliability of the working VNF. $c_i (c_i \geqslant 0)$ means the available idle computing resource and $b_i (b_i \geqslant 0)$ means the available bandwidth resource of link i.

Action A: We define the selection of VNF backup nodes as the action. When a VNF needs to be backed up, a certain node in the network is selected as the backup node according to certain constraints. Thus, we define the action as $a(t) \in A$, $a(t) = (a_1, \ldots, a_i, \ldots a_N)$, $a_i \in \{0, 1\}$, where $a_i = 1$ means that the physical node i is selected as the backup node at time t, otherwise, $a_i = 0$ means that the physical node i is abandoned as the backup node at time t.

Transition Probability P: Let $P = \{p_{s,s'}^{a} | s, s' \in S, a \in A\}$ represent the set of transition probabilities of the states. $p_{s,s'}^{a} = P[s(t+1) = s | s(t) = s', a(t) = a]$ means that

the transition probability of performing action a from current state s to next state s'.

Reward R: When action $a(t)$ is performed, the corresponding reward will be obtained. Many factors should be taken into consideration when defining the reward function. First, the state $s'(t)$ after taken action $a(t)$ should satisfy all the constraints of (6) $\sim$ (14), (19), (22). Thus we define $R(s, a)$ as (25) if all constraints are satisfied and otherwise,$R(s, a) = -\phi$, where ϕ is a penalty factor, which is a sufficiently large positive real number. As our performance is jointly determined by reliability, delay, and resource occupation rate, we define the reward function as

$$R(s, a) = R_{re}(s, a) \times R_d(s, a) \times R_{ce}(s, a), \tag{26}$$

where $R_{re}(s, a)$ is the reliability reward function at state $s(t) = [(j, k), \theta_1, \theta_2, \ldots, \theta_F, c_1, c_2, \ldots, c_N, b_1, b_2, \ldots, b_E]$. $R_{re}(s, a)$ is defined as

$$R_{re}(s, a) = \begin{cases} 1, & if\ \psi \geqslant \psi_{req} \\ \psi, & otherwise \end{cases}, \tag{27}$$

where ψ is the service reliability after backup. $R_d(s, a)$ is the delay reward and is defined as [15]:

$$R_d(s, a) = \begin{cases} 0, & \bar{t} \geqslant \tau \times \eta \\ \dfrac{\tau}{\tau - 1} - \dfrac{\bar{t}}{\eta(\tau - 1)}, & \tau \times \eta > \bar{t} \geqslant \eta \\ 1, & \bar{t} < \eta \end{cases}, \tag{28}$$

where η is the average SFC mapping delay, and τ is the delay tolerant factor. If the delay of the backup link is increased too much to satisfy the URLLC requirement, it will be an invalid backup.

$R_{ce}(s, a)$ is the resource occupation reward function and is defined as

$$R_{ce}(s, a) = \chi \times \frac{\sum c_i}{C_{total}} + \omega \times \frac{\sum b_i}{B_{total}}, \tag{29}$$

where C_{total} is the sum of all computing resources of the physical nodes, and B_{total} is the sum of bandwidth resources of the physical link. The coefficients χ and ω represent weight values, satisfying $\chi + \omega = 1$. At state s, considering the backup relationship (j, k), the VNF is unnecessary to backup if the backup node is the same as the original working node. Meanwhile, to occupy as few resources as possible, the value of the reward function should be greater if the occupied resource is less.

B. THE Q-LEARNING BASED HIGH RELIABILITY BACKUP ALGORITHM

The Q-learning algorithm is a value-iterative algorithm, which has nothing to do with the environment model, and thus it does not depend on the state transition matrix. The state-action matrix Q is the key to the Q-learning algorithm. At any time t, the corresponding action a is chosen based on the state-action matrix Q, and it should satisfy

$$Q(s, a) = \max_a \{Q(s, a)\}, \tag{30}$$

where $Q(s, a)$ is the Q value in the state-action matrix under action a and state s.

When a specific action is performed, the system enters the next state, obtains a feedback value at the same time, and updates Q value iteratively according to the following training function

$$Q(s, a) = Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)], \tag{31}$$

where s' and a' represent the next state and next action respectively. $\alpha > 0$ is the learning rate, and γ is the discount factor. It can be seen that the larger the learning rate α, the less the effect of retaining from the previous training. The larger the discount factor γ, the greater the role of subsequent decisions. After that, the Agent performs a loop operation on the next state until the optimal value $Q^*(s, a)$ in the Q-value matrix $Q(s, a)$ satisfies the Bellman equation:

$$Q^*(s, a) = E[r + \max Q^*(s', a')]. \tag{32}$$

Compared with traditional statistic backup solutions, our proposed Q-Learning based high reliability backup algorithm is more flexible as it updates states through iterative interactions with the network. When for solving problems such as network congestion, busy computing resources, and node outages occur in the network, the Q-learning algorithm can update the backup scheme in real time according to the current network status and make dynamic adjustments. The Q-Learning based high reliability backup algorithm is illustrated as algorithm 1.

Algorithm 1 : Q-Learning Based High Reliability Backup Algorithm

1: **Step 1: Initialization**
2: Initialize $Q = 0$, state value $s \in S$, action $a \in A$, learning rate α, and discount factor γ.
3: Obtain current reward $r(s, a)$.
4: **Step 2: Chose and execute action**
5: Chose an action based on $\varepsilon - greedy$ algorithm.
6: If the chosen action $a' \in A$, obtain current reward $r(s, a)$ and observe next state s'.
7: **Step 3: Update Q value**
8: Update Q value based on $Q(s, a) \leftarrow Q(s, a) + \alpha[r(s, a) + \gamma \max_{a'} Q(s', a') - Q(s, a)]$.
9: **Step 4: Update state**
10: Update state as $s = s'$, repeat Step 2.

V. SIMULATION AND RESULT ANALYSIS

Based on the analyzing result of [5]–[7], [9] that the service reliability can be improved by backing up related virtual network functions (VNFs), in this paper we develop a Q-learning based algorithm (QL-P) for parallel SFC to obtain the optimized VNF backup strategy efficiently. And then, we verify the advantage of our proposed QL-P algorithm by comparing the performance of different VNF backup strategies obtained by different algorithms. To the best of our knowledge, there is no existing known algorithm for the parallel SFC backup

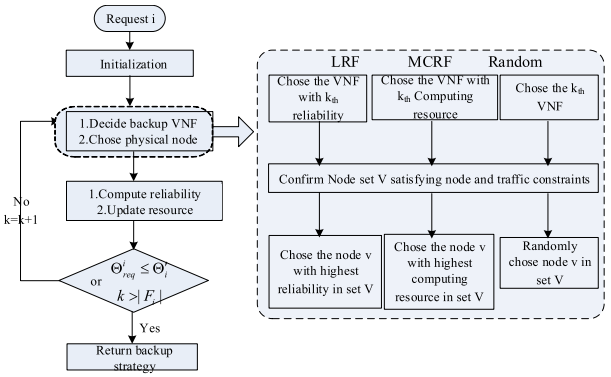

FIGURE 5. Flowchart of LRF, MCRF and Random algorithms.

problem can be used as comparison reference. Thus, we use three heuristic based backup strategies for performance evaluation of our proposed algorithm, namely Lowest Reliability First (LRF), Minimum Computing Resource First (MCRF), and Random algorithms. To highlight the advantage of the parallel SFC, the Q-learning based algorithm is realized for both parallel SFC and sequential SFC back up. Thus, five algorithms are compared to verify the performance of our proposed algorithm in this section, which are parallel SFC based Q-learning (QL-P) algorithm, reliability-first algorithm, computing-resource-first algorithm, random backup, and sequential SFC-based Q-learning (QL-S).

As shown in Fig. 5, the LRF algorithm starts from examining the reliability of the SFC. When the reliability of the SFC does not meet the conditions, the most reliable node of the physical node is backed up to the less reliable VNF in the SFC under the constraint conditions. The MCRF algorithm considers the limited computing resources. When the request does not meet the reliability requirements, the node with the most abundant computing resources is selected as the backup node. The Random algorithm only considers reliability. When the reliability of the request does not meet the conditions, the backup node is randomly selected under the constraint of the backup node. To evaluate the performance of the proposed QL-P algorithm, we assume that one VNF is required to be backed up to meet the service reliability requirement. However, the QL-P algorithm is also feasible for the situation that more than one VNFs backup is required.

A. SIMULATION ENVIRONMENT AND PARAMETERS SETTING

The network topology considered in this paper has 14 nodes, 21 links, and the computing resources are counted with the number of processor cores of the physical nodes. The computing resources of each node are any number between 4 to 64. The bandwidth of each link is any between 80$Mbps$ to 100$Mbps$ [16], [17], and the reliability of each node is between 0.9999 to 0.99999 [17]. Commonly, the number of VNFs for each service request is 4, and the deployment of backup VNF requires The computing resources are uniformly

TABLE 1. Simulation parameters.

Parameters	Value Range
Computing resource (/node)	4-64
Bandwidth resource ($Mbps$)	80-100
Reliability (/node)	99.99%-99.999%
Computing resource for one VNF	1-5
Number of Ingress network elements	1-14
Number of Outgress network elements	1-14
Number of VNFs/request	4
Latency tolerance factor τ	2
Learning rate α	0.5
Discount factor γ	0.8

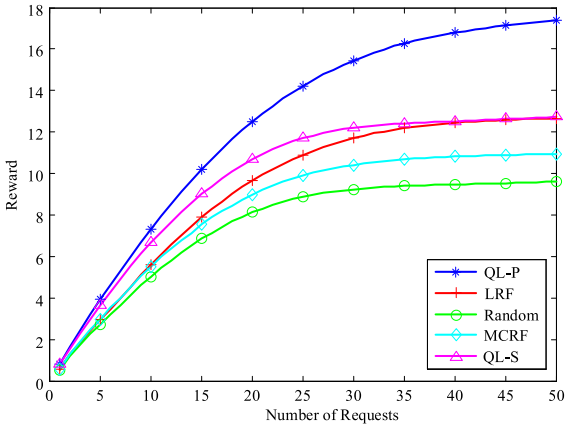

FIGURE 6. Reward comparison with increasing backed up request.

distributed in [1, 5]. The latency tolerance factor τ due to backups is 2. The learning rate $\alpha = 0.5$, and the discount factor $\gamma = 0.8$. All parameters are illustrated as table 1. The following performance metrics are used to evaluate the performance of our proposed backup scheme.

1) *Delay*: The average network function execution delay T_{ed}.
2) *Reliability*: The reliability of the entire service after backup $\psi_i{}'$.
3) *Computing occupation ratio*: All computing resources requested for backup reservation $Ratio_c$, $Ratio_c = \sum_{i\in S}\sum_{j\in F_i}\sum_{k\in V} x_{i,j}^k \times c_{i,j}$.
4) *bandwidth occupation ratio*: All bandwidth resources requested for backup reservation $Ratio_b$, $Ratio_b = \sum_{i\in S}\sum_{j\in F_i}\sum_{m\in E} y_{i,j}^m \times b_i$.

B. NUMERICAL RESULTS AND DISCUSSIONS

1) REWARD

Fig. 6 shows the cumulative reward function as a function of the number of requests. We can observe that as the number of requests continues to increase, the value of the reward function continues increasing and eventually converges to a fixed value. In the same situation, the Q function is the highest reward function, while the stochastic algorithm has the lowest reward function. The definition of the reward function in the Q-learning algorithm is related to the three aspects of

reliability, resource occupancy, and average execution network function delay. The LRF algorithm backup method considers physical nodes with high reliability nodes to ensure high reliability. The value of the reward function of LRF is higher than that of Random and MCRF. The MCRF algorithm considers selecting a physical node with more resources as a backup node to ensure less resource occupation. When the Random algorithm is used, the backup node is randomly selected without considering reliability, resource occupation, and delay, so the reward value obtained is relatively low. Meanwhile, when the sequential SFC adopts the Q-learning algorithm under the same circumstances, the return obtained is lower than the return value of the parallel SFC. It is because the introduction of network function parallelism makes fewer physical nodes mapped.

2) RELIABILITY

The reliability comparison among all the algorithms is illustrated as Fig. 7. We can see that the reliability decreases with the number of backed up requests. The is due to the limited physical resources in the underlying network, as the number of backed up service requests increases, the system cannot provide sufficient resources to meet the backup requirement for all service requests. When system resources are sufficient, all algorithms can meet the reliability requirements of the service. According to the definition of the reward function, the reward value equals 1 if the reliability constraint is satisfied. In this case, as the Q-learning based algorithm comprehensively considers the trade-offs between reliability, resource occupation and delay, it is less reliable than other algorithms. However, in the resource limited scenario, Q-learning considers the balance between resource occupation and reliability, so it can satisfy more requests. Meanwhile, if the reliability does not meet the reliability requirements, the Q-learning based algorithm can obtain the maximum reliability compared to the other algorithms. The QL-S algorithm is dedicated for sequential SFC, so different VNFs have to be deployed on different physical nodes. However, as parallel network functions in the parallel SFC can be deployed and executed on the same physical node, the number of involved physical nodes for an SFC in the parallel execution way is less than that for the same SFC in the sequential execution way. Based on the calculation of reliability, it can be known that the more physical nodes mapped by the VNF, the lower the reliability, so that the more resources need to be backed up to achieve the same reliability. Obviously, the Q-learning algorithm is appropriate to meet the reliability requirements of URLLC scenarios in 5G, and it can obtain higher reliability when resources are insufficient.

3) RESOURCE OCCUPANCY

Fig. 8 shows a comparison of bandwidth and computing resource occupancy. As the number of SFC requests increases, the resources occupied by VNF backups gradually increase accordingly. When the underlying physical resources cannot provide backups service, the resource occupation tends to stabilizing. The Q-learning based algorithm considers resource occupation as a part of the reward function, and thus the Q-learning method uses the least computing and bandwidth resources. Fig. 8 (a) shows the comparison of bandwidth resource occupancy rate for different algorithms. Using the Q-learning based algorithm can reduce the bandwidth occupancy rate for both parallel SFC and sequential SFC backup, and the difference of bandwidth

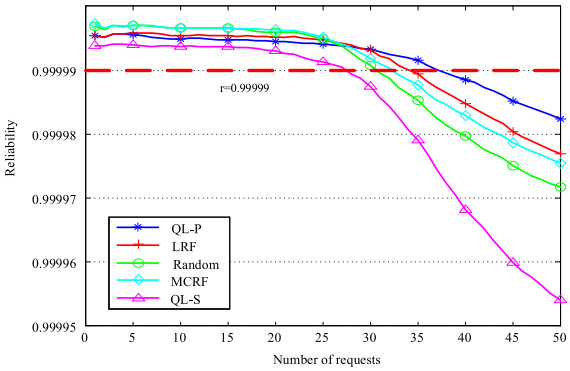

FIGURE 7. Reliability comparison with increasing backed up request.

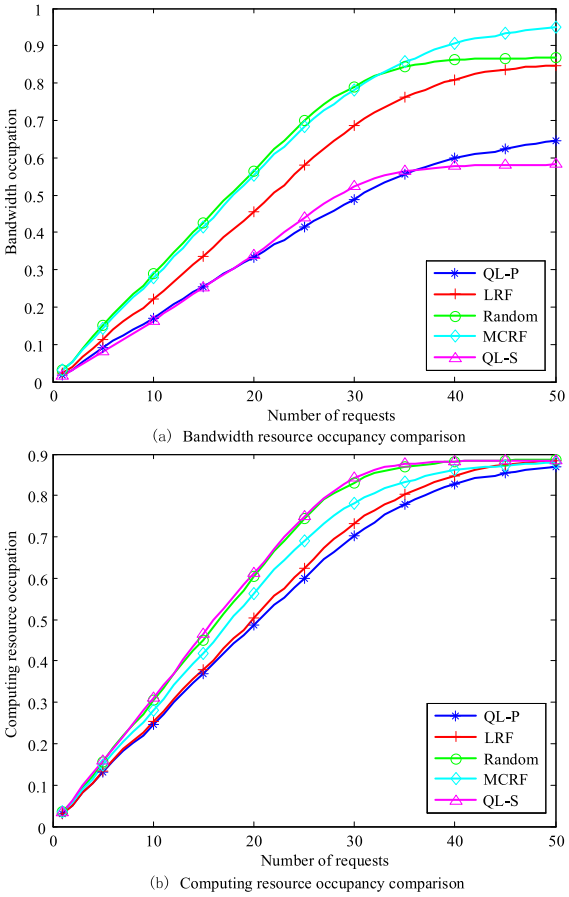

FIGURE 8. Resource occupancy comparison.

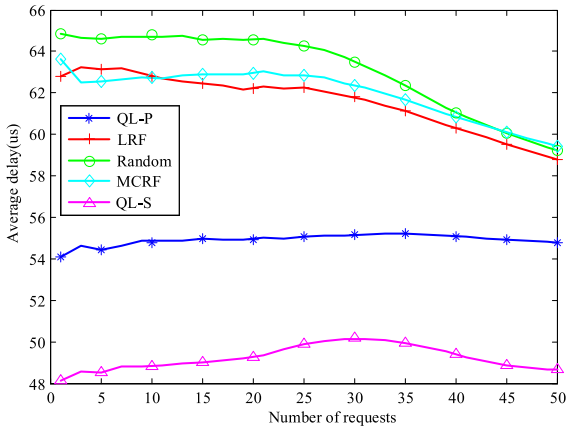

FIGURE 9. The average network function execution delay comparison.

resources occupation is very little for the sequential SFCs and parallel SFCs. The MCRF algorithm, LRF algorithm, and Random algorithm select backup nodes with the most computing resources, maximum reliability, and randomness at the time of backup. They do not consider the limitations of bandwidth resources. Therefore, these three algorithms occupy more bandwidth than the Q-learning based algorithm. The comparison of computing resource occupation rate is shown as Fig. 8(b). We can observe that compared with other algorithms, the Q-Learning based algorithm occupies less computing resources, but the QL-S algorithm occupies more computing resources. This is because that the sequential SFCs require more VNFs to be backed up for the same reliability, which may result in more computing resources occupied.

It is obvious that the Q-Learning based algorithm can occupy the least bandwidth and computing resources, obtain the greatest reliability improvement, and effectively utilize the remaining resources in the actual physical network topology.

4) THE AVERAGE EXECUTION TIME OF NETWORK FUNCTION

The average execution time of network function for different algorithms is compared in Fig. 9. We can observe that in the same environment, the QL-P algorithm can reduce the average execution time by 20% compared with the LRF, MCRF, and Random algorithms. The average execution time of the QL-S algorithm is 12.5% lower than that of the QL-P algorithm. The average execution time of the QL-P algorithm and the QL-S algorithm is relatively stable. The average execution time is a part of the value of the reward function. When selecting a backup node, the node with the least delay will be considered as the deployment node. As the parallel network function needs to consider the parallel link during the backup, so the average execution is larger than the sequential network function. For LRF, MCRF and Random algorithms, when the number of the SFC requests is small, which means that resources are sufficient to provide backup service, the average execution time is stable. As the number of SFC requests to be backed up increases, the average execution time decreases due to the resources are insufficient to provide backup service. It is because that based on the definition of average delay, when no backup is selected for a VNF, the average delay is equal to the average delay of the working VNF. Thus, the backups cannot be completed when there are insufficient resources and the average delay is reduced. Obviously, the QL-P algorithm can obtain a smaller average delay, and when a network function is migrated to a backup physical node for execution, the end-to-end delay difference is smaller.

VI. CONCLUSION

In this paper, we have addressed the requirements of low-latency and high-reliability scenarios for 5G networks. We have considered the issue of improving reliable backup of parallel SFC, and proposed a Q-learning-based backup mechanism to obtain the optimal backup solutions in resource-constrained networks. To satisfy the high reliability requirement, the backup deployment of parallel SFC is firstly considered by jointly considering the impact of reliability improvement, resource consumption, as well as the impact of additional delays, brought by selecting a VNF backup node. Simulation experiments show that the algorithm can meet the reliability requirements of low-latency and high-reliability scenarios with the lowest resource consumption, and considers the problem of increased delay caused by backup. Compared with traditional algorithms, Q-learning algorithm has great advantages and is suitable for the backup problem of parallel SFC.

REFERENCES

[1] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, "A survey on low latency towards 5G: RAN, core network and caching solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3098–3130, 4th Quart., 2018.

[2] L. Peng and M. C. Joined, "Analysis of 5G low latency technology," *Telecom Power Technol.*, 2018.

[3] J. Liu and Q. Zhang, "Offloading schemes in mobile edge computing for ultra-reliable low latency communications," *IEEE Access*, vol. 6, pp. 12825–12837, 2018.

[4] C. Sun, J. Bi, Z. Zheng, H. Yu, and H. Hu, "NFP: Enabling network function parallelism in NFV," in *Proc. Conf. ACM Special Interest Group Data Commun.*, Aug. 2017, pp. 43–56.

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

[6] J. Fan, Z. Ye, C. Guan, X. Gao, K. Ren, and C. Qiao, "GREP: Guaranteeing reliability with enhanced protection in NFV," in *Proc. ACM SIGCOMM Workshop Hot Topics Middleboxes Netw. Function Virtualization HotMiddlebox*, 2015, pp. 13–18.

[7] S. Bijwe, F. Machida, S. Ishida, and S. Koizumi, "End-to-end reliability assurance of service chain embedding for network function virtualization," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2017, pp. 1–4.

[8] 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 Manage.*, vol. 14, no. 3, pp. 554–568, Sep. 2017.

[9] L. Qu, M. Khabbaz, and C. Assi, "Reliability-aware service chaining in carrier-grade softwarized networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 558–573, Mar. 2018.

[10] M. Liu, G. Feng, J. Zhou, and S. Qin, "Joint two-tier network function parallelization on multicore platform," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 3, pp. 990–1004, Sep. 2019.

[11] J. Sun, G. Zhu, G. Sun, D. Liao, Y. Li, A. K. Sangaiah, M. Ramachandran, and V. Chang, "A reliability-aware approach for resource efficient virtual network function deployment," *IEEE Access*, vol. 6, pp. 18238–18250, 2018.
[12] A. Engelmann and A. Jukan, "A reliability study of parallelized VNF chaining," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.
[13] D. Pham and D. Karaboga, *Intelligent Optimisation Techniques: Genetic Algorithms, Tabu Search, Simulated Annealing and Neural Networks*. Springer, 2012.
[14] H. Qing, Z. Weifei, and L. Julong, "Virtual network protection strategy to ensure the reliability of SFC in NFV," in *Proc. 6th Int. Conf. Inf. Eng. (ICIE)*, 2017, p. 17.
[15] E. Stevens-Navarro, Y. Lin, and V. W. S. Wong, "An MDP-based vertical handoff decision algorithm for heterogeneous wireless networks," *IEEE Trans. Veh. Technol.*, vol. 57, no. 2, pp. 1243–1254, Mar. 2008.
[16] M. Mechtri, C. Ghribi, and D. Zeghlache, "A scalable algorithm for the placement of service function chains," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 3, pp. 533–546, Sep. 2016.
[17] J. Xia, Z. Cai, and M. Xu, "Optimized virtual network functions migration for NFV," in *Proc. IEEE 22nd Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2016, pp. 340–346.

JIANHONG ZHOU (Member, IEEE) received the M.Eng. degree in electronics and electrical engineering from Nanyang Technological University (NTU), Singapore, in 2008, and the Ph.D. degree in computer software and theory from the University of Chinese Academy of Sciences, in 2016. She is currently an Associate Professor with the School of Computer and Software Engineering, Xihua University, China. She also holds a postdoctoral position with the National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China (UESTC). Her research interests include next generation cellular networks, the IoT, low-latency ultra-reliable communication, artificial intelligence, and so on.

GANG FENG (Senior Member, IEEE) received the B.Eng. and M.Eng. degrees in electronic engineering from the University of Electronic Science and Technology of China (UESTC), in 1986 and 1989, respectively, and the Ph.D. degree in information engineering from The Chinese University of Hong Kong, in1998. He joined the School of Electric and Electronic Engineering, Nanyang Technological University, in December 2000, as an Assistant Professor and promoted as an Associate Professor, in October 2005. He is currently a Professor with the National Laboratory of Communications, UESTC. He has extensive research experience and published widely in computer networking and wireless networking research. His research interests include resource management in wireless networks, next generation cellular networks, and so on.

YI GAO (Member, IEEE) received the B.Eng. and M.Eng. degrees in communication and information system from the University of Electronic Science and Technology of China (UESTC), in 2016 and 2019, respectively. His research interests include next generation cellular networks, network function virtualization, ultra reliable and low-latency communication, and the industrial IoT.