applied sciences

MDPI

Article

Joint Deployment Optimization of Parallelized SFCs and BVNFs in Multi-Access Edge Computing

Ying Han, Junbin Liang * and Yun Lin

Guangxi Key Laboratory of Multimedia Communications and Network Technology, School of Computer and Electronics Information, Guangxi University, Nanning 530004, China; hailin@st.gxu.edu.cn (Y.H.); 2007310439@st.gxu.edu.cn (Y.L.)
* Correspondence: liangjb@gxu.edu.cn

Abstract: In multi-access edge computing (MEC) networks, parallelized service function chains (P-SFCs) can provide low-delay network services for mobile users by deploying virtualized network functions (VNFs) to process user requests in parallel. These VNFs are unreliable due to software faults and server failures. A practical way to address this is to deploy idle backup VNFs (BVNFs) near these active VNFs and activate them when active VNFs fail. However, deploying BVNFs preempts server resources and decreases the number of accepted user requests. Thus, this paper proposes a reliability enhancement approach that uses BVNFs satisfying the delay requirement as active VNFs to form P-SFCs, which contributes to the delay reduction and reliability enhancement. Since the resource capacities of edge servers can only deploy a certain number of P-SFCs and BVNFs, establishing how to deploy the minimum number of P-SFCs and BVNFs to satisfy the delay and reliability requirements of mobile users and maximize the number of accepted user requests is a challenging problem. In this paper, we first model the dynamics of delay and reliability caused by VNF parallelization and BVNFs deployment, then formulate the joint deployment problem of P-SFCs and BVNFs. Next, we design an approximation algorithm to deploy critical VNFs and BVNFs on a target edge server and schedule the data traffic of user requests processed by P-SFCs. Experimental results based on real-world datasets show that our proposed algorithm outperforms two benchmark algorithms in terms of throughput, delay, reliability, and resource utilization.

Keywords: network function virtualization; parallelized SFCs; BVNFs; delay and reliability

Citation: Han, Y.; Liang, J.; Lin, Y. Joint Deployment Optimization of Parallelized SFCs and BVNFs in Multi-Access Edge Computing. *Appl. Sci.* **2023**, *13*, 7261. https://doi.org/10.3390/app13127261

Academic Editors: Ireneusz Kubiak, Tadeusz Wieckowski and Yevhen Yashchyshyn

Received: 11 May 2023
Revised: 15 June 2023
Accepted: 16 June 2023
Published: 18 June 2023

1. Introduction

The multi-access edge computing (MEC) network, which benefits from proximity communication with mobile users and extensive service convergence of edge servers, is envisioned as a promising distributed network [1–3]. The MEC network can provide low-delay services to mobile users by leveraging the closest edge server to process user requests. To ensure the security and reliability of communication transmission before reaching the closest edge servers, user requests typically steer their traffic flows along a sequence of dedicated equipment that implements certain network functions (e.g., firewalls, packet inspection, and intrusion prevention/detection systems). However, this equipment needs to be manually "patched" into the existing network infrastructure, with fixed locations and significant expenditure [4]. This method of implementing network functions brings challenges in the flexible management of network service and the rapid development of network functions.

Network function virtualization (NFV) decouples network functions from dedicated hardware and virtualizes network functions as software implementations known as virtual network functions (VNFs) [5,6]. Different network services requested by mobile users can be provided in the form of a service function chain, which is a chain of VNFs. Thus, NFV offers a flexible way to design, place, and manage network services by deploying VNFs

on edge servers in the MEC network and chaining them into different SFCs [7]. Mobile users generally carry large traffic data that need to be processed by certain SFCs with strict reliability and delay requirements. However, SFCs often incur a high transmission delay since VNFs in SFC are deployed in different edge servers, and user requests need to traverse these VNFs sequentially [8,9]. Splitting the user request into multiple sub-flows can decrease transmission delay significantly. One of the key issues in realizing splitting is establishing how to route these sub-flows along VNFs and map these VNFs onto the underlying MEC network. Parallelized SFCs (P-SFCs) can deploy two or multiple VNF instances for each VNF in SFC to process user requests in parallel [10,11]. These parallel VNF instances form a set of P-SFCs, and each P-SFC is also a sequence of VNFs. Thus, the user request can be split into multiple sub-flows that traverse these P-SFCs in parallel.

In the MEC network, due to software faults and server failures (e.g., operational error and server overload), active VNFs that process user requests are prone to failure. The failure of any single VNF in SFC will interrupt the network service and lead to significant economic losses and threats to individual safety [12,13]. Additionally, the reliability of SFC decreases significantly as the number of VNFs in SFC increases [14]. Deploying idle backup VNFs (BVNFs) near the active VNFs and activating BVNFs when the active VNFs fail is a practical way to satisfy the reliability requirement of user requests and continuously maintain the network services. That is to say, if the reliability of P-SFCs is below the reliability requirement, extra BVNFs must be deployed until the service reliability provided by P-SFCs and BVNFs satisfies the reliability requirement.

Deploying idle BVNFs will preempt server resources available to process incoming user requests and then reduce the number of accepted user requests. Reducing the number of idle BVNF deployments is key to mitigating this problem. In this paper, we propose that BVNFs satisfying the delay requirement can be used as active VNFs to form P-SFCs. This approach can enhance reliability, as the special BVNFs can also provide network services when VNFs fail. Meanwhile, if one active VNF instance in P-SFCs fails, other parallel instances may replace it to provide network service while satisfying the delay requirements. These parallel VNFs can be used as BVNFs, thus reducing the deployment number of idle BVNFs and enhancing the reliability of P-SFCs. Thus, the deployment location and number of P-SFCs influence the deployment number of idle BVNFs and vice versa. Moreover, since they both preempt resources to enhance reliability, the increase in the number of parallel VNFs will reduce the resources available to deploy BVNFs, and vice versa. However, the resource capacities of edge servers can only deploy a certain number of P-SFCs and BVNFs. Thus, establishing how to optimally deploy P-SFCs and BVNFs to maximize the throughput of user requests while satisfying their delay and reliability requirements is a challenging problem.

In this paper, the throughput maximization problem is first formulated as an integer linear programming problem. Due to the high computational complexity, we devise a proactive approximation algorithm to maximize the throughput by trading off the resource preemption of P-SFCs and BVNFs. Then, we evaluate the performance of our proposed algorithm through experimental simulations. Experimental results demonstrate that our proposed algorithm outperforms other counterpart algorithms in terms of delay, reliability, and throughput. The contributions of this paper can be summarized as follows:

- We propose a new approach to enhance the reliability of P-SFCs and reduce the deployment number of idle BVNFs. Then, we model the dynamics of delay and reliability caused by the deployment of P-SFCS and BVNFs and formulate the throughput maximization problem;
- We design an approximation algorithm to find a near-optimal solution. It defines delay-reduction priority and reliability-enhancement priority to identify the critical VNF in SFC and designs two schemes to deploy P-SFCs and BVNFs and steer user requests traversing these VNF instances;
- To evaluate the effectiveness of our proposed algorithm, we conduct extensive simulation experiments based on real-world datasets from the central business district of

Melbourne, Australia. Experimental results show that compared to two benchmark algorithms (see Section 5), our proposed algorithm can significantly decrease the delay and increase throughput while improving resource utilization.

The remainder of this paper is organized as follows. Section 2 presents recent works. Section 3 presents the system model and formulates the problem. Section 4 designs an approximate algorithm for the throughput maximization problem. The performance of our proposed algorithms is empirically evaluated in Section 5, and Section 6 concludes the paper.

2. Recent Works

Since MEC and NFV appeared, many researchers in the industry and research community have worked on how to provide low-delay and reliable network services for mobile users by deploying SFCs and BVNFs on edge servers. We review the recent work about the deployment of P-SFCs and BVNFs as the joint deployment of SFC and BVNFs with parallelization or not.

2.1. Joint Deployment of SFC and BVNFs without Parallelization

Huang et al. [15] studied a reliability-aware VNF deployment problem by deploying active and backup VNF instances at different cloudlets in the mobile edge network and devised an approximation algorithm. Li et al. [14] assumed that user satisfaction with a service is heavily impacted by the service's reliability and delay. They formulated a user satisfaction problem aimed at maximizing the accumulative user satisfaction and devising an approximation algorithm for it.

Shang et al. [16] studied the highly available and cost-effective SFC deployment problem under edge resource limitations and time-varying VNF failures. They proposed a reliability-aware adaptive deployment scheme to deploy SFCs, static backup, and dynamic backup to guarantee reliability. Li et al. [17] assumed that the reliability of VNF instances is affected by the software implementation, execution duration, workload among edge servers, and so on. They predicted reliability based on digital twin techniques and proposed two optimization problems of reliable service provisioning: the service cost minimization problem and the dynamic service admission maximization problem. Rui et al. [18] proposed an SFC reliability evaluation method and reliability optimization algorithm. They analyzed all of the potential influencing factors for reliability, including resource preemption, common cause failure, fault recovery and redundant backup, and execution time. Qiu et al. [19] assumed that the fault probability of each VNF is dynamic and fluctuates according to time and workload. They proposed a long-term provisioning problem to maximize the throughput of receiving requests while minimizing receiving costs and solved it via an online approximation scheme. Liang et al. [20] studied a novel service reliability augmentation problem for each admitted request and designed a deterministic heuristic for the problem without any resource violation.

Yang et al. [21] focused on a dynamic network environment where a user request arrived randomly and formulated the maximization problem of the number of accepted user requests. Karimzadeh–Farshbafan et al. [22] formulated dynamic reliability-aware service placement as an infinite horizon Markov decision process to minimize the placement cost and maximize the number of accepted user requests. They designed an iteration algorithm to find the optimal policy. Unlike Karimzadeh–Farshbafan et al. [22], Li et al. [23] assumed that requests arrived into the system one by one without the knowledge of future arrivals and proposed a reliable VNF provisioning problem. They developed two efficient online algorithms under two different backup schemes: on-site (local) and off-site (remote) schemes. Jia et al. [24] formulated the problem of SFC scheduling in a dynamic 5G environment aiming to maximize the number of accepted user requests. They first proposed an algorithm to decide the redundancy of VNFs while minimizing delay. Then, they designed a reinforcement learning approach for SFC scheduling to increase the success rate of SFC requests.

2.2. Joint Deployment of SFC and BVNFs with Parallelization

There are extensive studies on the joint deployment problem of P-SFCs and BVNFs with parallelization while satisfying delay and reliability requirements in MEC networks. We classify the related studies into two categories: parallelism with traffic duplication and parallelism with traffic splitting.

Parallelism with traffic duplication leverages all active VNF instances and VNF replicas to construct a primary SFC and multiple backup SFCs. Then, it transmits multiple duplications of user requests along multiple instances. Unlike re-routing to a reliable VNF in case an active VNF fails, this approach directly leverages VNF backups to increase the service's reliability, as seen in [25,26]. For example, Qu et al. [25] considered the trade-off of reliability, delays, and resource consumption. To satisfy the delay and reliability requirements and minimize the communication bandwidth usage, they deployed multiple VNF backups and duplicate flow to traverse all active VNF instances and VNF backups. To further improve resource usage, Qu et al. [26] proposed a VNF deployment problem while supporting VNF decomposition and hybrid multiple-path routing and designed a heuristic algorithm to solve the problem. However, the algorithm wastes resources to process multiple traffic duplications. Therefore, it is suitable in cases where the length of SFC is short and the reliability of VNF is not high.

Parallelism with traffic splitting allows user requests to be split into multiple subflows that are processed in parallel by multiple SFCs, thus reducing the service delay. Many studies have been devoted to parallelism with traffic, such as [27,28]. For example, Engelmann et al. [27] studied end-to-end service reliability with flow and SFCs parallelism. They defined four different backup deployment strategies and analyzed the service reliability of these deployment strategies. Their results show that parallelism significantly increases service reliability and resource utilization compared to serial processing. However, these strategies increase reliability at the cost of resources. To further improve service reliability and reduce resource consumption, they [28] studied service reliability with erasure coding and the failures of the path segment, VNF, and server failures. They derived analytical expressions for service reliability with encoding or VNF redundancy and showed that the combination of both methods increases service reliability. These aforementioned studies analyzed the service reliability of SFC in detail; however, they did not study the reliability-guaranteed problem under a multi-user resource-constrained environment and did not consider the service delay.

Kianpisheh et al. [29] proposed a parallel VNF processing approach through pools of candidate servers that host VNFs processing the traffic. Considering parallel VNF processing without backup VNFs decreases reliability and incurs additional communication and process costs. Promwongsa et al. found [10] focused on the problem of minimizing the sum of reliability degradation and operational costs while satisfying delay requirements. Due to the complexity of this problem, they proposed a Tabu search-based algorithm to find sub-optimal solutions. Wang et al. [30] also focused on the parallelized SFC deployment problem to minimize physical link consumption. They designed three deployment strategies and a hybrid deployment algorithm to solve this problem.

Unlike VNF decomposition, which reduces the computing resources of VNF replicas in [26], Alleg et al. [31] considered diversity which splits a single active VNF into a pool of active instances. Then, they proposed a joint selective diversity and tailored redundancy mechanism to provide resilient service. Based on this mechanism, they proposed the deployment problem of SFC to minimize resource consumption and proposed three solutions to solve the problem.

To address the reliability-aware, delay-guaranteed, and resource-efficient SFC deployment problem, Thiruvasagam et al. [32] proposed a novel SFC sub-chaining method to enhance service reliability. Then, they formulated the reliable SFC deployment problem of minimizing the number of active physical nodes allocated for SFCs deployment. Given the high computational complexity, they proposed a modified stable matching algorithm to provide a near-optimal solution. However, the failure of any VNF of these sub-flows will

interrupt service provision and decrease the service reliability. Moreover, to process these sub-flows and improve reliability, it needs to deploy multiple active VNF instances and VNF replicas consuming massive resources.

2.3. Our Research

Due to the flexibility of traffic scheduling and the advantages of parallelism, this approach of splitting user requests into multiple sub-flows can flexibly schedule traffic and balance load, which is suitable for satisfying the requirements of delay-sensitive applications in the MEC network. However, the work above only considers that the reliability decreases significantly as the number of parallel VNFs increases or only that the parallel VNFs can directly replace failure VNFs. The former ignores the situation where parallel VNFs can improve reliability, and the latter does not consider whether the delay requirements are satisfied after replacing the failed VNFs. Based on our discussions above, we propose a new scheme to reduce delay and enhance the reliability of P-SFCs and formulate the problem of maximizing throughput.

3. System Model and Problem Statement

3.1. Network Topology

The NFV-based MEC network consists of several edge servers, base stations, and mobile users, as shown in Figure 1. We model the network as an undirected graph $G = (BS \cup V,\ E)$, where BS is a set of base stations, and E represents all physical links between base stations [33]. $E_{i,j}$ indicates the physical links between edge server V_i and V_j with a transmission rate $B_{i,j}$. V is a group of edge servers $V_i \in V$ with a resource capacity C_i. We assume that there are $|F|$ types of VNFs in the network with different network functions. The set of different VNFs is denoted as $F = \left\{f_1,\ f_2, \ldots,\ f_{|F|}\right\}$. The resource demand, processing capacity, and reliability of each VNF $f_n \in$ F are C_{f_n}, P_{f_n}, and R_{f_n}, respectively. To facilitate discussion, we use the number of resource units to represent resource demand, which can be obtained by the weighted sum of the number of different resources (e.g., CPU cores, storage). The reliability of each VNF is characterized in terms of mean time between failure (MTBF) and mean time to repair (MTTR), i.e., MTBF/(MTBF + MTTR). Then, we assume that the active VNF instances and BVNFs of each VNF in SFC have the same resource demand, processing capacity, and reliability.

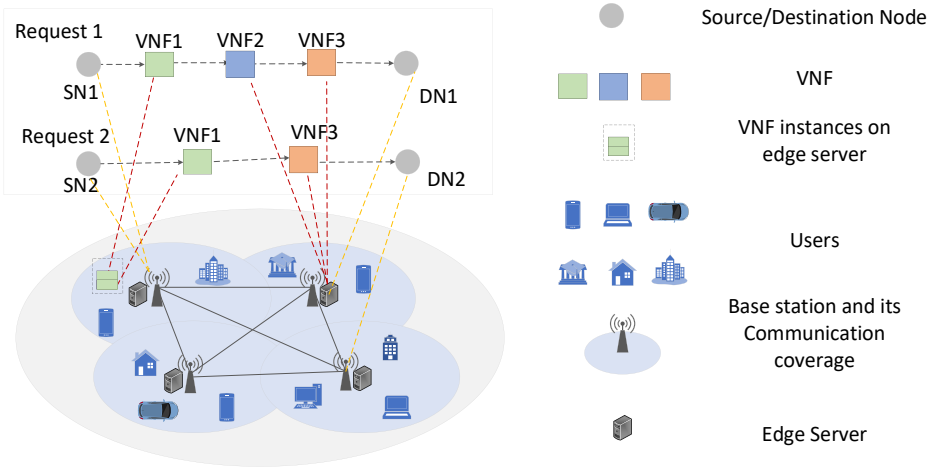

Figure 1. The MEC network. Basic settings: There are two user requests, r_1 and r_2, with two SFCs, i.e., $SFC_1\langle VNF_1, VNF_2, VNF_3\rangle$ and $SFC_2\langle VNF_1, VNF_3\rangle$. The source and destination nodes of these two user requests are BS_1 and BS_2 and BS_1 and BS_4, respectively. For user request r_1, the VNF_1 instance is deployed in edge server V_1, with the VNF_2 instance and the VNF_3 instance in edge server V_2. Thus, user request r_1 can obtain the required network service by traversing the instances in the edge servers in order.

Then, we define that the user request from mobile user U_i is represented by a tuple $r_i = \langle s_i, d_i, D_i, R_i, SFC_i, t_i, n_i \rangle$, where s_i is the source node, i.e., the nearest base station where U_i sends user request r_i, d_i is the destination node of user request r_i, i.e., the edge server which processes user request r_i and return results, D_i is the end-to-end delay requirement, R_i is the reliability requirement with $0 < R_i$ 1, SFC_i is the requested VNF chain, which consists of L_i network functions VNF^1, $VNF^2, \ldots, VNF^{L_i}$ in order, and t_i is the data size of user request r_i. The set of user requests from all mobile users is denoted as R. We also define n_i as the maximum number of sub-flows that each traffic flow of r_i can be split into. These sub-flows can merge to traverse the same VNF and be split to traverse different VNFs flexibly. As an assumption, when a mobile user is located in the overlapping area of the communication coverage of multiple base stations, it sends user requests to the nearest base station. We also assume that there is only one user request associated with each mobile user. Note that our work can be extended to deal with the case where there are multiple user requests from each mobile user by treating them as multiple user requests from different virtual users, and all of these virtual users are the aforementioned mobile user. Key notations used in this paper are summarized in Table 1.

Table 1. List of key notations in this paper.

Notation	Definition
$(BS \cup V, E)$	The MEC network
$E_{i,j}$	The physical links between edge servers V_i and V_j
C_i	The resource capacity of edge servers V_i
F	The set of different VNFs
C_{f_n}, P_{f_n} and R_{f_n}	The resource demand, processing capacity, and reliability of each VNF f_n
$\langle s_i, d_i, D_i, R_i, SFC_i, t_i, n_i \rangle$	seven tuple definition: s_i is the source node; d_i is the destination node of user request r_i; D_i is end-to-end delay requirement; R_i is the reliability requirement; SFC_i is requested VNF chains; and t_i is data size of user request r_i; n_i is the maximum number of sub-flows
$X_{i,k}^{l,j}$	Indicate the deployment of VNF^l in SFC_i on the edge server V_j for processing sub-flow k of user request r_i
$Y_{l_e}^{k,i}$	Indicate whether sub-flow k of traffic flow r_i pass through link $E_{j,h}$
$Z_{i,l,j}$	Indicate if BVNFs for VNF^l of traffic flow r_i is deployed on edge server V_j

3.2. P-SFCs and BVNFs Deployment

Each user request can be accepted under the following conditions: (i) starting from the source node, the user request traverses P-SFCs and reaches the destination node. The overall time does not exceed the delay requirement; (ii) the service reliability provided by P-SFCs and BVNFs must satisfy the reliability requirement. Therefore, when the mobile user sends their user request, P-SFCs and BVNFs are deployed in edge servers near users and process incoming user requests while satisfying delay and reliability requirements. To model the service reliability and delay of P-SFCs and BVNFs, we first define the deployment variables of P-SFC and BVNFs.

The binary variable $X_{i,j}^{l,k}$ is used to present the deployment of VNF^l in SFC_i on the edge server $V_j \in V$ for processing sub-flow k of user request r_i such that:

$$X_{i,j}^{l,k} = \begin{cases} 1, & \text{if } VNF^l \text{ of } SFC_i \text{ is deployed on } V_j \text{ for sub-flow k of } r_i \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

We use the discrete variable $x_{i,j}^{l,k}$ to indicate the resource demand of VNF^l instances that are deployed in V_j for processing sub-flow k of user request r_i. We also use the discrete variable $r_{i,j}^{l,k} \in [0,\ t_i]$ to indicate the data size of sub-flow k, which is processed by VNF^l instances on edge server V_j.

Considering the traffic routing, we next use a binary variable, whether sub-flow k of traffic flow r_i passes through link $E_{j,h} \in E$ such that:

$$Y_{i,k}^{j,h} = \begin{cases} 1, & \text{if sub-flow } k \text{ of traffic flow } r_i \text{ passes through link } E_{j,h} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

To record the number and preemption resources of BVNFs for traffic flow r_i deployed in edge server $V_j \in V$, we use a binary variable $Z_{i,j}^l$ to indicate if BVNFs for VNF^l of traffic flow r_i are deployed in the edge server $V_j \in V$ such that:

$$Z_{i,j}^l = \begin{cases} 1, & \text{if BVNFs for } VNF^l \text{ of } r_i \text{ is deployed on edge server } V_j \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

We also use the discrete variable $z_{i,j}^l$ to indicate the resource demand of BVNF instances for VNF^l deployed in V_j.

Next, we model the reliability and delay of P-SFCs and BVNFs. Note that the VNF instance failures are independent and do not impact each other. In the MEC network, the reliability of all VNF^l instances in SFC is positively correlated with the incoming traffic size and processing capacities. First, we define all instances of VNF^l as an instance set, record the length of it as $|N_{a,l,i}|$, and obtain all non-repeating subsets of length $1, 2, \ldots, |N|$. Next, we calculate the end-to-end delay when these instances in a subset process a user request simultaneously with all VNF instances of other $VNF^l \in SFC_i$. If the delay satisfies the delay requirement, we define the subset of these instances as a feasible set and delete any remaining subset that contains all instances of the feasible set. Then, we calculate the reliability of all feasible sets. The operation example of calculating the reliability $r_{VNF^{l,i}}$ of VNF^l on SFC_i is shown in Figure 2. To avoid calculating reliability repeatedly, it is necessary to obtain an ordered permutation of all feasible sets and then find the intersection of feasible sets. For example, the intersection of two feasible sets, i.e., [*V4, V5, V6*] and [*V4*, *V5*, *V7*], is [*V4*, *V5*]. The reliability of these two sets is equal to the reliability of [*V4*, *V5*] times the reliability of [*V6*, *V7*]. Since both *V4* and *V5* must remain active, only one of *V6* and *V7* remain active, and the reliability of [*V6*, *V7*] is equal to 0.96 (i.e., $1 - 0.2^2$). The reliability of [*V4*, *V5*] is equal to 0.00512 (i.e., $0.2^* \times 0.2 \times 0.2 \times 0.8 \times 0.8$). The value of 0.2^3 is to ensure that *V1*, *V2*, and *V3* fail and avoid the intersection between the reliability of these two sets, and the feasible set contains *V1*, *V2*, and *V3*.

The reliability of P-SFCs is the product of all active instances of P-SFCs. The service reliability of P-SFCs for SFC_i is denoted as R_{P-SFC_i}. The accumulative reliability of R_{P-SFC_i} is as follows.

$$R_{P-SFC_i} = \sum_{p=1}^{L_i} r_{VNF^{p,i}} \tag{4}$$

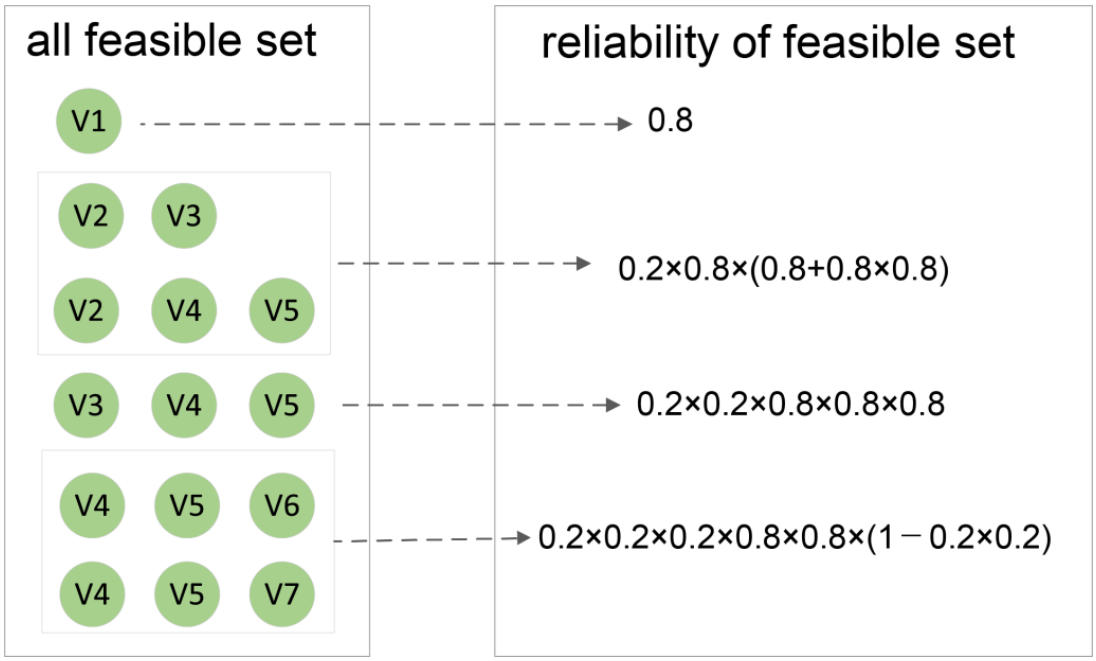

Figure 2. The operation to calculate reliability. The reliability of any single VNF instance is 0.8. All feasible sets are sequentially sorted by the ID of edge servers in each column while ensuring that the elements in each row are unchanged. The overall reliability $r_{VNF^{l,i}}$ of VNF^l on SFC_i is the sum of the reliability of all feasible sets.

When there are $\left|N_{a,l,i}\right|$ BVNFs for VNF^l in SFC_i, let $r_{i,l,1}, r_{i,l,2}, \ldots, r_{i,l,N_{a,l,i}}$ be their reliability, respectively. The accumulative reliability R_{SFC_i} of user request r_i is, therefore:

$$R_{SFC_i} = \sum_{p=1}^{L_i} \left(1 - \left(1 - r_{VNF^{p,i}}\right) \sum_{j=1}^{N_{a,l,i}} \left(1 - r_{i,l,j}\right) \right) \tag{5}$$

Here, we consider two kinds of delay, namely, the transmission delay along multiple P-SFCs between the source node and the destination node and the processing delay in each edge server. The overall processing delay of sub-flow k of traffic r_i at all edge servers is denoted as $d_{pro}(f_{i,k})$. Let $P(s_i, d_i,\ k)$ be the routing path for sub-flow k of user request r_i between the source node and the destination node. The transmission delay along the path $P(s_i, d_i,\ k)$ is $d(P(s_i, d_i, k)) = \sum_{E_{j,h} \in P(s_i, d_i, k)} d\left(B_{j,h}\right)$, where $d(E_{j,h})$ is the transmission delay of sub-flow k on link $E_{j,h} \in E$. The accumulative delay of user request r_i is, therefore:

$$D_{SFC_i} = max\left\{d(P(s_i, d_i, k)) + d_{pro}(f_{i,k})\right\} \tag{6}$$

3.3. Problem Statement

Now, we describe all constraints related to the deployment of P-SFCs and BVNFS.

Constraint 1: Ensures that all sub-flows k of traffic flow r_i are processed by the ordered SFC.

$$\sum_{k \in r_i} \sum_{VNF^l \in SFC_i} r_{i,j}^{l,k} = 1, \forall VNF^l \in SFC_i, \forall r_i \in R \tag{7}$$

Constraint 2: Ensures that the number of sub-flows cannot exceed the maximum number of sub-flows for user request r_i. That is, the number of all instances for VNF^l cannot exceed the maximum number.

$$0 \leq \sum_{V_j \in V} X_{i,j}^{l,k} \leq n_i,\ \forall VNF^l \in SFC_i, \forall r_i \in R \tag{8}$$

Constraint 3: For each user request, if it can be successfully accepted, we should make sure the end-to-end delay will not exceed its delay requirement.

$$max\{d(P(s_i, d_i, k)) + d_{pro}(f_{i,k})\} \leq D_i, \forall r_i \in R \tag{9}$$

Constraint 4: For each user request, if it can be successfully accepted, we should make sure the reliability is no less than its reliability requirement.

$$R_{SFC_i} \geq R_i, \forall r_i \in R \tag{10}$$

Constraint 5: To guarantee that the total resource preemption of each edge server V_j does not exceed the resource capacity C_j, we have:

$$\sum_{r_i \in R} \sum_{k \in r_i} \sum_{VNF^l \in SFC_i} X_{i,j}^{l,k} x_{i,j}^{l,k} + Z_{i,j}^{l} z_{i,j}^{l} \leq C_j, \forall V_j \in V \tag{11}$$

We use $x_i = 1$ to represent the fact that the user request r_i is successfully accepted. Given a MEC network $G = (BS \cup V, E)$, in which a set U of mobile users with each mobile user $i \in U$ sends a user request r_i, the network throughput is $\sum_{r_i \in R} x_i$, and the throughput maximization problem can be formulated as follows:

$$\begin{gathered} max \sum_{r_i \in R} x_i \\ s.t.\ (7), (8), (9), (10), (11) \end{gathered} \tag{12}$$

4. Proposed Solution

Since the throughput maximization problem is computationally expensive to derive the optimal solution, we design an approximation algorithm to solve the problem. To minimize resource preemption, the approximation algorithm aims at deploying parallel VNFs to form P-SFCs with low delay and high reliability and the minimum number of BVNFs to satisfy reliability requirements. The pseudo-code of our proposed algorithm is outlined in Algorithm 1.

4.1. Throughput Maximization Algorithm

To maximize throughput efficiently, we design two schemes to deploy P-SFCs and BVNFs with the highest performance enhancement. First, to decrease the end-to-end delay and the resource preemption of P-SFCs, a delay-reduction priority is defined to measure the priority of deploying parallel VNF instances for each VNF in SFC. The priority for VNF^l on SFC_i is defined as follows:

$$P_{i,l}^{d} = max\{d(P(p_{l-1}, p_l, k)) + d_{pro}(f_{l,k}) + d(P(p_l, p_{l+1}, k))\} \tag{13}$$

where $d(P(p_{l-1}, p_l, k))$ is the transmission delay of sub-flow k traversing from edge server p_{l-1} deploying VNF^{l-1} instances to the next edge server p_l deploying VNF^l instance.

To ensure that deploying parallel VNFs can obtain high delay reduction and the total data size of these sub-flows remains the same, we design a delay-reduction scheme to select VNFs and edge servers to deploy parallel VNF instances and determine the split and merge of user requests.

1. If VNF^l has the highest priority and the number of parallel VNF^l instances does not exceed the maximum number of sub-flows, we select VNF^l as the critical VNF to obtain the high potential of delay reduction by deploying parallel VNF instances of VNF^l. In reverse, we select the next VNF with high priority as the critical VNF. The next step is to find a target sub-flow and split it into smaller sub-flows;
2. Then, we sort all sub-flows of user requests by the delay of sub-flows traversing from the previous server to the current server and from the current server to the next edge server. We select the sub-flow with the maximum delay as the target sub-flow. Next,

we record the previous server, current server, and next server that were traversed by the target sub-flow. We also record the communication distance to traverse these three servers as the target communication distance. The next step finds all feasible edge servers;

3. Next, we define a set of edge servers that deploy VNF^l instances traversed by the user request as a comparison set. Then, we select edge servers whose resource capacity exceeds resource demand from all edge servers except for edge servers in the comparison set. This operation ensures that the target sub-flow is split into two sub-flows and traverses different paths to decrease delay. Next, we insert these edge servers into a candidate set and record the transmission and processing capacities of different paths from the previous server to these edge servers and from these edge servers to the next server. Next, we find the target edge server to deploy a parallel VNF instance;
4. Then, we select the edge server with the highest capacity as the target server and define the flow that traverses it as the candidate flow. The next step splits and merges sub-flows to reduce delay;
5. To decrease delay, we split the target sub-flow into two sub-flows according to their transmission and processing capacities. We do not evenly split the size of the user request between all sub-flows and the candidate flow or the size of the target sub-flow between the target sub-flow and the candidate flow. Although the former can significantly reduce delay, it affects the delay and reliability of subsequent VNFs. The latter does not guarantee delay reduction if the communication distance of the candidate flow is too large.

Based on the above discussion, there are two special cases that require different methods of splitting and merging: (i) when the number of sub-flows traversing the previous server and the next edge server exceeds one, we split the total data size of these sub-flows. As they traverse the same next edge server, the method does not affect the subsequent VNFs; (ii) when the situation depicted in Figure 3c occurs, we must guarantee that the communication distance of the candidate flow is less than the communication distance of the target sub-flow. The different ways to split and merge sub-flows are shown in Figure 3. Additionally, the flowchart of the delay-reduction scheme is shown in Figure 4.

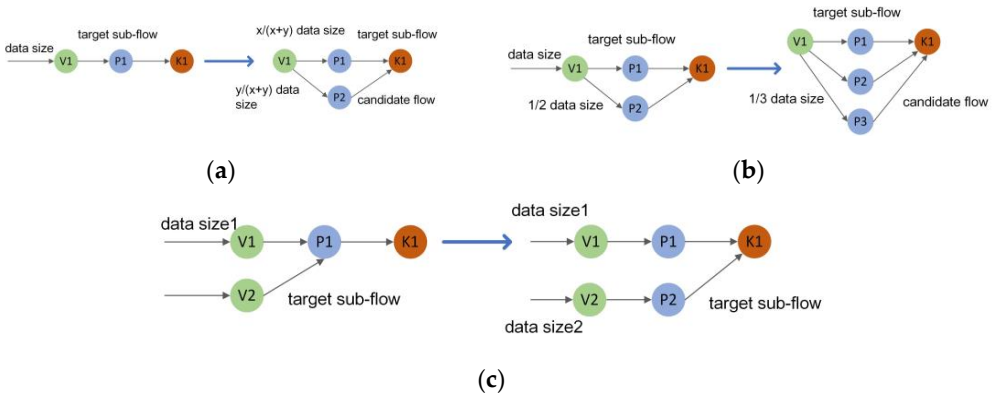

Figure 3. (**a**) Split target sub-flows; (**b**) merge and split multiple sub-flows; (**c**) schedule the traffic of target sub-flow to candidate flow. In Figure 3a, the sum of transmission and processing capacity of the target sub-flow and candidate flow is x and y, respectively.

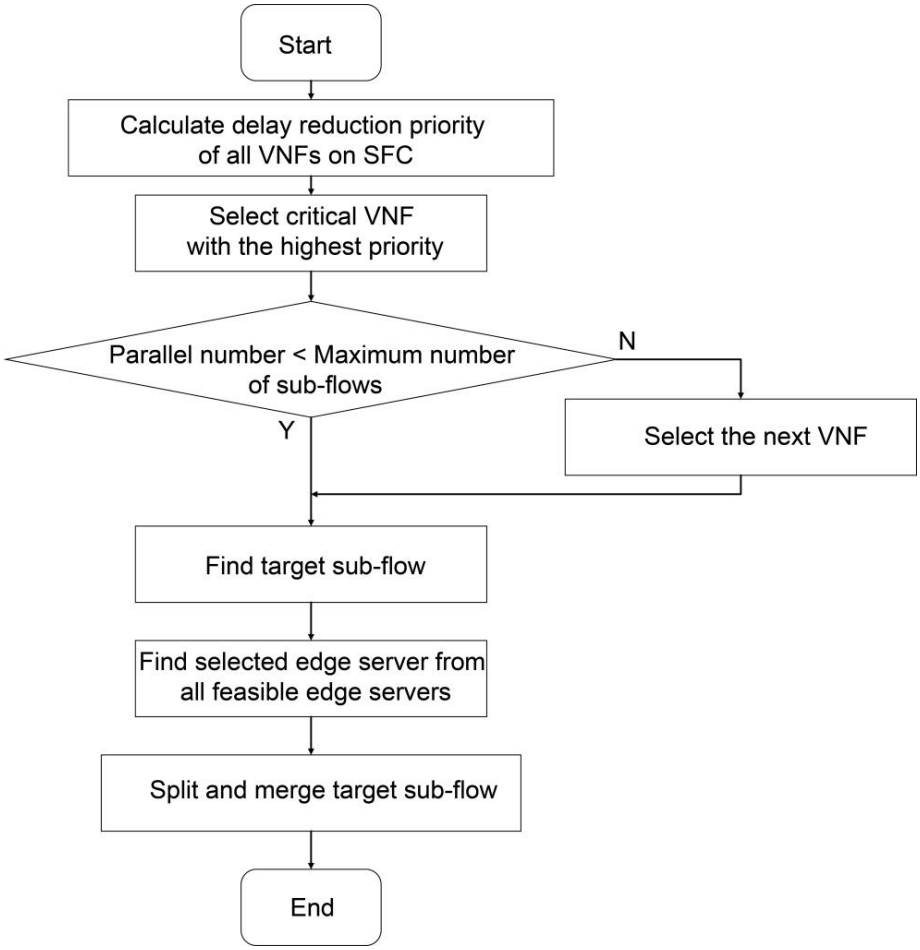

Figure 4. Explanation of the delay-reduction scheme in our algorithm.

Second, to reduce the number of BVNFs, a reliability-enhancement priority is defined to measure the priority of deploying parallel VNFs for each VNF in SFC. The priority for VNF^l on SFC_i is defined as follows:

$$P^r_{i,l} = 1/r_{VNF^{p,i}} \tag{14}$$

To ensure that deploying parallel VNFs can obtain high-reliability enhancement and the total size of these sub-flows remains the same, we design a reliability-enhancement scheme to select the VNF, sub-flow, and edge server.

1. If VNF^l has the highest priority and the number of parallel VNF^l instances does not exceed the maximum number of sub-flows, we select VNF^l as the critical VNF to obtain a high potential of reliability enhancement by deploying parallel VNF instances of VNF^l [25,34]. In reverse, we select the next VNF with high priority as the critical VNF. We execute the above steps 2–3 to find the target sub-flow and all feasible edge servers. Then, we sort feasible edge servers by their transmission and processing capacities. We record the comparison set that satisfies the delay requirement and the reliability set that consists of all edge servers which deploy parallel VNF^l instances used to enhance reliability. The next step is to find the target edge server to deploy parallel VNF instances;
2. Then, we define the feasible edge server with the highest capacity as the selected edge server. Next, we calculate the end-to-end delay when replacing one edge server in the comparison set one by one with the selected edge server. If the end-to-end delay satisfies the delay requirement, we replace other edge servers in the comparison set one by one with the edge server in the reliability set. Then, if the end-to-end delay

satisfies the delay requirement, we record the selected edge server and other edge servers as a feasible set and the number of all feasible sets. If the number exceeds the length of the comparison set, we define the selected server as the target server and the flow that traverses it as the candidate flow. In reverse, we select the next edge server with high priority as the selected edge server. Then, we execute the above step 5 to split and merge the target sub-flow.

Notice that to improve resource utilization, we control the length of the feasible set to be the length of the comparison set. The reason for this is that service reliability significantly decreases as the number of VNF instances that must simultaneously remain active to satisfy the delay requirement increases. Thus, if VNF^l instances need to process a user request simultaneously with many VNF instances, the reliability enhancement can be ignored. The operation example to obtain all feasible sets is shown in Figure 5. Additionally, the flowchart of the reliability-enhancement scheme is shown in Figure 6.

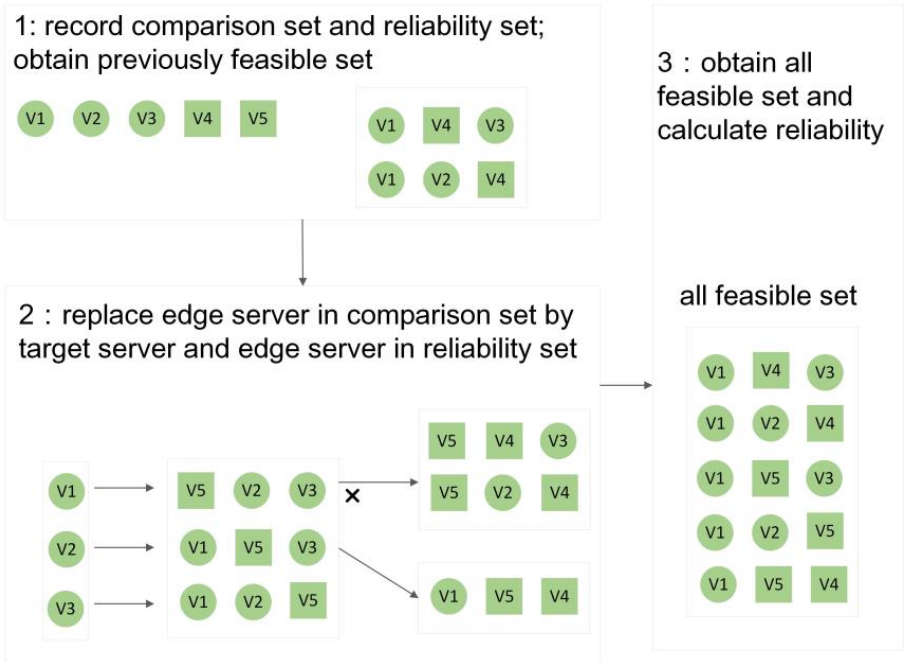

Figure 5. The operation to obtain all feasible sets and calculate reliability when we determine whether to deploy parallel VNF instances on target edge server *V5*. There are three edge servers [*V1, V2, V3*] in the comparison set and one edge server [*V4*] in the reliability set. We first replace the edge servers in the comparison set with *V5*, and then we replace the other edge servers with *V4*.

Based on two measures, we design a throughput maximization algorithm. The pseudo-code of our proposed algorithm is outlined in Algorithm 1.

This algorithm works as follows: first, it calculates the shortest path and communication distances among edge servers and the nearest base station to each mobile user by means of the Dijkstra Algorithm and the Euclidean Metric in lines 2–7. For user request r_i, it sorts edge servers by the communication distances between the edge server deploying VNF^{l-1} instances and all feasible edge servers. Then, it starts finding the nearest edge server for deploying one instance of VNF^l on SFC_i in order on lines 9–14. However, if there are no more VNF instances that can be deployed in any edge servers without violating their resource capacities, it terminates the loop, drops the user request r_i, and releases the resource it preempts. Otherwise, this procedure continues until all VNF instances are deployed. Then, it calculates the end-to-end delay.

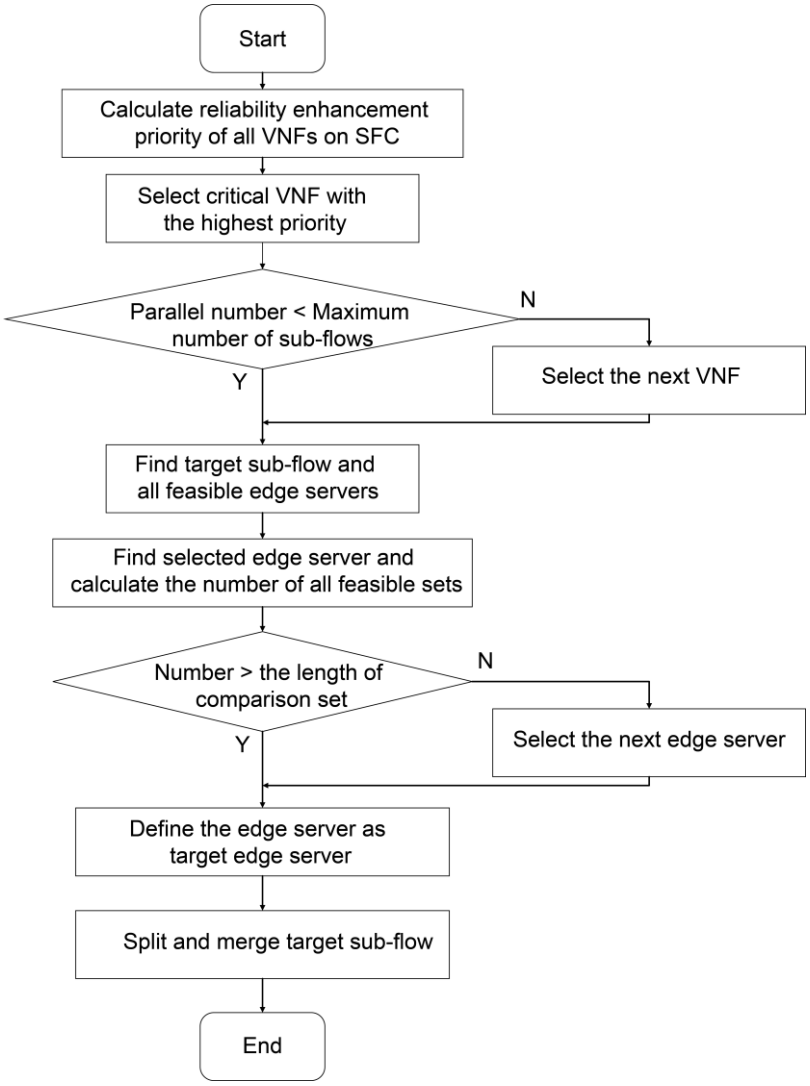

Figure 6. Explanation of the reliability-enhancement scheme in our algorithm.

If the delay exceeds the delay requirement, this algorithm reduces the delay by deploying parallel VNF instances in lines 16–23. It calculates the delay-reduction priority of all VNFs in SFC and executes the delay-reduction scheme to find critical VNFs and target servers and determine the split and merge of user requests. This procedure continues until the delay requirement is satisfied. However, if the delay requirement is not satisfied when the number of parallel instances for all VNF^l in SFC equals the maximum number of sub-flows or no edge server resources are available to deploy VNF instances, it terminates the loop, drops user request r_i, and releases the resource it preempts. Otherwise, it deploys parallel VNFs to satisfy the reliability requirement.

Algorithm 1 calculates the reliability-enhancement priority for each VNF in SFC and then executes the reliability-enhancement scheme. This procedure continues until the reliability requirement is satisfied. However, if the reliability requirement is not satisfied when the number of parallel VNF instances for all VNF^n in SFC equals the maximum number of sub-flows or no edge server resources are available to deploy VNF instances, the next step is to deploy BVNFs for user request r_i on lines 27–36.

Algorithm 1 Throughput Maximization Algorithm

Input: G, R;
Output: A, P;
$A = 0$;
FOR all V_i, $V_j \in V$ DO
    Calculate P$[V_i][V_j]$ by Dijkstra algorithm;
END FOR
FOR all $U_i \in U$, $BS_j \in BS$ DO
    Calculate D$[U_i][BS_j]$ and find the nearest base station for U_i;
END FOR
FOR all $r_i \in R$ DO
    FOR all $VNF^l \in SFC_i$ DO
        Find the edge server V_k with the lowest transmission distance;
        IF $C_k > C_{f_{VNF^l}}$ THEN
            Deploy one VNF^l instance in V_k;
            Calculate the delay and reliability of r_i;
        END IF
    END FOR
    WHILE the delay of $r_i > D_i$ DO
        Execute the delay-reduction scheme;
        IF all $\left|VNF^l\right| >= n_i$ THEN
            Drop r_i;
            Release all resources of r_i;
            BREAK;
        END IF
    END WHILE
    IF the delay of $r_i <= D_i$ DO
        WHILE the reliability of $r_i < R_i$ DO
            Execute the reliability-enhancement scheme;
            IF all $|VNF^l| >= n_i$ THEN
                Find VNF^l with the highest priority;
                Find server V_k with the lowest communication distance;
                IF $C_k > C_{f_{VNF^l}}$ THEN
                    Deploy one backup instance in V_k;
                    Calculate the delay and reliability of r_i;
                ELSE
                    BREAK;
                END IF
            END IF
        END WHILE
    END IF
    IF the delay of $r_i <= D_i$ && $r_i >= R_i$ DO
        $A = A + 1$;
    END IF
END FOR
Return A, P;

This algorithm also calculates the reliability-enhancement priority for each VNF in SFC and selects the VNF with the highest priority, and selects VNF^l with high priority as the critical VNF. Then, it finds the nearest edge server that deploys an active VNF^l instance and a sub-flow that traverses the edge server. Additionally, it records the previous and next edge servers that are traversed by the sub-flow and deploys VNF^{l-1} and VNF^{l+1} instances, respectively. Finally, it selects the edge server with the lowest communication distance from the previous server to the server and from the server to the next edge server. If there are no more VNF instances that can be deployed in any edge servers without violating their resource capacities, it terminates the loop, drops the user request r_i, and releases the

resource it preempts. Otherwise, this procedure continues until the reliability requirement is satisfied.

4.2. Complexity Analysis

In Algorithm 1, finding the shortest paths among edge servers and the nearest base station for all mobile users requires $O(|v|^3 + |U||V|)$. Then, deploying one VNF instance for each VNF in SFC for all user requests requires $O(|U||L_i||V|)$. Then, deploying parallel VNF instances to satisfy delay and reliability requirements for all user requests requires $O(|U||L_i||V| + |U||L_i||V||n_i|^2)$. Next, deploying BVNF instances to satisfy the reliability requirements for all user requests requires $O(|U||L_i||V|)$. Thus, the time complexity of the throughput maximization algorithm is $O(|v|^3 + |U||L_i||V||n_i|^2)$.

5. Performance Evaluation

In this section, we evaluate the performance of our proposed algorithms through experimental simulations. The experiments are implemented with the MATLAB platform and run on a personal computer with Intel Core CPU i7-9750H @ 2.60 GHz, 24 GB RAM.

We use the real-world EUA (https://github.com/swinedge/edu-dataset, accessed on 16 June 2023) dataset to simulate the MEC network environment. In our experiments, the locations of base stations and users are from the EUA dataset. There is one edge server deployed alongside each base station. The resource capacity of each edge server is randomly between 50 and 150 units. The data size, delay, reliability requirements, and maximum number of sub-flows of each user request r_i are randomly between 100 and 300 KB, 100 and 300 ms, 0.7 and 0.99 and 1 and 4, respectively. The number of VNF types is 20. The resource demand, processing capacity, and reliability of each VNF are randomly between 1 and 10 units, 10^4 and 10^5 KB/s and 0.6 and 0.9, respectively. We conducted experiments with the above settings. To further understand the effectiveness of our proposed algorithm, we compared it with two benchmark algorithms. We also define our proposed algorithm for the joint deployment of P-SFCs and BVNFs with dynamic reliability as DRPD:

1. The Joint Deployment algorithm of P-SFCs and BVNFs with Static Reliability (SRPD): this algorithm deploys parallel VNF instances only to decrease delay. Then, it deploys BVNF instances to satisfy the reliability requirement. The reliability of parallel VNF instances is the product of the reliability of all VNF instances;
2. The Joint Deployment algorithm of P-SFCs and BVNFs for Decreasing BVNFs (DBPD): this algorithm deploys parallel VNF instances only to decrease delay and enhance reliability for decreasing BVNFs. When deploying parallel instances to decrease delay, it calculates delay-reduction priority and finds the target edge server by comparing the communication distances from the previous edge server traversed by the target sub-flow to all feasible edge servers.

We also investigate the impact of important parameters on the performance of our proposed algorithm. The important parameter settings in our experiments are shown in Table 2.

Table 2. List of important parameters in this paper.

Notation	Explanation	Value
$\|U\|$	Number of mobile users	50, 100, 150, 200, 250
$\|V\|$	Number of edge servers	50, 75, 100, 125
$\|L_i\|$	Number of VNFs in SFC	6, 8, 10, 12, 14
$\|R\|$	Reliability of each VNF	0.6, 0.7, 0.8, 0.9

5.1. The Impact of the Number of Mobile Users on Different Algorithms

We first analyze the impact of the number of mobile users on network performance.

Figure 7a shows that the average delay increases with the number of mobile users. As the number of mobile users increases, the resource capacities of nearby edge servers are

not enough to deploy the critical VNF. The critical VNF needs to be deployed in further edge servers. Meanwhile, DRPD is superior to BMPD and SRPD in terms of the difference between the average delay and the average delay requirement. DRPD deploys parallel instances for critical VNFs with the highest delay between the previous server and the next server, while BMPD deploys it for critical VNFs with the highest delay between the previous server and the current server. However, SRPD does not deploy parallel instances for critical VNF.

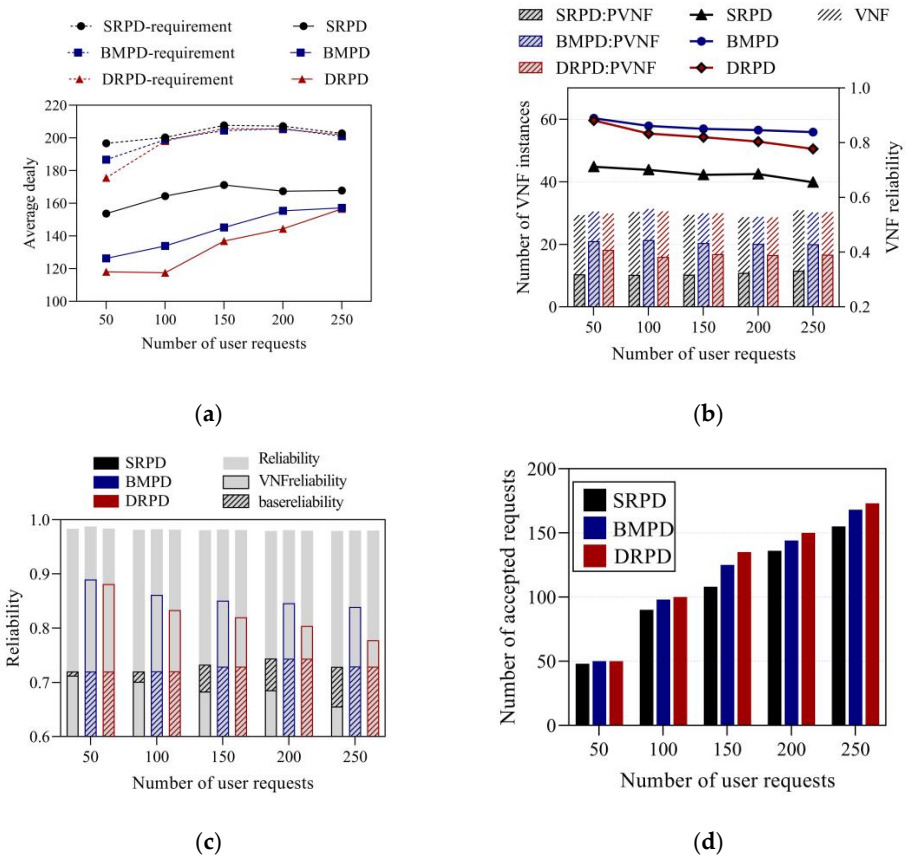

Figure 7. (**a**) Average delay and average delay requirement of all accepted user requests when the number of users changes; (**b**) the number of VNF instances consisting of active VNF instances and BVNF instances and average VNF reliability provided by all active VNF instances of all accepted user requests when the number of users changes; (**c**) the average reliability provided by one VNF instance, all active VNF instances and all VNF instances of all accepted user requests when the number of users changes; (**d**) throughput when the number of users changes.

Then, we define the average reliability provided by all parallel instances for each VNF as the average VNF reliability. When there is only one instance that is deployed for each VNF, we define the average reliability of each VNF as the base reliability.

In Figure 7b, BMPD is superior to our proposed algorithm in terms of the number of parallel VNF instances and average VNF reliability. DRPD selects VNF with the highest delay between the previous server and the current server as the critical VNF. When replacing edge servers in the comparison set with the target edge server, the number of all feasible sets that satisfy the delay requirement is lower than BMPD. The reason for this probably is that after replacement, the delay of the user request traversing the target edge server in DRPD is higher than the delay before replacement. Thus, BMPD can deploy more parallel instances than DRPD. The enhancement of VNF reliability in BMPD is also higher than in DRPD.

In Figure 7c, we can observe that the difference between VNF reliability and base reliability decreases with the increase in the number of mobile users. As the number of

mobile users increases, algorithms may deploy VNFs on further target edge servers. The increase in delay leads to a decrease in the number of all feasible sets about the target edge server, thus resulting in a decrease in reliability. For SRPD, the increase in delay leads to an increase in the number of parallel VNFs, thus decreasing reliability.

Figure 7b,c also show that there is no significant performance difference in terms of the number of instances for each VNF and the final reliability provided by all instances among the three algorithms. This is because, to satisfy the reliability requirement, the average reliability for each VNF in SFC provided by active and backup VNFs remains the same. The number of BVNFs needed to satisfy reliability generally remains the same in SRPD. Additionally, we select BVNFs for which the number of feasible sets exceeds the length of the comparison set. Thus, the reliability enhancement is close to the reliability improvement of deploying one BVNF.

Figure 7d shows that the throughput increases with the number of mobile users while the rate of increase gradually slows down. This is mainly because the limited resource capacities of edge servers can only deploy a limited number of VNFs and BVNFs. Meanwhile, our proposed algorithm is superior to the other two comparison algorithms. This can be explained by Figure 7b,c. Although there is no significant performance difference in terms of the number of instances in SFC, the average delay of DRPD is less than that of BMPD and SRPD.

5.2. The Impact of the Number of Edge Servers on Different Algorithms

In this section, we compare our proposed algorithm with two other algorithms. Figure 8a shows that the average delay decreases with the number of edge servers. When the number of edge servers increases, the critical VNF can be deployed in nearby edge servers.

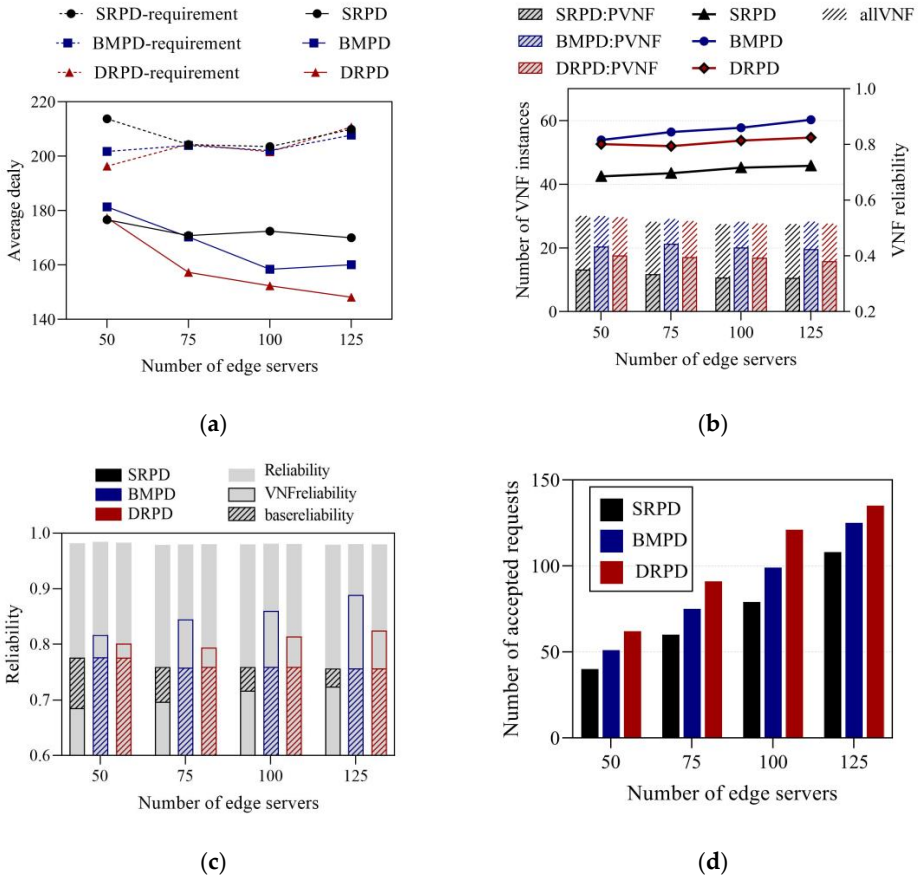

Figure 8. (**a**) Throughput when the number of edge servers changes; (**b**) the number of VNF instances

and VNF reliability of all accepted user requests when the number of edge servers changes; (**c**) the average reliability provided by one VNF instance, all active VNF instances and all VNF instances of all accepted user requests when the number of edge servers changes; (**d**) throughput when the number of edge servers changes.

Figure 8b,c show that the average VNF reliability rises as the number of edge servers increases. Since the number of edge servers increases, algorithms can select edge servers with shorter communication distances to deploy parallel VNFs. Thus, the number of feasible sets and the reliability enhancement increase. For SRPD, the decrease in delay leads to a decrease in the number of parallel VNFs, thus enhancing reliability.

Figure 8d shows that the throughput increases with the number of edge servers. When the number of edge servers increases, the resource capacity of the MEC network increases and can process more user requests.

5.3. The Impact of the Number of VNFs on Each SFC on Different Algorithms

In this section, we vary the number of VNFs on each SFC. Figure 9a shows that the number of parallel VNF instances increases with the VNF number. Since the transmission delay increases with the number of VNFs, the number of parallel VNFs needed to satisfy the delay requirement increases significantly. The reliability enhancement of all feasible sets decreases with the number of VNFs. Thus, the reliability provided by all parallel instances for each VNF in SFC decreases with the VNF number, as shown in Figure 8b. Moreover, to satisfy reliability requirements, the number of BVNFS is increasing.

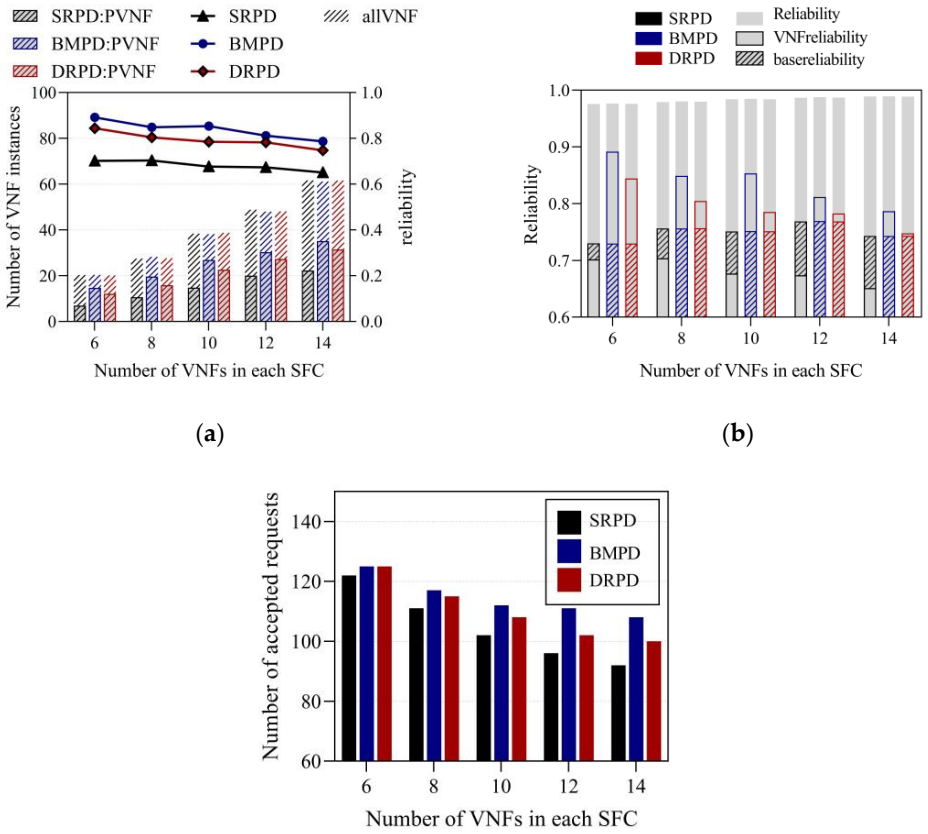

Figure 9. (**a**) Throughput when the number of VNFs in SFC changes; (**b**) the number of VNF instances and the VNF reliability of all accepted user requests when the number of VNFs in SFC changes; (**c**) the average reliability provided by one VNF instance, all active VNF instances and all VNF instances of all accepted user requests when the number of VNFs in SFC changes.

In Figure 9b, we can observe that the difference between VNF reliability and base reliability increases with the VNF number in SRPD. The increase in delay leads to an increase in the number of parallel VNFs, thus decreasing reliability.

Figure 9c shows that the throughput decreases with the VNF number. When the VNF number increases, the resource demand and transmission delay of user requests increase. However, the resource capacity in the network is fixed, so the throughput will decrease.

5.4. The Impact of the Reliability of Each VNF on Different Algorithms

In this part, we vary the reliability of VNF in each SFC. The number of all VNF instances decreases with the reliability of VNF, and the reliability provided by all parallel instances for each VNF in SFC increases with the reliability of VNF, as shown in Figure 10a,b. When the reliability of VNF increases, the reliability enhancement achieved by deploying the same number of parallel instances increases. Thus, fewer BVNFs are deployed to satisfy the reliability requirement.

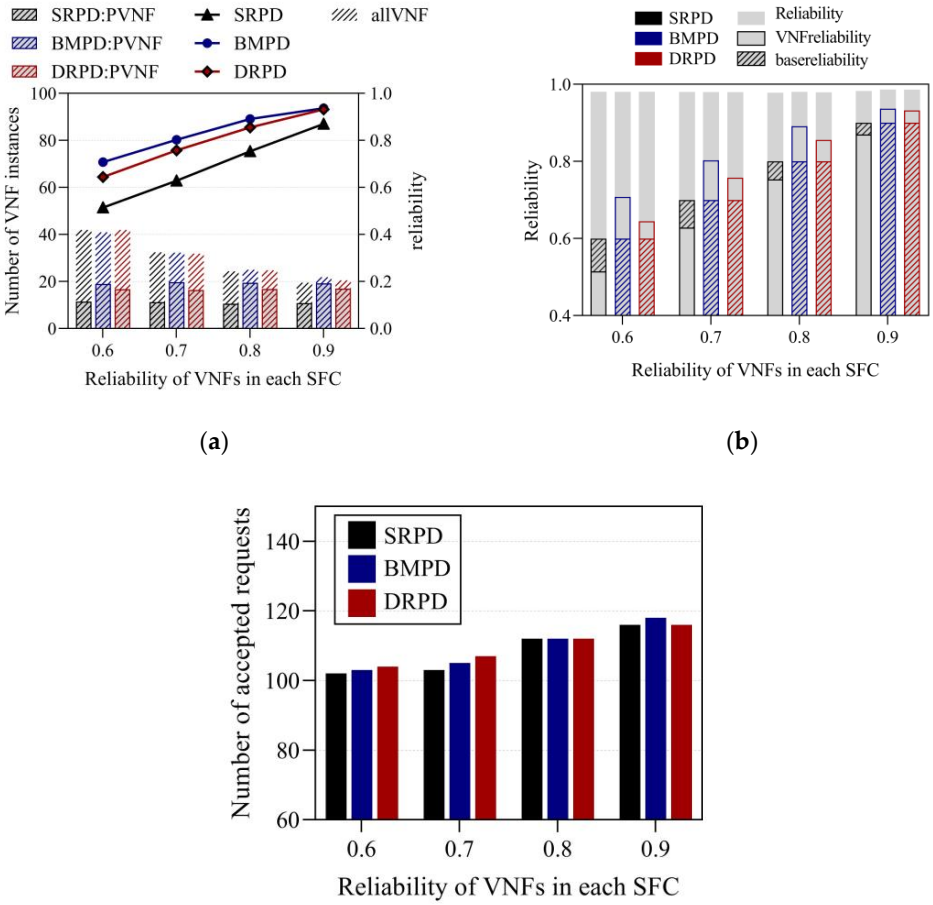

Figure 10. (**a**) Throughput when the reliability of VNF in SFC changes; (**b**) the number of VNF instances and VNF reliability of all accepted user requests when the reliability of VNF in SFC changes; (**c**) the average reliability provided by one VNF instance, all active VNF instances and all VNF instances of all accepted user requests when the reliability of VNF in SFC changes.

Figure 10c shows that the throughput increases with the reliability of VNF. The total number of VNF instances deployed to satisfy the reliability requirement decreases with the reliability of VNF. The fixed resource capacity in the network can accept more user requests.

6. Conclusions

The integration of multi-access edge computing (MEC) and parallelized service function chains (P-SFCs) can provide low-delay network services for mobile users by deploying virtualized network functions (VNFs) to process user requests in parallel. Multiple backup VNFs (BVNFs) are deployed near these VNFs to satisfy reliability requirements. In this paper, we studied the problem of maximizing the number of accepted user requests in the resource-limited MEC network. We first proposed a new approach to enhance the reliability of P-SFCs and reduce the number of idle BVNFs. Then, we modeled the dynamics of delay and reliability caused by VNF parallelization and BVNF deployment and formulated the joint deployment problem of P-SFCs and BVNFs. Next, we designed an approximation algorithm to find a near-optimal deployment of P-SFCs and BVNFs by finding the critical VNF, target sub-flow, and edge server and determining the split and merge of sub-flows. Experimental results based on real-world datasets show that our proposed algorithm outperforms two representative algorithms.

For future work, we would like to extend our research in two directions. The first direction is to study a multi-objective optimization problem by considering the trade-offs of reliability, delay, and resource preemption. Meanwhile, the second direction is parallel scheduling and congestion control in a dynamic MEC network.

Author Contributions: Conceptualization, Y.H. and J.L.; methodology, Y.H.; software, Y.L.; validation, Y.H.; formal analysis, Y.H. and J.L.; investigation, Y.L.; resources, Y.L.; data curation, Y.H.; writing original draft preparation, Y.H.; writing—review and editing, Y.H. and J.L.; visualization, Y.H. and Y.L.; supervision, Y.H. and J.L.; funding acquisition, Y.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K.B. A Survey on Mobile Edge Computing: The Communication Perspective. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2322–2358. [CrossRef]
2. Mach, P.; Becvar, Z. Mobile Edge Computing: A Survey on Architecture and Computation Offloading. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1628–1656. [CrossRef]
3. Nguyen, C.T.; Nguyen, D.N.; Hoang, D.T.; Phan, K.T.; Niyato, D.; Pham, H.A.; Dutkiewicz, E. Elastic Resource Allocation for Coded Distributed Computing Over Heterogeneous Wireless Edge Networks. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 2636–2649. [CrossRef]
4. Zhang, X.; Huang, Z.; Wu, C.; Li, Z.; Lau, F.C.M. Online Stochastic Buy-Sell Mechanism for VNF Chains in the NFV Market. *IEEE J. Sel. Areas Commun.* **2017**, *2*, 392–406. [CrossRef]
5. Mijumbi, R.; Serrat, J.; Gorricho, J.; Bouten, N.; De Turck, F.; Boutaba, R. Network Function Virtualization: State-of-the-Art and Research Challenges. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 236–262. [CrossRef]
6. Yang, S.; Li, F.; Trajanovski, S.; Yahyapour, R.; Fu, X. Recent Advances of Resource Allocation in Network Function Virtualization. *IEEE Trans. Parallel Distrib. Syst.* **2021**, *32*, 295–314. [CrossRef]
7. Huang, H.; Miao, W.; Min, G.; Tian, J.; Alamri, A. NFV and Blockchain Enabled 5G for Ultra-Reliable and Low-Latency Communications in Industry: Architecture and Performance Evaluation. *IEEE Trans. Ind. Inform.* **2021**, *17*, 5595–5604. [CrossRef]
8. Ma, W.; Sandoval, O.; Beltran, J.; Pan, D.; Pissinou, N. Traffic aware placement of interdependent NFV middleboxes. In Proceedings of the IEEE INFOCOM 2017, Atlanta, GA, USA, 1–4 May 2017; pp. 1–9.
9. Li, B.; Cheng, B.; Liu, X.; Wang, M.; Yue, Y.; Chen, J. Joint Resource Optimization and Delay-Aware Virtual Network Function Migration in Data Center Networks. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 2960–2974. [CrossRef]
10. Promwongsa, N.; Abu-Lebdeh, M.; Kianpishesh, S.; Belqasmi, F.; Glitho, R.H.; Elbiaze, H.; Crespi, N.; Alfandi, O. Ensuring Reliability and Low Cost When Using a Parallel VNF Processing Approach to Embed Delay-Constrained Slices. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 2226–2241. [CrossRef]

11. Polese, M.; Chiariotti, F.; Bonetto, E.; Rigotto, F.; Zanella, A.; Zorzi, M. A Survey on Recent Advances in Transport Layer Protocols. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3584–3608. [CrossRef]
12. Taleb, T.; Ksentini, A.; Sericola, B. On Service Resilience in Cloud-Native 5G Mobile Systems. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 483–496. [CrossRef]
13. Fan, J.; Jiang, M.; Qiao, C. Carrier-grade availability-aware mapping of Service Function Chains with on-site backups. In Proceedings of the 2017 IEEE/ACM 25th International Symposium on Quality of Service (IWQoS), Vilanova i la Geltrú, Spain, 14–16 June 2017; pp. 1–10.
14. Li, J.; Liang, W.; Xu, W.; Xu, Z.; Jia, X.; Zomaya, A.Y.; Guo, S. Budget-Aware User Satisfaction Maximization on Service Provisioning in Mobile Edge Computing. *IEEE Trans. Mob. Comput.* **2022**, 1–13. [CrossRef]
15. Huang, M.; Liang, W.; Shen, X.; Ma, Y.; Kan, H. Reliability-Aware Virtualized Network Function Services Provisioning in Mobile Edge Computing. *IEEE Trans. Mob. Comput.* **2020**, *19*, 2699–2713. [CrossRef]
16. Shang, X.; Huang, Y.; Liu, Z.; Yang, Y. Reducing the Service Function Chain Backup Cost over the Edge and Cloud by a Self-adapting Scheme. In Proceedings of the IEEE INFOCOM 2020, Toronto, ON, Canada, 6–9 July 2020; pp. 2096–2105.
17. Li, J.; Guo, S.; Liang, W.; Chen, Q.; Xu, Z.; Xu, W.; Zomaya, A.Y. Digital Twin-Assisted, SFC-Enabled Service Provisioning in Mobile Edge Computing. *IEEE Trans. Mob. Comput* **2022**, 1–16. [CrossRef]
18. Rui, L.; Chen, X.; Gao, Z.; Li, W.; Qiu, X.; Meng, L. Petri Net-Based Reliability Assessment and Migration Optimization Strategy of SFC. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 167–181. [CrossRef]
19. Qiu, Y.; Liang, J.; Leung, V.C.M.; Wu, X.; Deng, X. Online Reliability-Enhanced Virtual Network Services Provisioning in Fault-Prone Mobile Edge Cloud. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 7299–7313. [CrossRef]
20. Liang, W.; Ma, Y.; Xu, W.; Xu, Z.; Jia, X.; Zhou, W. Request Reliability Augmentation With Service Function Chain Requirements in Mobile Edge Computing. *IEEE Trans. Mob. Comput.* **2022**, *21*, 4541–4554. [CrossRef]
21. Yang, L.; Jia, J.; Lin, H.; Cao, J. Reliable Dynamic Service Chain Scheduling in 5G Networks. *IEEE Trans. Mob. Comput.* **2022**, 1. [CrossRef]
22. Karimzadeh-Farshbafan, M.; Shah-Mansouri, V.; Niyato, D. A Dynamic Reliability-Aware Service Placement for Network Function Virtualization (NFV). *IEEE J. Sel. Areas Commun.* **2020**, *38*, 318–333. [CrossRef]
23. Li, J.; Liang, W.; Huang, M.; Jia, X. Reliability-Aware Network Service Provisioning in Mobile Edge-Cloud Networks. *IEEE Trans. Parallel Distrib. Syst.* **2020**, *31*, 1545–1558. [CrossRef]
24. Jia, J.; Yang, L.; Cao, J. Reliability-aware Dynamic Service Chain Scheduling in 5G Networks based on Reinforcement Learning. In Proceedings of the IEEE INFOCOM 2021, Vancouver, BC, Canada, 10–13 May 2021; pp. 1–10.
25. Qu, L.; Assi, C.; Shaban, K.; Khabbaz, M.J. A Reliability-Aware Network Service Chain Provisioning With Delay Guarantees in NFV-Enabled Enterprise Datacenter Networks. *IEEE Trans. Netw. Serv. Manag.* **2017**, *14*, 554–568. [CrossRef]
26. Qu, L.; Assi, C.; Khabbaz, M.J.; Ye, Y. Reliability-Aware Service Function Chaining With Function Decomposition and Multipath Routing. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 835–848. [CrossRef]
27. Engelmann, A.; Jukan, A. A Reliability Study of Parallelized VNF Chaining. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6.
28. Engelmann, A.; Jukan, A.; Pries, R. On Coding for Reliable VNF Chaining in DCNs. In Proceedings of the 2019 15th International Conference on the Design of Reliable Communication Networks (DRCN), Coimbra, Portugal, 19–21 March 2019; pp. 83–90.
29. Kianpisheh, S.; Glitho, R.H. Cost-Efficient Server Provisioning for Deadline-Constrained VNFs Chains: A Parallel VNF Processing Approach. In Proceedings of the 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 11–14 January 2019; pp. 1–6.
30. Wang, M.; Cheng, B.; Wang, S.; Chen, J. Availability- and Traffic-Aware Placement of Parallelized SFC in Data Center Networks. *IEEE Trans. Netw. Serv. Manag.* **2021**, *18*, 182–194.
31. Alleg, A.; Ahmed, T.; Mosbah, M.; Boutaba, R. Joint Diversity and Redundancy for Resilient Service Chain Provisioning. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 1490–1504. [CrossRef]
32. Thiruvasagam, P.K.; Kotagi, V.J.; Murthy, S.R. A Reliability-Aware, Delay Guaranteed, and Resource Efficient Placement of Service Function Cshains in Softwarized 5G Networks. *IEEE Trans. Cloud Comput.* **2022**, *10*, 1515–1531. [CrossRef]
33. Tran, T.X.; Pompili, D. Joint Task Offloading and Resource Allocation for Multi-Server Mobile-Edge Computing Networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 856–868. [CrossRef]
34. Dinh, T.; Kim, Y. An efficient improvement potential-based virtual network function selection scheme for reliability/availability improvement. In Proceedings of the 2018 International Conference on Information Networking (ICOIN), Chiang Mai, Thailand, 10–12 January 2018; pp. 461–463.