




## Article

# Multi-Source Reliable Multicast Routing with QoS Constraints of NFV in Edge Computing

Shiming He <sup>1,2</sup>, Kun Xie <sup>3,4,\*</sup>, Xuhui Zhou <sup>3</sup>, Thabo Semong <sup>5</sup> and Jin Wang <sup>1</sup>

<sup>1</sup> School of Computer and Communication Engineering, Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation, Changsha University of Science and Technology, Changsha 410114, China; smhe\_cs@csust.edu.cn (S.H.); jinwang@csust.edu.cn (J.W.)

<sup>2</sup> Key Lab of Broadband Wireless Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications, Ministry of Education, Nanjing 210003, China

<sup>3</sup> College of Computer Science and Electronics Engineering, Hunan University, Changsha 410082, China; zhouxuhui@hnu.edu.cn

<sup>4</sup> State Key Laboratory of Networking & Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

<sup>5</sup> Department of Computer Science and Information Systems, Faculty of Science, Botswana International University of Science and Technology (BIUST), Palapye 10071, Botswana; semongt@biust.ac.bw

\* Correspondence: xiekun@hnu.edu.cn; Tel.: +86-731-8882-1976

Received: 10 August 2019; Accepted: 24 September 2019; Published: 1 October 2019



**Abstract:** Edge Computing (EC) allows processing to take place near the user, hence ensuring scalability and low latency. Network Function Virtualization (NFV) provides the significant convenience of network layout and reduces the service operation cost in EC and data center. Nowadays, the interests of the NFV layout focus on one-to-one communication, which is costly when applied to multicast or group services directly. Furthermore, many artificial intelligence applications and services of cloud and EC are generally communicated through groups and have special Quality of Service (QoS) and reliable requirements. Therefore, we are devoted to the problem of reliable Virtual Network Function (VNF) layout with various deployment costs in multi-source multicast. To guarantee QoS, we take into account the bandwidth, latency, and reliability constraints. Additionally, a heuristic algorithm, named Multi-Source Reliable Multicast Tree Construction (RMTC), is proposed. The algorithm aims to find a common link to place the Service Function Chain (SFC) in the multilevel overlay directed (MOD) network of the original network, so that the deployed SFC can be shared by all users, thereby improving the resource utilization. We then constructed a Steiner tree to find the reliable multicast tree. Two real topologies are used to evaluate the performance of the proposed algorithm. Simulation results indicate that, compared to other heuristic algorithms, our scheme effectively reduces the cost of reliable services and satisfies the QoS requirements.

**Keywords:** edge computing; network function virtualization; multicast tree; bandwidth; latency

## 1. Introduction

Cloud computing and data center allow scalable and flexible service development. However, some services that are delay sensitive do not work due to the high latencies of centralized solutions [1,2]. Edge Computing (EC) [3–6] is a new paradigm that pushes the computing from the cloud towards the edge and near the user. EC allows processing to take place near the user, ensuring scalability and low latency.

Network Function Virtualization (NFV) [7] provides a new optional network architecture of EC that uses standard IT virtualization technologies to integrate many network device types into industry

standard high-capacity servers. Traditionally, dedicated hardware runs lots of network functions (NFs), such as web proxy, load balancer, and firewall. Once one device fails or network functions need to be upgraded, one is forced to repurchase a new device, resulting in high capital investment and operating expenses. NFV technology consolidates NFs into general-purpose hardware platforms to reduce cost efficiently. The NF, for instance, firewall, IDS/IPS, router, Internet anomaly detection [8], or traffic analysis [9] are termed virtual network function (VNF). The general-purpose hardware platforms are termed virtual machine (VM) nodes. The performance of a service (e.g., bandwidth, CPU resource, and delay) depends on the VNF deployment on VM nodes, which is valuable for service providers and users [10]. Consequently, a great quantity of VNF deployment strategies [11–13] have been proposed in various NFV orchestration frameworks. The main focus of VNF placement is on one-to-one unicast communication [14–18].

Nowadays, there are many artificial intelligence (AI) applications on the cloud, such as computer vision, object tracking, natural language processing, and voice recognition [19–22]. Additionally, AI with edge computing applications in Internet of Things (IoT) has made a significant breakthrough, such as smart wearable devices in healthcare, smart automotive industry, recommender systems, and financial analysis. Most of these applications are generally communicated by a group and have special Quality of Service (QoS) and reliable requirements, which cause the change of NFV deployment in the edge network.

Once multiple users request the same service, repeated operations of VNF deployment and traffic transmission are created from the unicast, causing a waste of resources. This resource wastage can be reduced by using multicast technology. There are multiple destinations for multicast. It has an extensive and applicable future, and plays an important role in software updates and data monitoring [23–25]. The bandwidth consumption of the backbone network is reduced by more than 50% with multicast [26]. To ensure the security and performance of the propagation, multicast traffic often needs to pass through certain VNFs before arriving at all the destinations. By connecting these VNFs one after the other, a service function chain (SFC) is formed. We can view an SFC as a concrete implementation of a service request.

In terms of communication forms, multicast can be divided into three categories: one-to-many communication, many-to-one communication, and many-to-many communication. Currently, almost all the multicast studies focus on one-to-many communication mode [27–30]. However, in practice, there are still quite a lot of application scenarios belonging to many-to-many communication. Figure 1 is an example of the implementation of multi-user video conferencing. Each user may receive packets flowing from different EC servers. Therefore, in this paper, we aim to explore the resource allocation of many-to-many communication in NFV. Similar to some research on one-to-many communication, which focuses on finding a multicast routing tree in a network topology [27,29,30], the challenge for many-to-many communication is finding a routing forest with minimal resource consumption. The research on NFV layout with many-to-many communication is rare. Kuo et al. [31] proposed a  $3\rho_{ST}$ -approximation algorithm named SOFDA to allocate resources. However, the time complexity of this algorithm is extremely expensive and the end-to-end delay is not accounted for. In the conference version [32], we proposed a heuristic algorithm for the VNF layout problem in a multi-source multicast. However, all the deployment costs of VNF on VM nodes are the same, and we did not consider the reliability of the multicast tree.

On one hand, the deployment cost of VNF on VM node is different. For instance, some VNF has been deployed on VM node and the deployment cost is zero. On the other hand, many important AI applications require reliable transmissions, it has been predicted that reliable multicast [33,34] plays a crucial role when an SDN operator plans to provide multicast services. Some VM nodes can store the packets when forwarding the packets, denoted by recovery nodes. When a packet is lost, recovery nodes can retransmit the packet instead of the source node to reduce the transmission cost and improve the reliability of multicast tree. In this work, we focus on the reliable routing problem with multicast in NFV which finds a multicast tree and minimizes the cost of transmission and recovery, while

taking into account the end-to-end delay constraints. The contributions of our work are summarized as follows:

- We first introduce the reliable routing mode and formulate the problem of reliable NFV layout with various deployment costs, while taking account of the end-to-end latency. The aim is to find the scheme that consumes the least resources (minimal cost), including both the construction cost and recovery cost. Meanwhile, to minimize the cost, we assume that a VM can host more than one type of network function.
- We prove that the problem is an NP-hard (non-deterministic polynomial-time hard) problem.
- With the help of multilevel overlay directed (MOD) network and Steiner tree, a heuristic algorithm named Multi-Source Reliable Multicast Tree Construction (RMTC) is proposed. This algorithm handles the various deployment costs by MOD, chooses appropriate SFC placements shared by users, and guarantees the service's QoS and reliability.
- Two real topologies are used to evaluate the performance of the proposed algorithm. Simulation results indicate that, compared to other heuristic algorithms, our scheme effectively reduces the cost of reliable services and stratifies the QoS requirements.

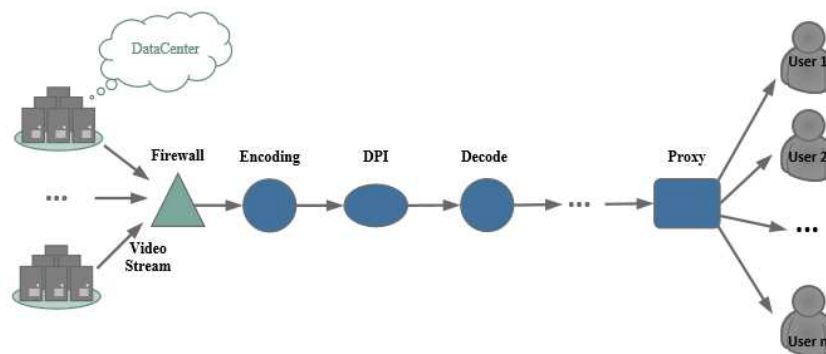


Figure 1. An instance of many-to-many communication.

The remainder of this paper is organized as follows: Section 2 briefly reviews the related works. Section 3 includes the system models, the reliable routing mode and problem formulation. Section 4 proposes a heuristic algorithm to find a reliable multicast tree. Section 5 uses simulations to analyze the performance. Section 6 concludes the work.

## 2. Related Work

This paper is related to routing, which is well studied in current work [35–40]. However, routing studies consider the issue of path finding; they do not consider the issues in network function embedding, which is the focus of this paper.

The main focus of VNF placement is on unicast communication. Cohen et al. [15] chose the placement location of the virtual functions, for it has a critical impact on the performance of the network, as well as on the reliability and operation cost, but this is not a requirement for VNF sequencing. The researcher tried to embed as many service chains as possible [16,17]. Li et al. [16] proposed an NFV-RT system that dynamically provides resources for timing guarantees. It aimed at maximizing the total number of requests that can be assigned to the cloud for each service chain, while ensuring that the assigned requests meet their deadlines. An online algorithm, named OSCEP [17], is proposed to solve the problem of maximum service chain embedding. Kuo et al. [18] studied the joint problem of VNF placement and path selection in order to better utilize the network for the relation between the link and server usage, which usually plays a crucial role in the problem. However, all of these works concentrate on unicast, which is costly for group services.

Multicast routing in NFV have started to attract more and more interest of late [27–30]. Xu et al. [27] aimed to minimize the cost for every multicast request or maximize the network throughput for a request sequence. They proposed a 2K approximation algorithm based on the assumption that all VNFs in a service chain are placed on the same server. Molnar et al. [28] considered the problem of QoS guarantee in the hierarchical structure of the network.

For service reliability, when a server or link fails, a backup server replaces the original server to complete the corresponding service on time. When the user does not receive the packet successfully because of network congestion, the data center needs to retransmit the packet. Shen et al. [29] proposed a new reliable multicast tree for SDN, named Recover-aware Steiner Tree (RST). The goal of RST is to minimize both the tree and recovery costs, which is proved to be NP-Hard and can be approximated within the number of destination nodes. RST only takes the link cost into account. Babarczia et al. [33] introduced a dynamic routing framework which provides optimal resilient capacity allocation against multiple link failures, both for routing and network coding. Ayoubi et al. [34] distinguished the failure of multicast source node from that of multicast terminal nodes and adopted a hop-to-hop search to handle receiver or terminal node failures, and a path convergence approach to recover from source node failures. The goal [30] was the reliability and approximation of the multicast tree.

For the deployment cost variety of VM nodes, Ren et al. [41] proposed a two stage service function tree embedding algorithm (TSE). It finds a multicast tree with minimum cost when some VNFs are ran on some VM nodes. It means the deployment cost of VNFs on VMs are different. However, the reliability of this tree solution is not guaranteed.

However, the QoS of multicast is not guaranteed in all of these works. For real-time communication, like video conference and IPTV, all these algorithms are ineffective. Our previous work in the conference version [32] proposed a heuristic algorithm for the VNF layout problem in multi-source multicast to guarantee the QoS. However, all the deployment costs on VM nodes are the same, and it did not consider the reliability of the multicast tree.

Therefore, inspired by [29,32,41], our work firstly considers the recovery cost and guarantees QoS requirement in various deployment cost environments which combine the situations among [29,32,41]. To store and handle the various deployment cost, we exploit a cost matrix and MOD networks similar to [41]. The recovery cost definition is from [29]. Based on the MOD networks, we minimize the recovery cost including link's cost and node's cost and check whether the path with the minimal cost satisfies the QoS requirement. According to the algorithm framework of our previous work [32], we design several sub-algorithms to solve it.

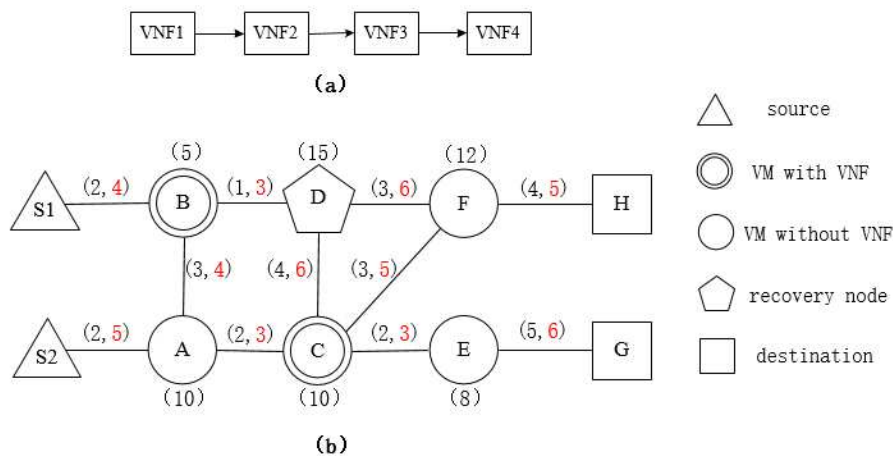
### 3. Problem Statement

In this section, we consider the Multi-Source Reliable Multicast Routing with QoS Constraints (MRMRQC) problem when deployment costs of VFN on VM are different. For sequences of service requirements, the goal of MRMRQC is to minimize the total cost, which consists of node cost, link cost, and recovery cost, while satisfying the end-to-end delay requirement. The node cost is the computing resources consumption on VM. The link cost represents the link's bandwidth consumption along the path passed by the flow. The recovery cost is the retransmission resource consumption after losing a packet. Note that a VM node can run multiple VNFs as long as it has enough resources. The computational resource of VNF varies from one VM to the other. For example, if a VM has run a VNF, other chains can use the same VNF in this VM directly, without having to run the same VNF again. Moreover, we take the cost of each VM into account only one time even if the flow passes through that VM several times. On the contrary, the cost of a link should be considered according to how many times a flow passed through it, while the link's cost in a flow is equal to the cost of link multiplied by the corresponding times.

### 3.1. System Model

The considered network is modeled as an edge-weighted vertex-weighted directed graph  $G = (V, E)$ ,  $V = V_S \cup V_M \cup V_D$ , where  $V_S$  is the source set.  $V_M$  represents the set of VM nodes that are real physical nodes and can run/host different kinds of VNFs. Each VM node has a limited computational capability  $C_v$  and can be hosted on a server.  $V_M = V_R \cup V_U$ , where  $V_R$  is the set of recovery nodes and  $V_U$  is the set of switch nodes. Recovery nodes are a kind of VM nodes that have storing function except for VNF implementation. Recovery node can store packets. It can re-transmit the packet when packets are lost. Re-transmitting from recovery node instead of source node reduces the transmission cost because recovery nodes are always closer to the destination than the source node. A switch without computational capability can only forward the traffic.  $V_D$  is the set of destination nodes requesting the same services, and, as shown in Figure 2, there are 10 nodes.  $V_S = \{S1, S2\}$ ,  $V_M = \{A, B, C, D, E, F\}$ ,  $V_R = \{D\}$ ,  $V_U = \{A, B, C, E, F\}$ ,  $V_D = \{G, H\}$ . Nodes B, E have run the VNF2 and VNF3, respectively, and node D is a recovery node.

$E$  is the set of links, and each link is associated with two non-negative weights  $b_{u,v}$  and  $d_{u,v}$ , denoting the bandwidth consumption and delay, respectively. The set of neighbor nodes of  $v$  in  $G$  is denoted by  $N_v$ . If there is a link from  $u$  to  $v$ , then  $u$  is a neighbor of  $v$  and  $u$  belongs to  $N_v$ . In Figure 2, the first black digit near the edge represents bandwidth consumption  $b_{u,v}$  and the second red digit represents delay  $d_{u,v}$ . The digit near the node  $v$  is the available computation resource of node  $C_v$ .



**Figure 2.** An example of a reliable multi-source network.

There is a set of VNFs  $F$ , and an SFC  $\mathcal{L} = (f_1, f_2, \dots, f_{|\mathcal{L}|})$  that is composed of some VNFs from  $F$ . Each VNF  $f$  running on VM  $m$  requires a certain computational resource  $c_{f,m}$ . All the computational resource requirements can be represented by a matrix, as shown in Figure 3. The row represents the VNF, and the column represents the VM. The element in the  $f$ th row and the  $m$ th column is the computational resource of VNF  $f$  running on VM  $m$ . The elements in the 2nd row and 2nd column, and the 3th row and 2nd column are both zeros, as a result of nodes B and E having run the VNF2 and VNF3. Therefore, the computational resource is zero and each column from the matrix represents the available computation resource of the nodes. We assume that each VM  $m$  supports all the VNFs in  $F$  and can run more than one VNF. Additionally,  $\delta$  is the upper bound threshold of delay that the user can wait for.

$$DC = \begin{matrix} & \begin{matrix} f_1 & f_2 & f_3 & f_4 \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} & \begin{pmatrix} 2 & 3 & 5 & 4 \\ 3 & 0 & 3 & 2 \\ 3 & 2 & 0 & 5 \\ 5 & 4 & 3 & 3 \\ 7 & 3 & 2 & 2 \\ 4 & 5 & 3 & 1 \end{pmatrix} \end{matrix} \quad \begin{matrix} 10 \\ 5 \\ 10 \\ 15 \\ 8 \\ 12 \end{matrix}$$

Figure 3. Deployment cost matrix.

### 3.2. Reliable Transmission Model

The node which stores the packets becomes a recovery node when it forwards the packets. Source and recovery nodes are denoted as recovery agents. Transmission cost is the sum of node cost and link cost. The recovery cost is the retransmission cost from the recovery agent to the destination nodes when the packet is lost. The reliable routing problem [34] in NFV is to construct a path that the flow traverses through all the selected VMs including recovery nodes to minimize the transmission cost and recovery cost. Figure 4 is a multicast tree of the network and acts as an example of a reliable transmission. Services VNF1, VNF2 and VNF3 in the service chain are deployed in nodes A, B, and C, respectively. The cost of all the links is 1, and node B is the recovery node. Assuming that we ignore the node cost, the transmission cost then becomes  $1 + 1 + 1 + 1 + 1 = 5$ . The recovery cost is determined by the packet loss which includes the following five cases:

- Nodes D and E receive the packet successfully and there is no packet loss, hence the recovery cost is zero.
- Nodes D or E do not receive the packet, and node B receives the packet successfully. There is packet loss in the links after node B, and, therefore, node B can retransmit the packet to nodes D or E, hence the recovery cost is 2.
- Nodes D and E do not receive the packet, and node B receives the packet successfully. There is packet loss in the links after node B, and, therefore, node B can retransmit the packet to nodes D and E, hence the recovery cost is 3.
- Nodes D or E do not receive the packet, and node B also does not receive the packet. There is packet loss in the links from source to node B, and, therefore, only the source node can retransmit the packet to nodes D or E, hence the recovery cost is 4.
- Nodes D and E do not receive the packet, and node B also does not receive the packet. There is packet loss in the links from the source node to node B, and, therefore, only the source node can retransmit the packet to nodes D and E, hence the recovery cost is 5.

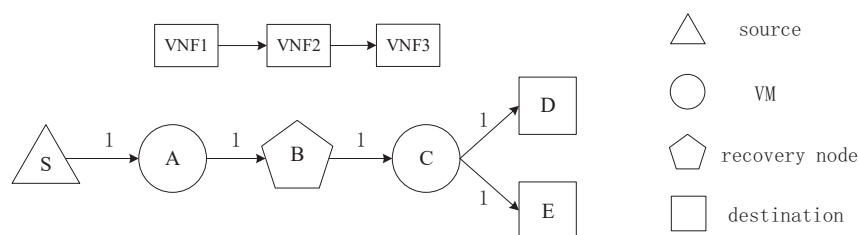


Figure 4. Reliable transmission example.

The recovery cost is determined by the packet loss that may occur at any link. It is necessary to take all the cases into account. Two parameters  $\rho_{u,v}$  and  $\kappa_d$  denote the loss probability of link  $e(u, v)$



and the recovery cost of destination  $d$ , respectively. When only the source node can retransmit the packet to the destination, and the path  $e_1, e_2, \dots, e_n$  from the source to the destination consist of  $n$  links, the retransmission probability from the source to the destination can be calculated by Equation (1):

$$\rho_{s,d} = 1 - \prod_{i=1}^n (1 - \rho_{e_i}). \quad (1)$$

Once the packet is lost at the link along the path, then retransmission is needed. The recovery cost or retransmission cost of the destination is given by Equation (2):

$$\kappa_d = \rho_{s,d} \cdot \sum_{i=1}^n c(e_i), \quad (2)$$

where  $c(e)$  is the link cost or link resource consumption when the packet passes through link  $e$ .

When there are some recovery nodes, then the recovery cost becomes a bit complicated. We assume that the path from the source to the destination passes the recovery node  $r$ . As shown in Figure 4, the recovery node B is on the path from S to D and E. When the destination node does not receive the packet successfully, then one of the two recovery agents needs to retransmit that packet, and the recovery cost is given by Equation (3):

$$\kappa_d = \rho_{s,r} \cdot \sum_{i=1}^n c(e_i) + (1 - \rho_{s,r}) \cdot \rho_{r,d} \cdot \sum_{i=k}^n c(e_i). \quad (3)$$

In Equation (3), the first part of the equation represents retransmission cost from  $s$ . Retransmission from  $s$  is only necessary as a result of the packet loss occurring in the links before  $r$ . The cost is equal to the loss probability from  $s$  to  $r$  multiplied by the whole link cost from  $s$  to  $d$ . The second part represents the retransmission cost from  $r$ , meaning that the packet loss occurs in the links from  $r$  to  $d$ . The cost equals the loss probability from  $r$  to  $d$  multiplied by the whole link cost from  $r$  to  $d$ , and the path  $e_k, e_{k+1}, \dots, e_n$  is the path from  $r$  to  $d$ .

### 3.3. Mixed Integer Linear Programming

We formulate the MRMRQC problem as a Mixed Integer Linear Programming (MILP). The corresponding VMs for deploying all the VNFs in SFC must be selected. The goal of MRMRQC is to construct a multicast topology  $MG$  to let the traffic arrive at all the destination nodes after passing through all the selected VMs at a minimum cost of construction and recovery. A binary variable  $\eta_{f,m}$  represents whether VNF  $f$  is run on the VM  $m$ , and a variable  $\varphi_{u,v}$  represents the number of times a link  $e(u,v)$  is passed by the flow. If a flow doesn't pass through link  $e(u,v)$ ,  $\varphi_{u,v}$  is 0 and  $\tau_v$  is a binary variable. The value is 1 when node  $v$  is a recovery agent, otherwise, it is 0, and  $T$  is the resultant multicast tree. Table 1 summarizes the notations and definitions used in this paper.

The goal is to construct a multicast tree while minimizing the transmission cost and recovery cost of each destination. The objective function can be described as Equation (4):

$$\min \sum_{f \in L} \sum_{m \in V_M} c_{f,m} \cdot \eta_{f,m} + \sum_{e(u,v) \in E} b_{u,v} \cdot \varphi_{u,v} + \alpha \cdot \sum_{d \in V_D} \kappa_d. \quad (4)$$

The first term represents the cost of VM running, and the computation cost of VNF varies from one VM to the other. The second term represents the link cost, and when a link is used several times, the link cost is also considered several times. The third term represents the recovery cost of all the destination nodes.  $\alpha$  denotes the weight of the recovery and, the higher the  $\alpha$ , the more reliable the multicast tree found becomes. If  $\alpha$  is 0, then the recovery cost is not considered. The problem formulation consists of the following four constraints:

(1) Resource constraints: The available resource of the VM node should be enough to run the allocated VNF:

$$C_v - \sum_{f \in L} c_{f,v} \cdot \eta_{f,v} \geq 0, \quad \forall v \in V_M. \quad (5)$$

(2) Routing constraints: Assuming that flow conservation holds for each node, to select a path, we must have the flow starting from at least one source node, then pass through several VMs before arriving at all the destination nodes:

$$\sum_{s \in V_S} \sum_{v \in N_s} \lambda_{d,s,v} - \sum_{s \in V_S} \sum_{v \in N_s} \lambda_{d,v,s} \geq 1, \quad \forall d \in V_D, \quad (6)$$

$$\sum_{v \in N_d} \lambda_{d,v,d} - \sum_{v \in N_d} \lambda_{d,d,v} = 1, \quad \forall d \in V_D, \quad (7)$$

$$\sum_{v \in N_u} \lambda_{d,u,v} = \sum_{v \in N_u} \lambda_{d,v,u}, \quad \forall d \in V_D, \forall u \in V_M \cup V_U, \quad (8)$$

$$\lambda_{d,u,v} \leq \varphi_{u,v}, \quad \forall d \in V_D, \forall e(u,v) \in E, \quad (9)$$

where  $\lambda_{d,u,v}$  is a binary variable, and the value is 1 when link  $e(u,v)$  is passed through by the flow from  $s$  to  $d$ . Constraint (5) ensures that at least one flow is sent from  $V_S$ . Constraint (6) ensures that each destination just receives the flow once. For non-source or destination nodes, constraint (7) ensures that the number of selected incoming links and outgoing links of node  $s$  are equal. When  $u$  is selected by MG, at least one incoming link and at least one outgoing link for  $u$  are selected; otherwise, both  $\lambda_{d,v,u}$  and  $\lambda_{d,u,v}$  are zero. Constraint (8) ensures that the number of times of  $\varphi_{u,v}$  that  $e(u,v)$  was traversed by the flow must be higher than 1 when  $e(u,v)$  is selected by MG.

**Table 1.** List of notations.

Notation	Definition
$V$	The node set
$E$	The link set
$C_v$	Available computation capacity of node $v$
$c_{f,m}$	Cost or required resource to deploy VNF (Virtual Network Function) $f$ on VM $m$
$d_{u,v}$	Delay of link $e(u,v)$
$b_{u,v}$	Bandwidth consumption of link $e(u,v)$
$F$	VNF set
$\mathcal{L}$	A service chain consisting of $(f_1, f_2, \dots, f_{ \mathcal{L} })$
MG	The constructed multicast routing topology
$N_v$	The neighbor node set of node $v$
$\eta_{f,m}$	A binary variable which is 1 when VM $m$ runs VNF $f$
$\varphi_{u,v}$	The number of times that $e(u,v)$ was traversed by the flow
$\lambda_{d,u,v}$	A binary variable whether the flow from $s$ to $d$ pass through $e(u,v)$
$\delta$	The upper bound of delay
$\bar{\delta}$	The upper bound of delay jitter
$\rho_{u,v}$	The loss probability of link $e(u,v)$
$\tau_v$	Recovery agent flag of node $v$
$\kappa_d$	The recovery cost of destination $d$
$T$	The found multicast tree

(3) QoS constraints: For real-time services, the QoS, such as end-to-end delay and the delay jitter among all the destinations should be satisfied:

$$\sum_{s \in V_S} \sum_{e(u,v) \in E} \lambda_{d,u,v} \cdot d_{u,v} \leq \delta, \quad \forall d \in V_D, \quad (10)$$



$$\left| \sum_{s \in V_S} \sum_{e(u,v) \in E} \lambda_{d_1,u,v} \cdot d_{u,v} - \sum_{s \in V_S} \sum_{e(u,v) \in E} \lambda_{d_2,u,v} \cdot d_{u,v} \right| \leq \bar{\delta}, \quad \forall d_1, d_2 \in V_D. \quad (11)$$

The total delay from  $s$  to each destination node should be lower than a threshold  $\delta$ , and the delay jitter between each two destination nodes should also be lower than a threshold  $\bar{\delta}$ .

(4) *Recovery constraints*: Only the recovery agent can retransmit the packet, and, therefore, the  $\tau_v$ 's values of the nodes which are not the sources or do not belong to  $V_R$  are zero:

$$\tau_v = 0, \quad \forall v \notin \{V_S \cup V_R\}. \quad (12)$$

### 3.4. The Challenge

The problem of MRMRQC is very challenging because it is the generalization of Multi-Source Multicast Routing with QoS Constraints (MMRQC) [32] when all the deployment costs of VNF on VMs is the same and there is only one recovery node. According to [32], MMRQC can be reduced to a Steiner Tree (ST) problem that is NP-hard. Therefore, we prove that the MRMRQC is also an NP-hard problem.

**Proof.** Multi-Source Multicast Routing with QoS Constraints (MMRQC) [32] is a special case of MRMRQC. MMRQC considers the implementation of multicast service request with minimal resource consumption with multiple sources. Retransmission cost is not considered in MMRQC and the costs of VNF deployed in different VM nodes are the same in the network of MMRQC. According to [32], we have proved that MMRQC is NP-hard because it can be reduced to a ST problem in polynomial time.

When the weight  $\alpha$  in the object function of MRMRQC is set to zero, reliability cost is not taken into account. When there is no reliability cost and all the deployment costs of VNF on VMs are the same, the MRMRQC is simplified into MMRQC. Since there are multiple sources, we can add a virtual node  $S$ , which is connected to all the nodes in  $V_S$  with link weights set to 0. Moreover, we assume that there is a VM node in the routing path from  $S$  to  $V_D$  that can run all VNFs of SFC. When QoS is ignored, the MRMRQC equals an ST problem. The two assumptions can be completed within polynomial time, and the check for QoS constraints can also be accomplished in polynomial time. When the weight  $\alpha$  is a non-zero or the deployment costs of VNF on VMs are varying, then the problem of MRMRQC is more complex than MMRQC. Therefore, the MRMRQC problem is NP-hard.  $\square$

## 4. Reliable Multicast Tree Construction

In order to solve the MRMRQC problem, an approximation algorithm, named Multi-Source Reliable Multicast Tree Construction (RMTC) is proposed in this section.

### 4.1. Overview

There are several sub-problems that are solved by sub-algorithms. Generally, RMTC consists of five phases: (1) Multilevel Overlay Directed Network Transformation Phase, (2) SFC Embedding Phase, (3) Multicast Tree Construction Phase, (4) Reliability Guarantee Phase, and (5) QoS Guarantee Phase. The first phase aims to transform the network into multilevel overlay directed (MOD) network for handling the various computational resources of VNF in different VM. In the second phase, the source nodes and a virtual source node are added into the MOD network. The shortest paths starting from a source node in  $V_S$  to every VM node in the last model of MOD network are found. The VNFs of SFC need to be deployed in these paths. In the third phase, a Steiner Tree is constructed from each node  $u$  in the last model of MOD network and all destination nodes. Then, we will get several multicast routing trees by connecting the shortest path and the corresponding Steiner Tree. In the fourth phase,

we calculate the node cost, link cost, and recovery cost of each multicast routing tree and select the tree with minimum cost. In the last phase, we calculate the end-to-end delay and maximum delay jitter between different destinations. We take the service request as acceptable when the end-to-end delay and maximum delay jitter are lower than the requirements.

#### 4.2. Multilevel Overlay Directed Network

Due to the varying deployment costs of VNF in different VM, we introduce the multilevel overlay directed (MOD) network to solve the problem. Multilevel overlay directed network contains all information of networks except for destination nodes.

- Step 1: Replicate all nodes of the network  $|L|$  times, where  $|L|$  is the length of the SFC. Arrange these nodes in a matrix form, in which columns denote VNF functions and rows denote nodes.
- Step 2: For each column, replicate all the nodes again and treat the column and its replicate as a model, meaning that there are  $|L|$  models. Connect each node to its replicate with dotted arrows from left to right in a model, and connect all the nodes to all their right neighbors with solid arrows between models.
- Step 3: Set the link weights in a model to equal the corresponding VNF computation resource costs on VM, and set the edge weights between modes to equal the corresponding shortest path costs between the two nodes in  $G$ .

Figure 5 is the MOD network transformed from Figure 2. In Figure 5, the values on the dotted arrow equal the matrix in Figure 3. The values on horizontal solid arrows are zero, such as  $A1 \rightarrow A$  and  $B1 \rightarrow B$ , and this means that the two VNF are running on the same VM, and there is no bandwidth consumption. The value on the non-horizontal solid arrow equals the shortest path bandwidth cost between two nodes in  $G$ , such as  $A1 \rightarrow B$ .

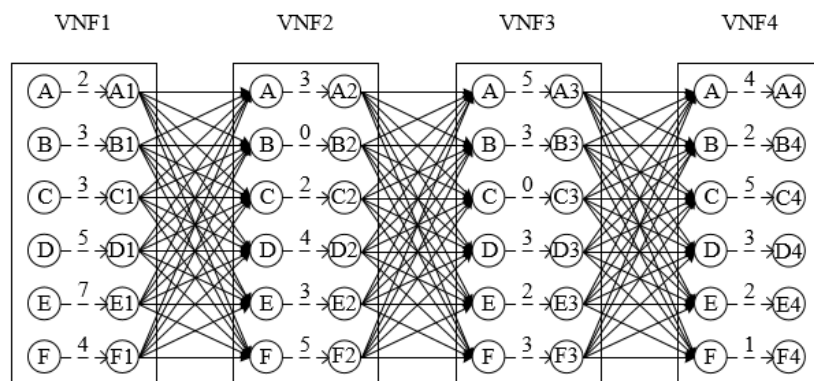


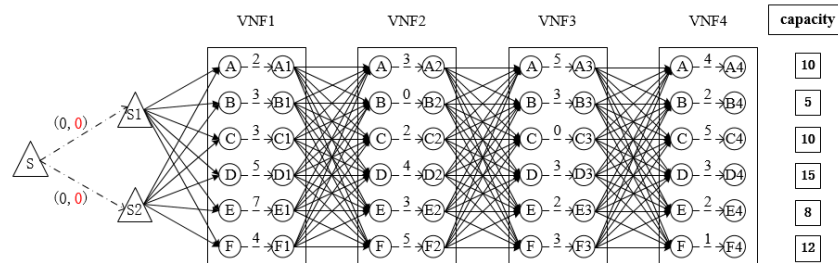
Figure 5. Multilevel overlay directed network example.

#### 4.3. SFC Embedding

After transforming the network into the MOD network, we add the sources and a virtual sources into the MOD network. As shown in Figure 6, we connect the sources with all VM nodes in  $l_1$ , and set the links weight to equal the corresponding shortest path costs from the sources to the VM node in  $G$ . The virtual source node is added to handle the multiple sources. It is not a physical node and is introduced only for the purpose of calculating the deployment result. The virtual source node is connected to all the source nodes with link weights set to 0. The right column represents the available computation resource of the VM nodes.

Then, we search for the shortest path from virtual source to the last model and satisfy the resource constraints. As shown in Figure 6, the shortest path from the virtual source to the last model found by

the improved Dijkstra algorithm [42–45] is  $S \rightarrow S1 \rightarrow B - B1 \rightarrow B - B2 \rightarrow D - D3 \rightarrow D - D4$ , which is  $S \rightarrow B \rightarrow D$  in the network  $G$ . Node  $B$  deploys VNF1 and VNF2, and node  $D$  deploys VNF3 and VNF4. VNF2 is running on node  $B$ ; therefore, only VNF1 is deployed in node  $B$ .

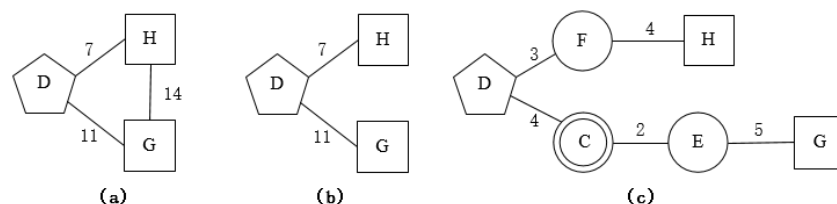


**Figure 6.** SFC (service function chain) embedding in the MOD (multilevel overlay directed) network.

#### 4.4. Multicast Tree Construction

All the VNFs run on the VM nodes on the path found in the first phase. The next step is to construct a tree to connect the last VNF's location and all destination nodes with minimum cost. The last VNF's location is always on the last model in the MOD network based on the first phase, and it is a Steiner tree problem, which is NP-hard. According to [32], we used an approximation algorithm to solve it, as shown in Algorithm 2. For example, we assume that node  $D$  is the last VNF's location, and, therefore, the Steiner tree should contain the nodes  $D, H$ , and  $G$ , as shown in Figure 7. Firstly, the nodes  $D, H$ , and  $G$  are selected to construct a complete graph, as shown in Figure 7a. The link's weight is the shortest path length between nodes in the original network  $G$ . Secondly, the minimal spanning tree of the complete graph network can be found by Prim algorithm. Finally, every link in the minimal spanning tree is replaced by the shortest path in the original graph  $G$ , and a minimal cost tree is constructed as shown in Figure 7c. Combining the shortest path from the first phase and the tree from the second phase, we can construct a multicast tree from the virtual source node to all the destination nodes. Figure 8 is the multicast tree of Figure 7.

The Steiner tree construction is determined by the location of the last VNF. Therefore, we need to find the shortest path to all the nodes, that is, it is necessary to find the shortest paths from virtual source to each VM node in the last model. Using Figure 6 as an example, the shortest paths from  $S'$  to  $A4, B4, \dots, F4$  are found and the Steiner trees from  $A, B, \dots, F$  to all destinations are constructed, respectively. In addition, we combine the shortest path and the corresponding Steiner tree into a multicast tree, hence we can get six multicast trees. We can get the constructing cost  $C(T)$  for each multicast tree by summing the nodes' cost and links' cost in the multicast tree.



**Figure 7.** An example of Steiner tree. (a) the complete graph; (b) the minimal spanning tree (MST) of (a); and (c) a Steiner tree by replacing the link with the shortest path in  $G$ .

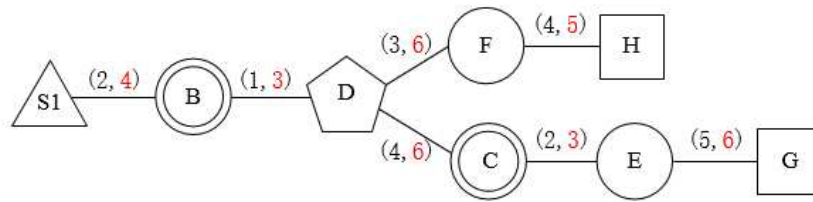


Figure 8. Constructed multicast tree.

#### 4.5. Reliability Guarantee

After getting multicast trees, we select the tree with minimum constructing cost and recovery cost. For each multicast tree, we calculate the tree's recovery cost  $R(T)$  by summing the recovery cost of each destination according to Equation (3). Getting the multicast tree with minimum constructing cost and recovery cost is our routing objective. The pseudo code of reliable multicast routing tree algorithm is shown in Algorithm 3.

---

#### Algorithm 1: MOD network transformation

---

**Input:**  $G = (V, E)$ ,  $V_S \subseteq V$ , a service chain  $\mathcal{L} = (f_1, f_2, \dots, f_L)$ , deployment cost matrix  $DC$ .  
**Output:** A MOD network  $G'$ .

- 1 Arrange all  $V_M$  nodes as column  $l_1$ , and set a matrix  $M = l_1$ ;
- 2 **for**  $i \leftarrow 2$  to  $|L|$  **do**
- 3     Replicate the column  $l_1$  as  $l_i$ ;
- 4     Splice matrix and  $l_i$  by  $M = M \oplus l_i$ ;
- 5 **end**
- 6 **for**  $i \leftarrow 1$  to  $|L|$  **do**
- 7     Replicate all nodes in column  $l_i$  again as column  $l'_i$ , and put  $l'_i$  on the right of  $l_i$ ;
- 8     Connect each node in  $l_i$  to its replicated node  $l'_i$  with dotted arrows from left to right in  $E'$ ;
- 9     Set the link weight equal to the corresponding VNF computation resource costs on VM;
- 10 **end**
- 11 **for**  $i \leftarrow 1$  to  $|L| - 1$  **do**
- 12     Connect each node in  $l_i$  to all their right neighbors  $l_{i+1}$  with solid arrows from left to right in  $E'$ ;
- 13     Set the link weight equal to corresponding shortest path costs between two nodes in  $G$ ;
- 14 **end**
- 15  $G' = (M, E')$ ;
- 16 **return**  $G'$

---



---

#### Algorithm 2: Steiner Tree Construction

---

**Input:**  $G = (V, E)$ ,  $V_D \subseteq V$ , the last VNF's location node  $u$ .  
**Output:** A Steiner tree  $ST$  with the least bandwidth consumption.

- 1 Calculate the shortest path between every two nodes in  $G$  by Floyd;
- 2 Pick out node  $u$  and  $V_D$  in  $G$  to construct a complete graph  $CG'$ , and the link's weight is the shortest path length between nodes in  $G$ ;
- 3 Find a minimal spanning tree  $ST'$  by Prim algorithm;
- 4 Get  $ST$  by replacing every link in  $ST'$  with the shortest path in  $G$ ;
- 5 **return**  $ST$

---

**Algorithm 3:** Reliable Multicast Tree Construction**Input:**  $G = (V, E)$ ,  $V_S \subseteq V$ , a MOD network  $G'$ .**Output:** A Reliable multicast tree  $T_{OPT}$  with minimal cost.

- 1 Calculate the shortest path length  $d_{s,v}$  from each source  $s \in V_S$  to each VM node  $v$ ;
- 2 Add all sources in the left of  $G'$ , connect each source to each node in the first model with solid arrow from left to right, and set the link weight as  $d_{s,v}$ ;
- 3 Add a virtual node  $S$  in  $G'$ , and connect virtual source to all sources with zero link weight;  
 $b_{S,V_S} \leftarrow 0$  and  $d_{S,V_S} \leftarrow 0$ ;
- 4 **for**  $u \in V_M$  **do**
- 5     Find the shortest path  $SP$  from virtual source  $S$  to VM node  $u$  in the last model;
- 6     Construct a Steiner tree  $ST$  from  $u$  to destination  $V_D$  by Algorithm 2;
- 7     Combine the shortest path  $SP$  and Steiner tree  $ST$  as a multicast tree  $T$ ;
- 8     Calculate the construction cost  $C(T)$  and recovery cost  $R(T)$ ;
- 9 **end**
- 10 Select the multicast tree with the minimum sum of  $C(T)$  and  $R(T)$  as  $T_{OPT}$ ;
- 11 **return**  $T_{OPT}$

**4.6. QoS Guarantee**

After the previous four stages, the multicast tree is found. The last step is to check whether the end-to-end delay and delay jitter of the multicast tree satisfies the requirements. The end-to-end delay from  $S_1$  to  $G$  and  $H$  is 18 and 22, respectively, as shown in Figure 8. The maximum end-to-end delay is 22 and maximum delay jitter is 4. When the requirements are  $\delta > 22$  and  $\bar{\delta} > 4$ , the multicast tree is acceptable and the service request can be implemented.

Based on the architecture of software defined network (SDN) and NFV, there are control plane and data plane. Control plane refers to all the functions and processes that determine which path to use. Data plane refers to all the functions and processes that forward packets/frames from one interface to another. The algorithm is running on the control plane and all the resource state of nodes and links are stored on the control plane.

Many service requests may use the same link due to the lowest cost. The service request will choose other links when the resource of lowest cost link does not satisfy the requirement of the service request. In our simulation, the cost matrix varies according to the load of VM nodes when a new service request arrives. Therefore, the lowest cost is not always along the same links.

**4.7. Computation Complexity Analysis**

To construct a multicast tree, we need to transform a network into a MOD network. In Algorithm 1, the shortest path between each two VM node pair is needed. We can get the shortest paths from a VM node to other VM nodes by the Dijkstra algorithm. There are  $|V|$  nodes; therefore, the time complexity is  $O(|V|^2)$ . The total time complexity of Algorithm 1 is  $O(|V_M| \times |V|^2)$  because  $|V_M|$  VM nodes find paths.

In Algorithm 3, we search for a path to deploy the SFC in MOD network  $G'$  with the improved Dijkstra algorithm. It is accomplished in  $O((|L| \times |V_M| + |V_S| + 1)^2)$  time. Then, we need to construct a Steiner tree to find the optimal routing path. In the process of building the Steiner tree, we have gotten the shortest path between every two nodes. The time complexity is  $O(|V - 1|^2)$ . The time complexity of recovery cost is  $O(|V_D| \times |V_r|)$ , where  $|V_r|$  is the number of recovery nodes in the path from  $S$  to  $d$ . The time complexity of algorithm 3 is  $[O((|L| \times |V_M| + |V_S| + 1)^2 + |V_M| \times |V - 1|^2 + |V_D| \times |V_r|)]$ , which can be shortened as  $O((|L| \times |V_M| + |V_S|)^2 + |V_M| \times |V|^2)$ . As a result, the total complexity is the sum of the two algorithms.

## 5. Performance Evaluation

### 5.1. Simulation Environment

The performance of the proposed algorithm is evaluated in two real network topologies, the backbone networks Internet2 [46] and inter-data-center network Cogent [47]. The topology of Internet2 is shown in Figure 9. Parameters of all the networks are presented in Table 2.



Figure 9. The topology of Internet2.

Table 2. Properties of the networks.

Case	Node	Link	Data Center
Internet2	64	69	1
Cogent	190	260	40

The bandwidth capacity of the link varies from 1000 Mbps to 10,000 Mbps [48]. The Euclidean distance is the bandwidth consumption of the link. The delay of the link varies from 15 to 25 uniformly. According to [49,50], the loss rate of the link is in the range of 1% to 10%. Moreover, the computing capacity of VM is in the range of 10 to 20.

The request service chain is generated from a set  $F$  consisting of 30 distinct VNFs [18]. Some parts of the VNFs have been deployed on VM nodes, and several VNFs are selected from  $F$  and linked as a service chain. The length of the service chain is 5.

The cost matrix is the cost of VM nodes to implement VNFs. The cost is different for implementing different VNFs on VM nodes. Therefore, the elements of the cost matrix are not the same. In addition, the cost of the same VNF on a VM node may be different due to the load of the VM node. Therefore, we assume that the cost matrix varies from service request to service request. The service requests arrive sequentially. When a service request arrives, a cost matrix is generated for this service request. Before the service request is implemented, the cost matrix does not vary. When a new service request arrives, a new cost matrix is generated for the new service request. A  $n \times k$  matrix is generated at random as the deployment cost matrix  $DC$ , where  $n$  is the number of VM nodes, and  $k$  is the length of the service chain. The value of the entry in the matrix is in the range of 0 to 10. When  $DC_{i,j}$  is zero, the deployment cost of the VFN  $j$  on the VM node  $i$  is zero because VFN  $j$  has been deployed on VM node  $i$ .

The default setting of parameter weight  $\alpha$  is 1; loss rate  $\beta$  is 5%; the number of the source nodes is 2; the number of the destination nodes is 10; the number of the recovery nodes is 7.

We compare the proposed algorithm with the following ones.



1. Two Stages Service Function Tree Embedding Algorithm (TSE) [41]: It also finds a multicast tree with minimum cost when some VNFs have run on some VM nodes. However, the reliability of this tree solution is not guaranteed. Therefore, we do some modification to support reliability by treating the source as recovery node and considering the recovery cost as a part of the total cost.
2. Recovery-aware Steiner Tree Algorithm (RST) [29]: It guarantees the reliability but does not have provisions for embedding SFC. Therefore, we find the shortest path  $P$  from the virtual source to embed the SFC, then find a Steiner tree and combine them to let the flow arrive at all destination nodes. Path  $P$  should contain the recovery node to provide reliable routing. If Path  $P$  does not contain a recovery node, we connect  $P$  with the closest recovery node.

We consider the effect of the loss rate  $\beta$ , the number of destinations, the number of recover nodes, and weight  $\alpha$  on the performance of the three different schemes with the same simulation settings.

## 5.2. Simulation Results

### 5.2.1. Impact of Loss Rate $\beta$

The loss rate  $\beta$  varies from 0 to 0.1. The total cost for the multicast tree construction increases as the loss rate increases in both the network topologies, as shown in Figure 10. When the loss rate is zero, then the costs of RMTC and TSE are equal, and no packet loss will mean there is no recovery cost. RMTC and TSE use the same method in the multicast tree construction phase. Therefore, the costs of RMTC and TSE are the same.

The cost of RMTC is always lower than that of the other two schemes at a non-zero loss rate. Compared with the other two schemes, the trend of RMTC is slower with the increase of loss rate because RMTC consumes less resources in the reliability guarantee phase.

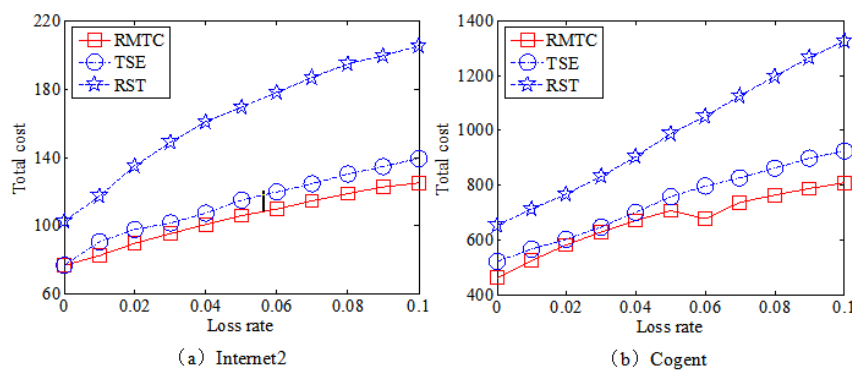
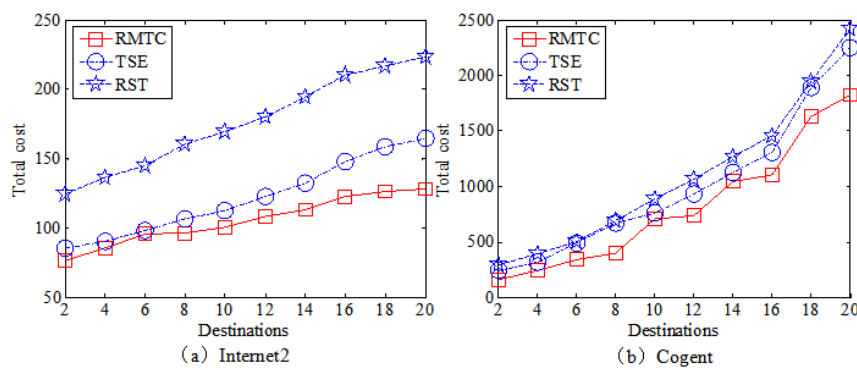


Figure 10. Impact of loss rate  $\beta$  in (a) Internet2; (b) Cogent.

### 5.2.2. Impact of the Number of Destinations

Figure 11 shows that the total cost increases as the number of destination increases before achieving the optimal cost for different topologies. The number of destinations grow from 2 to 20 at intervals of 2. Specifically, Figure 11a reveals that the increasing trend of RMTC and TSE are slow in Internet2 network. The number of destination nodes does not make a big difference for both RMTC and TSE, due to the fact that the range of route tree constructed by RMTC and TSE is wide. When adding or removing some nodes, the backbone of the route tree remains. For the cross-domain Cogent network, it is not easy to find a route tree to cover all destinations because of the large scale. Therefore, the increasing trend of the three schemes are similar in the Cogent network.

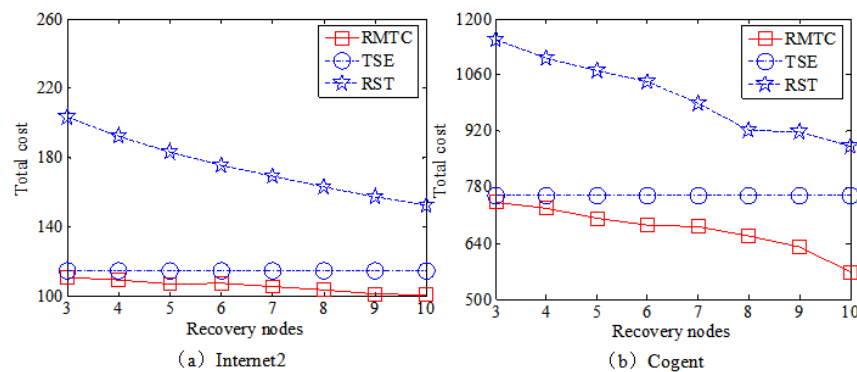
In addition, the total cost of RMTC and TSE are significantly lower than that of RST because of the low construction cost. The recovery path of RMTC is shorter at the same loss packet rate. Therefore, the recovery cost and total cost of RMTC are lower than that of TSE.



**Figure 11.** Impact of the number of destinations in (a) Internet2, (b) Cogent.

### 5.2.3. Impact of the Number of Recovery Nodes

In this subsection, the number of the recovery node is set from 3 to 10. Figure 12 illuminates that the line of TSE is horizontal, meaning that the total cost of TSE has no business relation to the number of recovery node. The re-route of TSE is only from the source nodes and, as the number of recovery nodes increase, the total costs of the other two schemes decline. When there are more recovery nodes, there are more available choices for the re-route 362 node and the total cost becomes low.



**Figure 12.** Impact of the number of recovery nodes in (a) Internet2, (b) Cogent.

### 5.2.4. Impact of Weight $\alpha$

To characterize the effect of weight  $\alpha$  on the total cost, we change the value of  $\alpha$  from 0 to 1 at intervals of 0.1 and from 1 to 10 at intervals of 1. Figure 13 shows that the total cost of the three schemes increase as weight  $\alpha$  increases. The increase of RST is significantly higher than that of RMTC and TSE because of the higher recovery cost in RST. When recovery weight  $\alpha$  is high, the total cost of RST grows fairly rapidly. When recovery weight  $\alpha$  is zero, then there is no recovery cost and the total cost is equal to the construction cost. RMTC consumes lower construction cost and recovery cost.

Figure 14 shows the total cost as weight  $\alpha$  grows from 1 to 10. The trend of total cost is similar in the two different topologies and the high recovery cost results in the rapid growth of the total cost in RST.

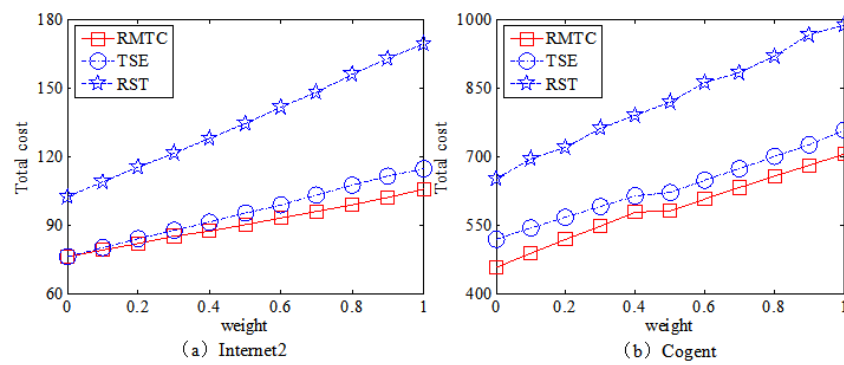


Figure 13. Impact of weight  $\alpha$  from 0 to 1 in (a) Internet2, (b) Cogent.

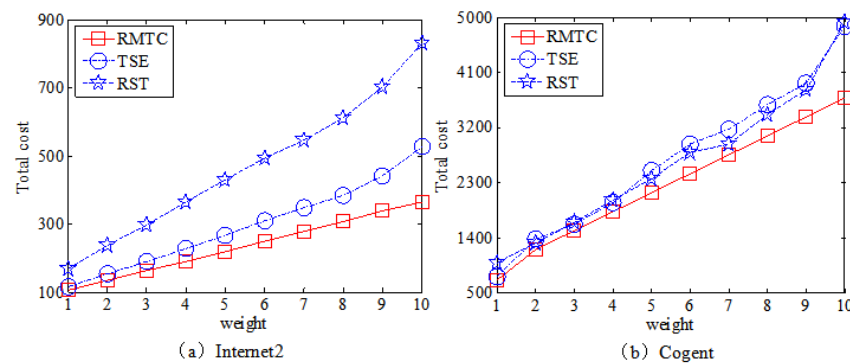


Figure 14. Impact of weight  $\alpha$  from 1 to 10 in (a) Internet2, (b) Cogent.

## 6. Conclusions

This paper focuses on the problem of reliable Virtual Network Function (VNF) layout with various deployment costs in multi-source multicast, while minimizing resource consumption and guaranteeing the Quality of Service (QoS). To guarantee the QoS, we take account of bandwidth, latency and reliability constraints. We first formulate our MRMRQC scheme into a MILP problem and prove its NP-hard hardness. We propose a heuristic algorithm, named Multi-Source Reliable Multicast Tree Construction (RMTC). This algorithm intends to find a common link to place the Service Function Chain (SFC) in the multilevel overlay directed (MOD) network of the original network, so that the deployed SFC can be shared by all users, thereby improving resource utilization. Two real topologies, namely backbone networks Internet2 and inter-data-center network Cogent, are adopted to evaluate the performance of the proposed algorithm. Simulation results demonstrate that our scheme effectively reduces the cost of reliable services and satisfies the QoS requirements.

**Author Contributions:** S.H. conceived the mechanism design and wrote the paper, K.X. gave the validation, X.Z. performed simulations, T.S. reviewed the paper, and J.W. revised the manuscript and contributed to the refinement of the article.

**Funding:** This research was funded by the National Natural Science Foundation of China grant number 61802030 and 61572184, the Hunan Provincial Natural Science Foundation of China grant number 2017JJ1010 and 2016JC2075, the Open Research Fund of Key Lab of Broadband Wireless Communication and Sensor Network Technology (Nanjing University of Posts and Telecommunications), Ministry of Education grant number JZNY201905, the International Cooperative Project for “Double First-Class”, CSUST grant number 2018IC24, the Open Research Fund of the Hunan Provincial Key Laboratory of Network Investigational Technology grant number 2018WLZC003, the open project funding of CAS KeyLab of Network Data Science and Technology grant number CASNDST201704, and the open project funding of State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences grant number CARCH201809, and the Open Foundation of State Key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) grant number SKLNST-2018-1-20.

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

## Abbreviations

The following abbreviations are used in this manuscript:

EC	Edge Computing
NFV	Network Function Virtualization
VNF	Virtual Network Function
QoS	Quality of Service
RMTC	Multi-Source Reliable Multicast Tree Construction
SFC	Service Function Chain
MOD	Multilevel Overlay Directed
NFs	Network Functions
VM	Virtual Machine
MRMRQC	Multi-Source Reliable Multicast Routing with QoS Constraints
MMRQC	Multi-Source Multi-Cast Routing with QoS Constraints
ST	Steiner Tree
SDN	Software Defined Network

## References

1. Yin, Y.; Chen, L.; Xu, Y.; Wan, J.; Zhang, H.; Mai, Z. QoS prediction for service recommendation with deep feature learning in edge computing environment. *Mob. Netw. Appl.* **2019**, 1–11. [CrossRef]
2. Yin, Y.; Xu, Y.; Xu, W.; Min, G.; Pei, Y. Collaborative Service Selection via Ensemble Learning in Mixed Mobile Network Environments. *Entropy* **2017**, 19, 358. [CrossRef]
3. Li, W.; Chen, Z.; Gao, X.; Liu, W.; Wang, J. Multi-Model Framework for Indoor Localization under Mobile Edge Computing Environment. *IEEE Internet Things J.* **2019**, 6, 4844–4853. [CrossRef]
4. Wang, J.; Gao, Y.; Yin, X.; Li, F.; Kim, H.J. An Enhanced PEGASIS Algorithm with Mobile Sink Support for Wireless Sensor Networks. *Wirel. Commun. Mob. Comput.* **2018**, 2018, 1–9. [CrossRef]
5. Nguyen, T.T.; Pan, J.S.; Dao, T.K. An Improved Flower Pollination Algorithm for Optimizing Layouts of Nodes in Wireless Sensor Network. *IEEE Access* **2019**, 7, 75985–75998. [CrossRef]
6. He, S.; Tang, Y.; Li, Z.; Li, F.; Xie, K.; Kim, H.J.; Kim, G.J. Interference-Aware Routing for Difficult Wireless Sensor Network Environment with SWIPT. *Sensors* **2019**, 19, 3978. [CrossRef] [PubMed]
7. Network Functions Virtualization-Introductory White Paper. Available online: [http://portal.etsi.org/NFV/NFV\\_White\\_Paper.pdf](http://portal.etsi.org/NFV/NFV_White_Paper.pdf) (accessed on 10 May 2018).
8. Xie, K.; Li, X.; Wang, X.; Xie, G.; Wen, J.; Cao, J.; Zhang, D. Fast tensor factorization for accurate Internet anomaly detection. *IEEE/ACM Trans. Netw.* **2017**, 25, 3794–3807. [CrossRef]
9. Xie, K.; Wang, L.; Wang, X.; Xie, G.; Wen, J.; Zhang, G.; Cao, J.; Zhang, D.; Xie, K.; Wang, X.; et al. Accurate Recovery of Internet Traffic Data: A Sequential Tensor Completion Approach. *IEEE/ACM Trans. Netw. (TON)* **2018**, 26, 793–806. [CrossRef]
10. Gu, S.; Li, Z.; Wu, C.; Huang, C. An efficient auction mechanism for service chains in the NFV market. In Proceedings of the 2016 IEEE Conference on Computer Communications (INFOCOM), San Francisco, CA, USA, 10–15 April 2016; pp. 1–9.
11. Li, L.E.; Liaghat, V.; Zhao, H.; Hajiaghay, M. PACE: Policy-Aware Application Cloud Embedding. In Proceedings of the 2013 IEEE Conference on Computer Communications (INFOCOM), Turin, Italy, 14–19 April 2013; pp. 638–646.
12. Anwer, B.; Benson, T.; Feamster, N.; Levin, D. Programming slick network functions. In Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research, Santa Clara, CA, USA, 17–18 June 2015; pp. 1–13.
13. Palkar, S.; Lan, C.; Han, S.; Jang, K.; Panda, A.; Ratnasamy, S.; Rizzo, L.; Shenker, S. E2:A framework for NFV applications. In Proceedings of the Symposium on Operating Systems Principles, Monterey, CA, USA, 4–7 October 2015; pp. 121–136.

14. Cohen, R.; Lewin-Eytan, L.; Naor, J.S.; Raz, D. On the effect of forwarding table size on SDN network utilization. In Proceedings of the 2014 IEEE Conference on Computer Communications (INFOCOM), Toronto, ON, Canada, 27 April–2 May 2014; pp. 1734–1742.
15. Cohen, R.; Lewin-Eytan, L.; Naor, J.S.; Raz, D. Near optimal placement of virtual network functions. In Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM), Hong Kong, China, 26 April–1 May 2015; pp. 1346–1354.
16. Li, Y.; Phan, L.T.X.; Loo, B.T. Network functions virtualization with soft real-time guarantees. In Proceedings of the 2016 IEEE Conference on Computer Communications (INFOCOM), San Francisco, CA, USA, 10–15 April 2016; pp. 1–9.
17. Lukovszki, T.; Schmid, S. Online Admission Control and Embedding of Service Chains. In Proceedings of the International Colloquium on Structural Information and Communication Complexity, Montserrat, Spain, 14–16 July 2015; pp. 104–118.
18. Kuo, T.W.; Liou, B.H.; Lin, C.J.; Tsai, M.J. Deploying chains of virtual network functions: On the relation between link and server usage. In Proceedings of the 2016 IEEE Conference on Computer Communications (INFOCOM), San Francisco, CA, USA, 10–15 April 2016; pp. 1–9.
19. Zhang, J.; Jin, X.; Sun, J.; Wang, J.; Sangaih, A.K. Spatial and semantic convolutional features for robust visual object tracking. *Multimed. Tools. Appl.* **2018**. [\[CrossRef\]](#)
20. Zhang, J.; Jin, X.; Sum, J.; Wang, J.; Li, K. Dual model learning combined with multiple feature selection for accurate visual tracking. *IEEE Access* **2019**, *7*, 43956–43969. [\[CrossRef\]](#)
21. Chen, Y.; Wang, J.; Chen, X.; Zhu, M.; Kai, Y.; Wang, Z.; Xia, R. Single-Image Super-Resolution Algorithm Based on Structural Self-Similarity and Deformation Block Features. *IEEE Access* **2019**, *7*, 58791–58801. [\[CrossRef\]](#)
22. He, S.; Li, Z.; Tang, Y.; Liao, Z.; Jin, W. Parameters Compressing in Deep Learning. *Comput. Mater. Continua* **2019**, 1–16.
23. Xie, K.; Peng, C.; Wang, X.; Xie, G.; Wen, J.; Cao, J.; Zhang, D.; Qin, Z. Accurate Recovery of Internet Traffic Data Under Variable Rate Measurements. *IEEE/ACM Trans. Netw.* **2018**, *26*, 1137–1150. [\[CrossRef\]](#)
24. Xie, K.; Li, X.; Wang, X.; Cao, J.; Xie, G.; Wen, J.; Zhang, D.; Qin, Z. On-Line Anomaly Detection with High Accuracy. *IEEE/ACM Trans. Netw.* **2018**, *26*, 1222–1235. [\[CrossRef\]](#)
25. Wang, J.; Zhang, Z.; Li, B.; Lee, S.; Sherratt, R.S. An enhanced fall detection system for elderly person monitoring using consumer home networks. *IEEE/ACM Trans. Consum. Electron.* **2014**, *60*, 23–29. [\[CrossRef\]](#)
26. Rajkumar, M.; Zhang, X.; Qiao, C. Benefits of multicasting in all-optical networks. In *All-Optical Networking: Architecture, Control, and Management Issues*; SPIE: Boston, MA, USA, 1998; Volume 3531, pp. C805–C805.
27. Xu, Z.; Liang, W.; Huang, M.; Jia, M.; Guo, S.; Galis, A. Approximation and Online Algorithms for NFV-Enabled Multicasting in SDNs. In Proceedings of the IEEE 36th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 5–8 June 2017; pp. 625–634.
28. Molnar, M. Exact Algorithm to Solve the Minimum Cost Multi-Constrained Multicast Routing Problem. *J. Comput. Commun.* **2016**, *4*, 57–79. [\[CrossRef\]](#)
29. Shen, S.H.; Huang, L.H.; Yang, D.N.; Chen, W.T. Reliable multicast routing for software-defined networks. In Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM), Hong Kong, China, 26 April–1 May 2015; pp. 181–189.
30. Huang, L.H.; Hsu, H.C.; Shen, S.H.; Yang, D.N.; Chen, W.T. Multicast traffic engineering for software-defined networks. In Proceedings of the 2016 IEEE Conference on Computer Communications (INFOCOM), San Francisco, CA, USA, 10–15 April 2016; pp. 1–9.
31. Kuo, J.J.; Shen, S.H.; Yang, M.H.; Yang, D.N.; Tsai, M.J.; Chen, W.T. Service overlay forest embedding for software-defined cloud networks. In Proceedings of the IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 5–8 June 2017; pp. 720–730.
32. Xie, K.; Zhou, X.; Semong, T.; He, S. Multi-Source Multicast Routing with QoS Constraints in Network Function Virtualization. In Proceedings of the IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–6.
33. Babarczia, P.; Pasica, A.; Tapolcaia, J.; Nemethb, F.; Ladoczkia, B. Instantaneous recovery of unicast connections in transport networks: Routing versus coding. *Comput. Netw.* **2015**, *82*, 68–80. [\[CrossRef\]](#)
34. Ayoubi, S.; Assi, C.; Chen, Y.; Khalifa, T.; Shaban, K.B. Restoration methods for cloud multicast virtual networks. *J. Netw. Comput. Appl.* **2017**, *78*, 180–190. [\[CrossRef\]](#)

35. Xie, K.; Wang, X.; Liu, X.; Wen, J.; Cao, J. Interference-Aware Cooperative Communication in Multi-radio Multi-channel Wireless Networks. *IEEE Trans. Comput.* **2016**, *65*, 1528–1542. [CrossRef]
36. Xie, K.; Wang, X.; Wen, J.; Cao, J. Cooperative routing with relay assignment in multiradio multihop wireless networks. *IEEE/ACM Trans. Netw.* **2015**, *24*, 859–872. [CrossRef]
37. Xie, K.; Ning, X.; Wang, X.; Xie, D.; Cao, J.; Xie, G.; Wen, J. Recover corrupted data in sensor networks: A matrix completion solution. *IEEE Trans. Mob. Comput.* **2017**, *16*, 1434–1448. [CrossRef]
38. Xie, K.; Ning, X.; Wang, X.; He, S.; Ning, Z.; Liu, X.; Wen, J.; Qin, Z. An efficient privacy-preserving compressive data gathering scheme in WSNs. *Inf. Sci.* **2017**, *390*, 82–94. [CrossRef]
39. Wang, J.; Ju, C.; Kim, H.j.; Sherratt, R.S.; Lee, S. A mobile assisted coverage hole patching scheme based on particle swarm optimization for WSNs. *Clust. Comput.* **2019**, *22*, 1787–1795. [CrossRef]
40. Wang, J.; Gu, X.; Liu, W.; Sangaiah, A.K.; Kim, H.J. An empower hamilton loop based data collection algorithm with mobile agent for WSNs. *Hum.-Centric Comput. Inf. Sci.* **2019**, *9*, 18. [CrossRef]
41. Ren, B.; Guo, D.; Tang, G.; Lin, X.; Qin, Y. Optimal Service Function Tree Embedding for NFV Enabled Multicast. In Proceedings of the IEEE 38th International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria, 2–5 July 2018; pp. 132–142.
42. He, S.; Xie, K.; Chen, W.; Zhang, D.; Wen, J. Energy-aware Routing for SWIPT in Multi-hop Energy-constrained Wireless Network. *IEEE Access* **2018**, *6*, 17996–18008. [CrossRef]
43. He, S.; Xie, K.; Xie, K.; Xu, C.; Jin, W. Interference-aware Multi-source Transmission in Multi-radio and Multi-channel Wireless Network. *IEEE Syst. J.* **2019**, *13*, 2507–2518. [CrossRef]
44. Wang, J.; Gao, Y.; Liu, W.; Sangaiah, A.K.; Kim, H.J. An intelligent data gathering schema with data fusion supported for mobile sink in wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **2019**. [CrossRef]
45. Wang, J.; Gao, Y.; Liu, W.; Wenbing, W.; Se-Jung, L. An Asynchronous Clustering and Mobile Data Gathering Schema based on Timer Mechanism in Wireless Sensor Networks. *Comput. Mater. Contin.* **2019**, *58*, 711–725. [CrossRef]
46. Internet2 Network Infrastructure Topology. Available online: <https://www.internet2.edu/> (accessed on 12 May 2018).
47. Cogent's Network Map. Available online: <http://cogentco.com/en/network/network-map> (accessed on 13 May 2018).
48. Knight, S.; Nguyen, H.X.; Falkner, N.; Bowden, R.; Roughan, M. The internet topology zoo. *IEEE J. Sel. Areas Commun.* **2011**, *29*, 1765–1775. [CrossRef]
49. Xu, Y.; Yu, C.; Li, J.; Liu, Y. Video telephony for end-consumers: Measurement study of Google+, iChat, and Skype. *IEEE/ACM Trans. Netw.* **2014**, *22*, 826–839. [CrossRef]
50. Nguyen, H.X.; Roughan, M. Rigorous statistical analysis of internet loss measurements. *Acm Sigmetrics Perform. Eval. Rev. - Perform. Eval. Rev.* **2010**, *38*, 361–362. [CrossRef]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).