Exploiting Reliable and Scalable Multicast Services in IaaS Datacenters

Junjie Xie, Deke Guo, Jie Wu, Bangbang Ren, Tao Chen, Honghui Chen

Abstract—A large number of servers are interconnected using a specific datacenter network to deliver the infrastructure as a service (IaaS). Multicast can jointly utilize the network resources and further reduce the consumption of network bandwidth more than individual unicast. The source of a multicast service, however, does not need to be in a specific location as long as certain constraints are satisfied. This means the multicast can have uncertain sources, which could reduce the network resource consumption more than a traditional multicast service and further improve the quality of service. In this paper, we propose a novel reliable multicast service with uncertain sources named ReMUS. The goal is to minimize the sum of the transfer cost and the recovery cost, although finding such a ReMUS is very challenging. Thus, we design a source-based multicast method to solve this problem by exploiting the flexibility of sources when no recovery nodes exist in the network. Furthermore, we design a general multicast method to jointly exploit the benefits of uncertain sources and recovery nodes to minimize the total cost of ReMUS. We conduct extensive evaluations under Internet2 and datacenter networks. The results indicate that our methods can efficiently realize the reliable and scalable multicast with uncertain sources, irrespective of the settings of networks and multicasts. To the best of our knowledge, we are the first to study the reliable multicast service under uncertain sources.

Index Terms—Reliable multicast, Data center, Content replica, Uncertain sources.

1 Introduction

Infrastructure as a Service (IaaS) is a form of cloud computing, which enables tenants to multiplex computing, storage and network resources in data centers. Multicast is an efficient method for delivering the same content from a source node to a group of destinations in those data centers. It can considerably save the amount of consumption of network resources more than a series of individual unicasts. Owing to avoiding unnecessary traffic duplication in intermediate nodes and links, multicast can effectively reduce the bandwidth consumption by around 50% in backbone networks [1]. Meanwhile, multicast can release the loads of the source node and associated network links [2].

A number of novel multicast methods have been proposed recently and can be roughly divided into two categories. The first one focuses on reducing the consumption of network bandwidth. Many multicast services prefer to deliver the same content to a group of destinations along a shortest-path tree, such as PIM-SM [3]. The multicast tree can reduce more bandwidth consumption if those shortest paths from the same source to the destinations share more links. The second one aims to ensure the reliable transmission of multicast. Nowadays, the reliable multicast becomes crucial in providing reliable services for many important Internet and datacenter applications [4]. To achieve a reliable transmission of multicast, the source-based reliable multicast methods aim to recover the lost packets from the source node directly [5]. It, however, suffers from the reliability and availability problems since only one source node serves the recovery requests from all destinations. Accordingly, Shen et al. propose the Recover-aware Steiner Tree (RST) problem [6]. They introduce at least one recovery node between the source and each destination to facilitate the local loss recovery. Then, they design an approximation algorithm, RAERA, to minimize the sum of the tree cost and the recovery cost.

Besides the above two categories, the traditional multicast experiences uncertain sources [7][8] which bring new challenges and opportunities to the design of multicast methods. The root cause is the wide usage of the content replica strategy in various networks. For example, each file block in GFS [9] and HDFS [10] has at least two replicas besides the original one across the datacenter. Furthermore, many content delivery services have adopted the content replica design to improve its robustness and efficiency [11], [12]. Each replica of a given file has the capability to serve as a source node for a multicast transfer. That is, the source of a multicast does not need to be fixed at a specific location as long as certain constraints are satisfied. This brings the multicast with uncertain sources, which further reduces the consumption of the network bandwidth.

In this paper, we reveal the reliable and bandwidth-efficient multicast service with uncertain sources, named as the ReMUS problem. A source node or recovery node will retransmit lost packets towards a destination and incur the recovery cost in the case of packet loss. The goal of ReMUS is to jointly minimize the cost of transfers and recoveries for the reliable multicast service. The ReMUS problem faces fundamental challenging issues. First, the bandwidth-efficiency and reliability somehow conflict with each other.

- *Corresponding authors: Deke Guo; Tao Chen.*
- *Junjie Xie, Deke Guo, Bangbang Ren, Tao Chen and Honghui Chen are with the Science and Technology Laboratory on Information Systems Engineering, National University of Defense Technology, Changsha Hunan, 410073, China. E-mail: {xiejunjie06, guodeke}@gmail.com.*
- *Jie Wu the Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA. E-mail: jiewu@temple.edu*

The use of recovery nodes is effective for realizing the reliable multicast, but can directly change the design of a multicast tree by increasing the transfer cost. Second, the appearance of uncertain sources incurs complicated impacts on the transfer cost as well as the recovery cost. Uncertain sources need to be carefully scheduled to reduce the transfer cost. Moreover, uncertain sources are also helpful in reducing the recovery cost.

Hu et al. proposed the minimum cost forest (MCF) building method for the multicast with uncertain sources [7]. It aims to minimize the transfer cost by carefully exploiting the flexibility of those uncertain sources, but does not support reliable multicast transmission. The RAERA method motivates to ensure the reliable transmission for the traditional multicast with recovery nodes [6]. It, however, neither supports the multicast of uncertain sources nor considers the cooperative use of uncertain sources and recovery nodes. Despite multicast with uncertain sources or failures has been studied individually, such prior methods and their simple combinations remain inapplicable to the ReMUS problem.

The challenge of ReMUS stems from the selection of source nodes and recovery nodes and their impacts on routing cost. Thus, we propose two efficient methods to approximate the optimal solution under two general scenarios. If no recovery nodes exist in the network, the source-based multicast method constructs the desired transfer forest, where all sources in the forest act as the recovery proxies. By contrast, a general building method of a multicast forest is designed, given the number and locations of recovery nodes. The building process of the multicast forest will employ some necessary recovery nodes, which act as the recovery proxies with some sources together if necessary. Under both scenarios, an intrinsic constraint about the forest is that each destination only reaches one source, no matter how many uncertain sources are employed by the forest. We then conduct extensive evaluations in Internet2 and datacenter networks. The results indicate that our two methods can efficiently realize the reliable and bandwidth-efficient multicast under uncertain sources, irrespective of the settings of networks and multicasts.

The major contributions of this paper are summarized as follows:

1) We first propose the reliable multicast service with uncertain sources (ReMUS) problem, and then, formally characterize the problem.
2) We design the source-based reliable multicast method, which can minimize the total cost of ReMUS when no recovery nodes exist in the network.
3) We further design a general recovery model for the ReMUS problem and propose the recovery node-based multicast method, which jointly exploits both recovery nodes and uncertain sources in networks.
4) We conduct extensive evaluations in Internet2 and datacenter networks. The results show the efficiency and effectiveness of our two methods under the various settings of networks and multicasts.

The rest of this paper is organized as follows. In section 2, we first present the background and preliminaries and then introduce the statement and formulation of the ReMUS problem. Section 3 designs the source-based and general methods for constructing a desired multicast forest to solve the ReMUS problem. In section 4, we evaluate the performances of our methods under the network topologies of Internet2 and datacenter networks. We summarize the related work and conclude this paper in Section 5 and Section 6, respectively.

2 Problem Description of ReMUS

In this section, we first introduce the background and preliminaries. After that, we describe the problem of ReMUS, and then formally characterize the ReMUS as an NLP problem and discuss its hardness.

2.1 Background and preliminaries

We introduce some background and preliminaries about the reliable multicast and the multicast with uncertain sources.

2.1.1 Reliable multicast

The problem of reliable *unicast* data delivery is well understood and a variety of solutions have been proposed. However, the multicast transmission offers the most promising approach for the reliable transmission of data to a potentially large group of receivers. For source-based reliable multicast (SRM) [5], when receiver(s) detect the missing data, they wait for a random time determined by their distance from the original source of the data then send a repair request. After the source receives a repair request, the source re-delivers lost packets to the corresponding receiver.

For example, when packet loss occurs in any one multicast link in Fig. 1(a), source s_1 retransfers these lost packets to the corresponding destinations via unicast. The source-based reliable multicast suffers from the reliability and scalability problems since the single source needs to provide loss recovery for a large number of destinations. To effectively address this crucial issue, recovery nodes are placed between the source and the destinations to facilitate the local loss recovery. The idea is similar to the web and multimedia cache/proxy servers wide deployed today, with the goal to facilitate local services and avoid load concentration on the source node. For a multicast tree, it is necessary to span suitable recovery nodes for local loss recovery and the minimization of the recovery cost.

Shen et al. proposed a new reliable multicast tree for the SDN, named Recover-aware Steiner Tree (RST) [6]. The goal of RST is to minimize both tree and recovery costs by the selection of recovery nodes. Fig. 1(b) shows a multicast tree, which spans suitable recovery nodes. In Fig. 1(b), r_4 will deliver lost packets to d_6 facilitating the local loss recovery, when d_6 fails to receive some packets. It is remarkable that the recovery from node r_4 will incur less recovery cost than recovering from source s_1. Therefore, the local recovery can reduce the unnecessary consumption of bandwidth.

2.1.2 Multicast with uncertain sources

Traditional multicast methods, however, assume prior knowledge of multicast characteristics, i.e., the source of each multicast group is fixed in advance. It is called as the *deterministic* multicast. However, in many cases, the characteristics of a multicast service are unknown in advance. In

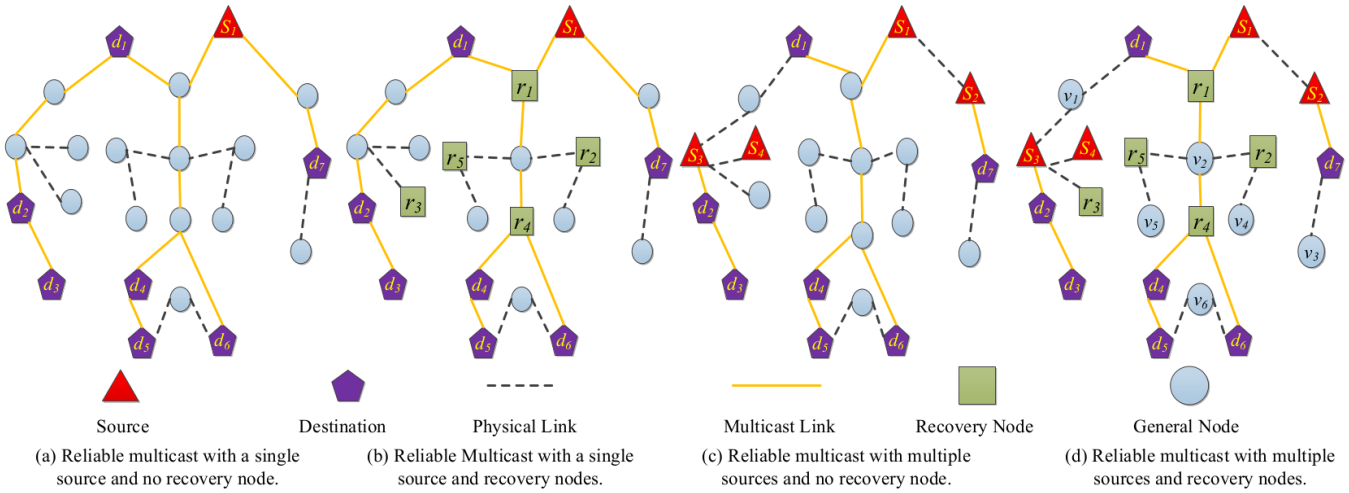


Fig. 1. Reliable multicast with the same destinations $\{d_1,d_2,d_3,d_4,d_5,d_6,d_7\}$ but different sources and recovery nodes.

this case, the multicast is with uncertain sources, as shown in Definition 2.1.

Definition 2.1 (Uncertain sources). Given a set of sources, some sources can be selected as the root nodes for a multicast service at the same time. In this case, for a multicast service, the number of root nodes and their locations are uncertain.

A major reason for uncertain sources is the wide used content replica designs for improving the robustness and efficiency in various networks. While delivering a content file to multiple destinations, the source of such a multicast service can be any one replica in theory [7][8]. Those data copies all can be utilized for a multicast service. The key challenge is how to select the source nodes for multicast with uncertain sources. Hu et al. proposed a new multicast scheme with uncertain sources for SDN, called uncertain multicast [7]. Given destinations and uncertain sources, together with the network topology, the uncertain multicast problem aims to construct a forest, which achieves the minimal cost of multicast transmission. More precisely, consider a network $G(V, E)$, where V and E denote the set of nodes (switches and servers) and edges (links), respectively. Each edge $e{\in}E$ is associated with a cost. Without of generality, the weight of each edge is set to unity to normalize the total transmission cost of an uncertain multicast, which is formalized in Definition 2.2.

Definition 2.2 (Uncertain multicast). Given a destination set $D{\subset}V$ and a potential source set $S{\subset}V$, an *uncertain multicast* means to build a multicast forest F to deliver the same content to all destinations in the set D from partial even all sources in the set S. A constraint of the multicast forest F is that each destination just reaches to one and only one source in set S.

As shown in Fig. 1(c), the source set S includes four source nodes $\{s_1, s_2, s_3, s_4\}$. The solid lines denote that the corresponding links are employed for the multicast transfer in Fig. 1(c). The link between s_1 and s_2 is not employed, which means that destination d_7 can just reach to source s_2 and can not reach to source s_1. The constraint for the uncertain multicast indicates that the employed links for a multicast transfer can not connect two source nodes because that results in the waste of network bandwidth. In Fig. 1(c), the forest for the multicast with uncertain sources only employs 10 links. However, the multicast employs 13 links for the same destinations in Fig. 1(a). More links mean more bandwidth consumption. Therefore, uncertain sources are helpful to reduce the network bandwidth consumption.

2.2 Problem Statement

In this paper, we propose the problem of reliable multicast service with uncertain sources (ReMUS). The ReMUS problem finds a desired forest F, which employs the necessary sources and recovery nodes. A constraint is that each destination only reaches one source. Meanwhile, the sources and recovery nodes can be jointly exploited to achieve the reliable multicast service. We firstly give the definition of recovery proxy in Definition 2.3. Given the number and locations of sources and recovery nodes, the building process of multicast forest will decide the locations of recovery proxies. When there are lost packets, the recovery proxies will retransfer the lost packets.

Definition 2.3 (Recovery proxy). Assume that there exists a candidate recovery node set $C{\subset}V$ in $G(V, E)$. For an uncertain multicast with a source node set $S{\subset}V$ and a destination node set $D{\subset}V$, the *recovery proxy* of a destination is a related source or a recovery node on the path from the related source to the destination. When destination $d{\in}D$ fails to receive some packets, the *recovery proxy* can retransfer lost packets to the destination.

We then formalize the ReMUS problem in Definition 2.4.

Definition 2.4 (ReMUS problem). Given a destination set $D{\subset}V$, a potential source set $S{\subset}V$, and a candidate set of recovery nodes $C{\subset}V$, the ReMUS means to build a multicast forest F to deliver the same content to all destinations in the set D from partial even all sources in the set S. A constraint is that each destination just reaches to one and only one source in set S. Meanwhile, recovery proxies can retransfer lost packets to the corresponding destinations, when packet loss occurs.

The objective of the ReMUS problem is to minimize the sum of the transfer cost and the recovery cost. The transfer

cost is represented by $c(F)$, which is the sum of the edge cost $c(e)$ for every edge $e \in F$. The entire recovery cost is calculated by the product of the recovery cost of each destination and the retransfer probability. The recovery cost κ_d of each destination d is related to the cost of the path towards its recovery proxy r. The recovery cost of the ReMUS is represented by $w(F)$, which is the sum of the recovery cost κ_d for every node $d \in D$.

Given a non-negative value α, the optimization goal of the ReMUS problem is to find a subset H from source set S, a subset R from recovery set C and a forest F encompassing H, R and D such that $c(F)+\alpha w(F)$ is minimized. If the network is heavily loaded, it is necessary to assign a larger α such that the recovery nodes will play a more important role in order to effectively reduce the recovery cost. To clarify this problem, we present an illustrative example.

Illustrative examples of the ReMUS problem. We show the impact of multiple sources on the reliable and bandwidth-efficient multicast. For the ReMUS problem, where multiple sources $\{s_1, s_2, s_3, s_4\}$ are available in Fig. 1(d). Meanwhile, Fig. 1(d) plots a desired forest, which includes all destinations $\{d_1, d_2, d_3, d_4, d_5, d_6, d_7\}$, several recovery nodes, and necessary sources. Each destination reaches only one source through this forest. The multicast forest employs 10 links, while the multicast tree in Fig. 1(b) employs 13 links. This indicates that uncertain sources are very effective for reducing the transfer cost of a traditional multicast. Moreover, the recovery proxy of destination d_6 changes with recovery node r_4 in Fig. 1(d) from S_1 in Fig. 1(c) after introducing recovery nodes. In this way, the recovery cost of destination d_6 decreases since its recovery path is shortened. This indicates that recovery nodes have the potential to reduce the recovery cost.

Given the number of source and recovery nodes and their locations, to minimize the total cost of the ReMUS, we will jointly consider the cost of transfer and recovery. Therefore, it is essential to consider the locations of candidate sources and recovery nodes during constructing the multicast forest. Meanwhile, it is also feasible that no recovery nodes exist on the path from a source to a destination in the multicast forest; hence, the source will act as the recovery proxy of the corresponding destination.

2.3 The recovery models for the ReMUS

In practice, it is crucial to design dedicated recovery models for the ReMUS problem under different settings of recovery nodes. In this section, we rethink the source-based recovery model under uncertain sources, and then design a general recovery model when some recovery nodes have been deployed in networks.

2.3.1 Rethinking the source-based recovery model

To realize the reliable multicast, the source-based recovery model is an intrinsic way when there is no any recovery node in the network. For example, dedicated source-based methods have been presented to support the reliable multicast transfer with a single source [5]. As shown in Fig. 1(a), source s_1 will retransfer if any node or link on the path from s_1 to d_6 loses packets. Here, we rethink this kind of recovery model in the case of uncertain sources. Multiple potential sources bring new opportunities to reduce the transfer cost and recovery cost, as shown in Fig. 1(c).

For a multicast with uncertain sources, it is required to design a desired multicast forest to span all destinations and necessary sources. Consequently, each destination reaches only one source, while any pair of destinations may reach different sources. Thus, each destination should be recovered by the right source. The caused recovery cost for any destination is equal to the product of the retransfer probability and the cost of recovery path towards the right source in the multicast forest. The recovery cost of ReMUS problem means the sum of the recovery cost of all destinations.

We use β_e to denote the probability of packet loss on any link e in the network. Assume that the path from destination d to its source s consists of n links, $e_1, e_2, \ldots, e_n$, in the multicast forest. The retransfer probability from s to d is calculated in Equation (1), which is equal to the probability of the packet loss on at least one link in the path.

$$\beta_{s,d}=1-\prod_{i=1}^{n}(1-\beta_{e_i}) \tag{1}$$

In this case, the recovery cost of destination d is given by Equation (2).

$$\kappa_d=\beta_{s,d}\times\sum_{i=1}^{n}c(e_i) \tag{2}$$

The recovery cost of ReMUS is the sum of the recovery cost of all destinations.

2.3.2 Designing a general recovery model

If a set of recovery nodes are deployed in the network in advance, the resultant multicast forest for any uncertain multicast may employ some recovery nodes. In this way, recovery node r between source s and destination d in the forest would deliver local recovery, if needed. Under this circumstances, the recovery cost for destination d is equal to the retransfer probability times the cost of the recovery path from r to d in the forest. This recovery cost for destination d is definitely less than the recovery cost from source s since recovery node r is closer to destination d than source s. Additionally, if the recovery node r also fails to receive these lost packets or there exists no recovery node on the path from a source to a destination, it is reasonable that the destination can be recovered by the related source in the forest. Therefore, this general recovery model is more suitable than the recovery models solely based on sources or recovery nodes.

To characterize this general recovery model, let $R(d)$ denote the set of all recovery proxies at destination d. It contains not only all recovery nodes but also the only source on the path from the corresponding source to destination d in a multicast forest. Let a recovery proxy $r(d)_1$ denote the related source of destination d since the source can also retransfer lost packets if necessary. Let $\eta_{d,r(d)_i}$ denote the probability of recovery from a recovery proxy $r(d)_i$, $1\leq i\leq|R(d)|$, which is calculated in Equation (3).

$$\eta_{d,r(d)_i}=(1-\beta_{s,r(d)_i})\times\beta_{r(d)_i,r(d)_{i+1}}, 1\leq i\leq|R(d)| \tag{3}$$

When destination d needs to be recovered from $r(d)_i$, it means that $r(d)_i$ has received the lost packets, but $r(d)_{i+1}$

has not received those packets. Otherwise, destination d can be recovered from $r(d)_{i+1}$. In equation (3), $\beta_{s,r(d)_i}$ and $\beta_{r(d)_i,r(d)_{i+1}}$ denote the probability of packet loss in the path from s to $r(d)_i$ and the path from $r(d)_i$ to $r(d)_{i+1}$, respectively. $(1-\beta_{s,r(d)_i})$ denotes the probability of having no packet loss in the path from s to $r(d)_i$. Furthermore, $\beta_{s,r(d)_i}$ and $\beta_{r(d)_i,r(d)_{i+1}}$ in Equation (3) are derived as Equation (1) and node $r(d)_{|R(d)|+1}$ is used to denote destination d.

The recovery cost of a destination is the total recovery cost resulting from its all potential recovery paths, which are dominated by the recovery proxies. For example, destination d_6 has three recovery paths with the recovery proxies s_1, r_1, and r_4, respectively, in Fig. 1(d). Thus, the recovery cost of d_6 is composed of three parts, which is related to the three recovery proxies. For any one part, the recovery cost is equal to the probability of adopting the recovery proxy times the transfer cost of the corresponding recovery path. Finally, the entire recovery cost of the ReMUS problem is the total recovery cost of all destinations.

2.4 Problem Formulation

After presenting the definition of the ReMUS problem and designing effective recovery models, we then formally characterize the ReMUS problem, which can capture the complicated impact of settings and usage of uncertain sources and recovery nodes on the total cost of the desired multicast forest.

We use an undirected graph $G(V,E)$ to denote the network topology. S denotes the set of sources, and D denotes a group of destinations. F denotes the set of links employed by a multicast forest for the ReMUS problem. For any node v in G, let N_v denote the set of neighbor nodes of v in G. A binary variable $e_{u,v}$ denotes whether there is an edge between any node pair of u and v in V. Another binary variable $\tau_{u,v}$ denotes whether the edge $e_{u,v}$ is in F. A feasible multicast forest should ensure that each destination can reach at least one source. A binary variable $\mathcal{P}_{u,v}$ denotes if there is a path from u to v in F. If $\mathcal{P}_{s,d}$=1, source s will transfer the data to destination d. To ensure there is only one path from a source to a destination, variable $\pi_{d,(u,v)}$ is needed to denote if edge $e_{u,v}$ is in the path from source s to destination d in F. The multicast forest F is just the combination of those paths to all destinations. $c(e_{u,v})$ denotes the cost of link $e_{u,v}$, where $c(e_{u,v})\geq 0$. $c(\mathcal{P}_{u,v})$ denotes the transfer cost of path $\mathcal{P}_{u,v}$ and is equal to the sum of the cost of all edges in the path.

To calculate the recovery cost of a multicast forest, κ_d denotes the recovery cost of any destination d in F, which is calculated in Formula (4).

$$\kappa_d = c(\mathcal{P}_{r(d)_1,d}) \times \eta_{d,r(d)_1} + c(\mathcal{P}_{r(d)_2,d}) \times \eta_{d,r(d)_2} + \cdots + c(\mathcal{P}_{r(d)_{|R(d)|},d}) \times \eta_{d,r(d)_{|R(d)|}}, \forall d \in D \quad (4)$$

Equality (4) shows the recovery cost of each destination. For each destination d, there are multiple feasible recovery paths, resulting from those corresponding recovery proxies of destination d. The recovery cost of destination d is equal to the sum of the recovery cost from all its recovery proxies based on the recovery model in Section 2.3.2.

For any node u, a binary variable ρ_u denotes whether it is a recovery proxy. ρ_u=1, if and only if u is a recovery node or a source. Let C denote the set of all recovery nodes. As aforementioned, $R(d)$ records the set of recovery proxies of destination d. The recovery proxy $r(d)_i$ is in the path $\mathcal{P}_{s,d}$, $r(d)_i \in R(d)$, $1 \leq i \leq |R(d)|$. That is, node $r(d)_i$ will retransfer lost packets to destination d when $r(d)_i$ has cached the lost packets loss occurs and $r(d)_{i+1}$ has not cached those packets. $\mathcal{P}_{r(d)_i,r(d)_{i+1}}$ denotes the sub path between $r(d)_i$ and $r(d)_{i+1}$ in the path from s to d. $\eta_{d,r(d)_i}$ denotes the probability of adopting the recovery proxy $r(d)_i$ to recover destination d, and the corresponding recovery path is $\mathcal{P}_{r(d)_i,d}$. The value of $\eta_{d,r(d)_i}$ is calculated as Equation (3).

Therefore, the objective function of the ReMUS problem is given by

$$\min\{\sum_{e_{u,v}\in E} c_{u,v}\tau_{u,v} + \alpha \sum_{d\in D} \kappa_d\}. \quad (5)$$

That is to derive out a feasible multicast forest, which can minimize the transfer cost and the recovery cost. The transfer cost is equal to the sum cost of all employed links by the multicast transfer. The recovery cost of the ReMUS equals to the sum of the recovery cost of all destinations, where α is a regulative parameter. The above formulation is a Non-Linear Programming (NLP) model, because $c_{u,v}$ is not sure if it is an integer and κ_d consists of the product of the retransfer cost and the retransfer probability. Meanwhile, to ensure a practical and feasible multicast forest, the following constraints should be satisfied.

$$\tau_{u,v} \leq e_{u,v} \quad (6)$$

$$\sum_{v\in N_s} \pi_{d,(s,v)} = 1, \mathcal{P}_{s,d} = 1, \forall d \in D, \exists s \in S \quad (7)$$

$$\sum_{u\in N_d} \pi_{d,(u,d)} = 1, \forall d \in D \quad (8)$$

$$\sum_{u\in N_v} \pi_{d,(u,v)} = \sum_{u\in N_v} \pi_{d,(v,u)}, \forall d \in D, \forall u \in V, v \neq s, v \neq d \quad (9)$$

$$\pi_{d,(u,v)} \leq \tau_{u,v}, \forall d \in D \quad (10)$$

$$\sum_{s\in S} \mathcal{P}_{s,d} = 1, \forall d \in D \quad (11)$$

$$\rho_u = 0, \forall u \notin \{C \cup S\} \quad (12)$$

$$\{e_{u,v}, \tau_{u,v}, \pi_{d,(u,v)}, \rho_u, \mathcal{P}_{u,v}\} \in \{0,1\}, \forall u \in V, \forall v \in V, \forall d \in D, \forall s \in S \quad (13)$$

The constraint of multicast. Inequality (6) ensures that all links in the resultant multicast forest F come from the link set E. Constraints (7), (8) and (9) ensure that a pair of a source and destination are reachable along only one path in the forest F. Equality (9) can guarantee the connectivity and uniqueness of the path. Constraint (10) means that $\tau_{u,v}$ must be 1 when edge $e_{u,v}$ is added into the path from a source to a destination. For each destination, equality (11) ensures that it only reaches one source. The multicast forest F is just the combination of such paths to all destinations.

The constraint of recovery. Equality (4) shows the recovery cost of each destination. Constraint (12) means that ρ_u=0, if u is neither in set C nor in set S. If and only if node u is a employed recovery node or the used source in F,

Algorithm 1 SR-based building method of multicast forest.

Require: $G(V, E)$, S, D, α and β.
Ensure: multicast forest, F
1: **for** i=1 to $|D|$ **do**
2: $P_i \leftarrow$ shortest paths from d_i to S;
3: **for** j=1 to $|S|$ **do**
4: $C(P_i)_j \leftarrow$ the total cost of P_{i_j};
5: **end for**
6: $P \leftarrow find(P_i, min\{C(P_i)\})$;
7: $F \leftarrow F \cup P$;
8: **end for**
9: **return** multicast forest, F;

ρ_u=1. Constraint (13) indicates that $e_{u,v}$, $\tau_{u,v}$, $\phi_{d,s}$, $\pi_{d,(u,v)}$, ρ_u, and $\mathcal{P}_{u,v}$ are binary variables.

In the objective function (Formula 5), α=0 means that it is unnecessary to consider the recovery cost of the ReMUS problem. Furthermore, the ReMUS problem becomes the multicast problem with uncertain sources when α=0. Note that the Steiner minimum tree problem of a traditional multicast with one source is NP-hard in graph theory [13]. The multicast problem with uncertain sources is more difficult than that of a single source, due to the flexible usage of uncertain sources. It has been shown that the multicast problem with uncertain sources is also NP-hard [7]. Therefore, when $\alpha \neq 0$, the ReMUS problem is more challenging.

3 Efficient solutions of ReMUS

The settings and usage of uncertain sources and recovery nodes jointly dominate the performance of the ReMUS problem. In the follows, we first design a source-based multicast method to solve the problem by exploiting the flexibility of sources, when no recovery node exists in the network. Furthermore, we design a general multicast method to jointly exploit the benefits of uncertain sources and recovery nodes when recovery nodes have been deployed in networks.

3.1 Source-based building method of multicast forest

Here, we consider a multicast transfer with uncertain sources in the network, where there are no recovery nodes. Accordingly, we propose a source-based method to find the routes of a multicast forest and adopt the source-based recovery model in Section 2.3.1 to solve the ReMUS problem, which is called SR-based multicast method. Meanwhile, it is worth noting that multiple sources can be exploited to reduce the transfer cost as well as the recovery cost. According to the previous analysis, it is not necessary to employ all sources for the ReMUS problem. Thus, it is very important to pick sources from all potential sources. Those employed sources will affect the performance of multicast forest. Meanwhile, different sources will incur a different total cost of the ReMUS. We need to select the sources, which incur the minimal total cost of ReMUS.

To solve the ReMUS problem, a simple method is to find the nearest source for each destination. However, the nearest source to the destination is not sure to incur the least transfer cost when each link has a different transfer cost. Moreover, the source that can incur the least transfer cost does not mean to bring the minimal recovery cost because each link has a different probability of packet loss. Given a network, when modeled as an undirected graph, we set the transfer cost of each link as its weight. In the weighted graph, the shortest path between nodes u and v incurs the lowest transfer cost while delivering content along the selected path.

However, for the source-based recovery model, the recovery cost of a destination is equal to the transfer cost of the path times the probability of packet loss on the path. Although the shortest path can ensure the lowest transfer cost, the associated recovery cost is not always the lowest since the probability of packet loss may be non-minimal. Even though we suppose each link has the same probability of packet loss to simplify the problem, The weighted shortest path still does not mean the minimal recovery cost. The recovery cost of a selected path is related to the number of links, the transfer cost of each link and the probability of packet loss in these links.

To construct the multicast forest, we design an efficient strategy to find a proper source for each destination. For each destination, we first compute the shortest paths from the destination to all sources. Second, for each of the shortest paths, we calculate the total cost, including the transfer cost and the recovery cost. Third, we select the shortest path with the lowest total cost for each destination. Inspired by the above analysis, we design the SR-based method to build the desired multicast forest. Algorithm 1 reports the details of the SR-based method. The basic insight is to select the optimal path for each destination using the above strategy. The multicast forest F can be derived by combing those resultant paths of all destinations. When a destination meets packet loss, the related source will retransfer the lost packets to their destinations based on the recovery model in Section 2.3.1. Therefore, the SR-based method can achieve a reliable and bandwidth-efficient multicast transfer.

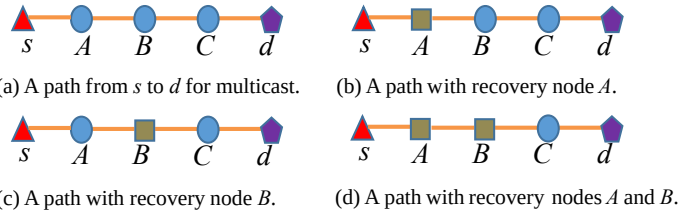
Additionally, when the transfer costs of all links are the same, the shortest path between two nodes can be simplified to find the path with the least number of links. Furthermore, if all links face the same probability of packet loss, the shortest path also incurs the minimal recovery cost at the same time. In this case, Algorithm 1 will still be efficient and find the shortest path for each destination.

Time Complexity. The time complexity of calculating the shortest paths from a node to all other nodes in the network is $O(|V|^2)$. Thus, in Algorithm 1, the time complexity of Step 2 is $O(|V|^2)$. The time complexity of the loop in Step 3 is $O(|S| \times |V|)$. The time complexity of Step 6 is $O(|V|)$. Then, the time complexity of the loop in Step 1 is $O(|D| \times |V|^2)$. Therefore, the entire time complexity of Algorithm 1 is $O(|D| \times |V|^2)$.

3.2 A general building method of multicast forest

When there are recovery nodes in the networks, we can adopt the recovery model in Section 2.3.2. Assume that recovery nodes are deployed in advance and their locations are fixed. The general building method of the multicast forest needs to consider the locations of recovery nodes and sources. Meanwhile, it also takes the shared links between multiple destinations into account.

Impact of recovery nodes. Our observations show that the quantity and locations of recovery nodes can affect the



(a) A path from s to d for multicast. (b) A path with recovery node A.

(c) A path with recovery node B. (d) A path with recovery nodes A and B.

Fig. 2. Routing paths from s to d under different settings of recovery nodes.

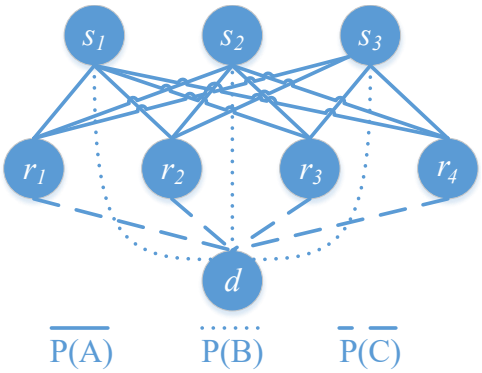


Fig. 3. Shortest paths from all sources to a destination.

recovery cost of the ReMUS. Assume that the transfer cost of each link is 1 and the packet loss probability of each link is 0.1 in Fig. 2. We then compare the recovery cost of different multicast paths.

When a destination fails to receive some packets, it will bring the recovery cost during the recovery process. In Fig. 2(a), source s will retransfer lost packets to destination d since no recovery nodes exist on the path. Therefore, the recovery cost of destination d is $4\times(1-0.9^4)$=1.3756 under the source-based recovery model. In Fig. 2(b), destination d has two recovery proxies, source s and recovery node A, to retransfer lost packets. When packet loss occurs in the path from s to A, it is essential to retransfer lost packets to d from source s. However, recovery node A can be exploited to retransfer lost packets to d when packet loss occurs in the path from A to d. Based on Formula (4), $c(\mathcal{P}_{s,d})$=4, $\eta_{d,s}$=0.1, $c(\mathcal{P}_{A,d})$=3, $\eta_{d,A}=1-0.9^3$. Thus, the recovery cost of destination d is $4\times0.1+3\times0.9\times(1-0.9^3)$=1.1317.

By comparing the recovery cost of destination d in Fig. 2(a) and Fig. 2(b), we can see that the new recovery node can be utilized to reduce the recovery cost. Similarly, the recovery cost of destination d in Fig. 2(c) and Fig. 2(d) are 1.0678 and 0.9778, respectively. By comparing Fig. 2(b) and Fig. 2(c), we can see that the locations of recovery nodes also affect the recovery cost of destination d where there are two recovery nodes in both Fig. 2(b) and Fig. 2(c). The recovery cost of ReMUS consists of the recovery cost of all destinations. Therefore, the locations of recovery nodes can indirectly affect the recovery cost of ReMUS. Furthermore, more recovery nodes lead to less recovery cost, comparing Fig. 2(d) with Fig. 2(b) and Fig. 2(c).

The design of Algorithm. The building process of a multicast forest needs to consider the recovery nodes, thus the method is also called the RN-based multicast method.

As shown in Algorithm 2, F records the multicast forest derived by the algorithm and is empty initially. The RN-based method firstly finds those shortest paths, $P(A)$, from each recovery node to all sources. Secondly, for each destination d, it needs to find the shortest paths, $P(B)$, from d to all sources. Then, the RN-based method will compute all the shortest paths, $P(C)$, from d to all candidate recovery nodes. For destination d, Fig. 3 shows the sets of paths $P(A)$, $P(B)$, $P(C)$. It is worth noting that adding the shortest paths $P(A)$ and $P(C)$ into the multicast forest F independently will result in that some destinations fail to receive the data. It is because that, when a shortest path in $P(C)$ from a destination to a recovery node is added into the multicast forest, the recovery node could not be connected to any one source. Therefore, to ensure that each destination can receive the data from a source, it is necessary to combine $P(A)$ and $P(C)$ as another set of shortest paths, $P(A+C)$. For a destination, each shortest path in $P(A+C)$ contains at least one recovery node.

Algorithm 2 RN-based building method of multicast forest.

Require: $G(V,E)$, source set S, destination set D, recovery set C, probability of packet loss β, and α.
Ensure: multicast forest, F.

```
for i=1 to |C| do
    calculate shortest paths P(A) from r_i to S;
end for
for i=1 to |D| do
    calculate shortest paths P(B) from d_i to S;
    for j=1 to |S| do
        C(B)_j ← addedCost(P(B)_j, C, F_2, α, β);
    end for
    calculate shortest paths P(C) from d_i to C;
    for k=1 to |C| do
        P(A+C)_k ← P(A)_k ∪ P(C)_k;
        C(A+C)_k ← addedCost(P(A+C)_k, C, F_2, α, β);
    end for
    P(D) ← find(P(B)∪P(A+C), min{C(B)∪C(A+C)});
    F ← F ∪ P(D);
end for
return multicast forest, F;
```

The main idea of Algorithm 2 is to select the path for each destination from $P(B)$ and $P(A+C)$. All selected paths between destinations and sources constitute the multicast forest F where Algorithm 2 adds the selected paths into F one by one. It is worth noting that there are some common links among different paths for those destinations. That is, when selecting a path for a destination, the candidate paths could have some shared links with F. For example, when adding a path to destination d into F, the total cost of multicast forest F increases, and that will produce the added total cost. More precisely, the total cost of a path is the cost of all links in a newly selected path, and the added total cost of that path is the cost of those links in the path not including the links that have been in F. Therefore, the selection strategy of Algorithm 2 is to pick the path that has the minimum added total cost instead of the path that has the minimum total cost for each destination. Then, Algorithm 2 adds the selected paths for destinations into the multicast forest F. For the paths in $P(B)$, it is necessary to calculate the added total cost by calling the function $addedCost()$ shown in Algorithm 3, while adding the shortest path into F. Similarly, for each path in $P(A+C)$, we will compute the added total cost. The path that has the lowest added total cost will be selected. Finally, Algorithm

Algorithm 3 $addedCost(P, C, F, \alpha, \beta)$.

Require: path P, recovery set C, multicast links F and probability of packet loss β.
Ensure: added total cost, $Acost$
1: $links \leftarrow \{F \cup P\} - F$;
2: $Tcost$=$c(links)$;
3: $temp$=$intersect(P, C)$;
4: **if** isempty(temp) **then**
5: $Rcost$= the recovery cost of path P;
6: **else**
7: find the locations of recovery nodes in the path P;
8: $temp$=$temp \cup s$
9: **for** i=1 to $|temp|$ **do**
10: $Rcost \leftarrow Rcost$+the recovery cost of recovery path from $temp_i$ to d;
11: **end for**
12: **end if**
13: $Acost$=$Tcost + \alpha \times Rcost$;
14: **return** added total cost, $Acost$;

2 returns the multicast forest F.

Algorithm 3 plots the details about how to compute the added total cost while adding path P into the multicast forest F. P denotes a path from source s to destination d. The added total cost $Acost$ consists of the added transfer cost $Tcost$ and the added recovery cost $Rcost$. The function $addedCost()$ firstly derives the added links, which are in P but not in F, and then computes the transfer cost of such links, which equals to $Tcost$. Secondly, the function identifies the recovery nodes in path P. If there are no recovery nodes in path P, it is essential to adopt the source-based recovery model described in Section 2.3.1. Thirdly, we need to get the locations of recovery nodes in path P when there exist recovery nodes. Fourthly, Algorithm 3 calculates the recovery cost $Rcost$ of destination d based on Equation (4). Finally, after achieving $Tcost$ and $Rcost$, we can get the added total cost $Acost$ of path P .

After building a desired multicast forest using Algorithm 2, all employed sources and recovery nodes will be utilized to realize the reliable multicast transfer. When a destination fails to receive data, the nearest source or recovery node will retransfer lost packets. If the nearest recovery proxy has also not received those packets, it tries to receive packets from a previous recovery proxy. In other words, when a destination fails to receive some packets, the upstream nearest recovery proxy caching those lost packets will retransfer those packets to the destination.

Time Complexity. It is well known that Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph. Given a node, the time complexity for finding the shortest path between two nodes is the same as the time complexity for finding the shortest paths from the node to all the other nodes and is $O(|V|^2)$ where V is the number of nodes in the network. Therefore, the time complexity of the first loop in Algorithm 2 is $O(|C| \times |V|^2)$. The time complexity of function $addedCost(P, C, F, \alpha, \beta)$ is $O(|V|)$, and then, the time complexity of steps 6 and 10 are $O(|S| \times |V|)$ and $O(|C| \times |V|)$, respectively. The time complexity of steps 5 and 9 is $O(IVI^2)$. Therefore, the time complexity of the second loop in line 4 is $O(|D| \times |V|^2)$. In summary, the time complexity of Algorithm 2 is $O((|C| + |D|) \times |V|^2)$.

3.3 Discussion

Some actual problems need to be carefully designed after two building methods of the multicast forest are proposed. In this section, we concentrately discuss the network dynamics, the packet loss detection and the packet loss ratio. Uncertain sources bring new challenges and opportunities to those problems.

Network dynamics. The multicast forest is constructed based on the network state at that time. After that, when the network performance and loads change, it is not required to reconstruct the multicast forest, but could cause the retransmission of packets. Meanwhile, the change of the recovery nodes will not result in the reconstruction of the multicast forest. However, when the sources and the destinations change, the multicast forest will also change. Recall that the multicast forest includes multiple multicast trees. The advantage of ReMUS is that the change of sources and destinations in a multicast tree will not affect other multicast trees. For example, when an employed source node departs, the control plane just needs to find a new source for the affected destinations. The join of a new source has no influence on the multicast forest. Furthermore, when a new destination joins the network, the control plane inserts the new destination to an appropriate position in one multicast tree that is nearest to the new destination. The new destination starts to get data from the multicast tree and the buffer of its recovery nodes. Note that ReMUS does not guarantee that the newly joined destination can receive the data from the first byte of the multicast session before its join. Moreover, the new destination will not have an influence on the destinations in other multicast trees of the multicast forest. When a destination leaves the multicast session, only the multicast tree with the leaving destination will be updated.

Packet loss detection. Each packet is assigned with a sequence number. When a receiver fails to receive a sequenced packet, the receiver forwards a negative acknowledgment (NAK) message [14] according to the following way. First, when a receiver detects the packet loss, it sends a NAK to the upstream recovery node. If the recovery node has received the lost packet, it will retransmit the lost packet to the receiver. Otherwise, the recovery node will continue to forward the NAK to its upstream recovery node. When there exist no recovery nodes, the NAK will be directly forwarded to the source node. The recovery nodes are closer to the destination than the source. Therefore, the recovery nodes save the bandwidth not only for the loss packet retransmission but also for the NAK messages.

Packet loss ratio. The multicast forest of ReMUS is closely related to the packet loss ratio. The source can exploit the links with less packet loss ratio to transfer data. Meanwhile, the source neighboring the links with less packet loss achieves also a greater possibility to be employed for the ReMUS. Based on the packet loss ratio recorded by the historical statistics, the parameter β can be further adjusted in Section 2.3. Meanwhile, the retransmitted packets may also suffer from the packet loss, which also can be recorded to improve the multicast forest. Furthermore, when the packet loss ratio of some employed links goes up to a certain threshold or the packet loss is incurred by the link failures,

it is essential to adjust the employed links.

In addition, the control plane has the global network view and decides the routes for flows in SDN [15][16]. Based on this design, the control plane can conquer the utilization of links and the packet loss ratio, which occurs in links. It is feasible to record the occurrence of the packet loss, and then get the packet loss ratio in SDN. The link failures can be detected by the control plane. The adjusted multicast forest can be achieved through that the control plane inserts or deletes some routing rules. The adjusting objectives are to ensure all destinations can receive data and minimize the adjusting to the existing multicast forest.

4 Performance Evaluation

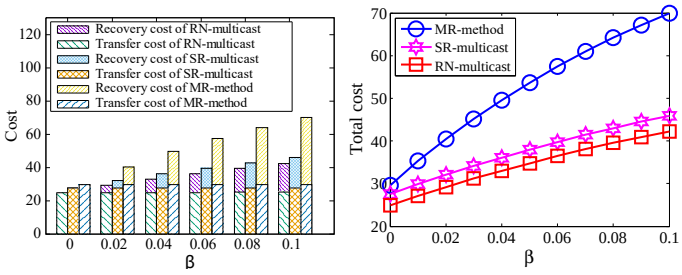
In this section, we first introduce the experimental setting. Then, we conduct massive experiments to evaluate the performance of our SR-based and RN-based methods comparing with the modified RAERA method [6], under varied settings of networks and multicast transfers.

4.1 Experiment setting

The setting of networks. We simulate our algorithm in the real backbone network of Internet2 [17]. Moreover, we also evaluate our algorithm in large representative Fat-tree networks of production data centers [18] with thousands of nodes and links to test the scalability of the proposed algorithm. Based on the statistic data [19], [20], we set the packet loss rate of each link from 1% to 10%. The recovery processes may also suffer from packet loss. The sources, destinations, and candidate recovery nodes are chosen randomly from each network. To eliminate the influence of this random selection, each simulation result is averaged over 100 samples.

Compared methods. We compare three different reliable multicast methods including our SR-based method, RN-based method and the modified RAERA method (MR-method). RAEAR [6] is proposed to find the reliable multicast routing for the multicast with only one source in software-defined networks. RAEAR firstly builds a tree structure, which is composed of all shortest paths from one source to all destinations. Then, it iteratively re-routes some involved destinations on the resultant tree to reduce the tree cost. Meanwhile, it is essential to ensure that there is at least one recovery node on the path from the source to each destination during the re-routing process [6]. Under uncertain sources, we further modified the RAEAR method, called MR-method. The MR-method is to build multicast trees from each source to all destinations using the RAERA method [6], and then selects the multicast tree with the minimal total cost for a multicast transfer.

Performance metrics. The evaluation metric is the total cost of ReMUS, which consists of the transfer cost and the recovery cost. For any setting of ReMUS, the three methods would generate different multicast forests. Accordingly, we can calculate the transfer cost, the recovery cost and the total cost of such resultant multicast forests. The total cost of a multicast forest can be calculated by Formula (5) and is composed of the transfer cost and the product of α and the recovery cost. When α=1, the total cost is the sum of the transfer cost and the recovery cost. Without loss of generality, we suppose that the transfer cost of each link is 1. However, our models and algorithms can apply to the situation where each link has a different transfer cost.



(a) The recovery and transfer cost of the three methods. (b) The changing trends of total cost.

Fig. 4. The impact of β on the performance of the three methods under the Internet2 topology.

4.2 Experiments on Internet2

Assuming α=1, which means recovery cost obtains the same weight as transfer cost, and the total cost is equal to the sum of transfer cost and recovery cost. Unless otherwise specified, we set the number of sources, destinations and recovery nodes to 3, 10 and 10, respectively. Additionally, the probability of packet loss β is an important parameter because it indicates the expectation of recovery for a destination. Without losing generality, we suppose that the probability of packet loss β in each link is the same.

4.2.1 *Impact of the probability of packet loss β*

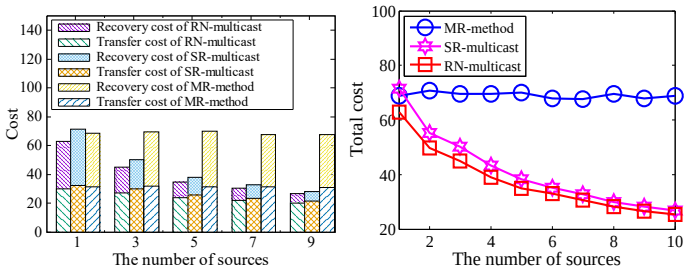
To evaluate the impact of link failure possibility β on the performance of three methods, we conduct experiments, while varying β from 0 to 0.1 with the interval of 0.01.

Fig. 4 shows the impact of β on the transfer cost, recovery cost and total cost of the three methods given that α=1. In Fig. 4(a), the recovery cost increases with the increase of β. Comparing these costs of our RN-based method, SR-based method and the MR-method, it's remarkable to see the increasing trend of the total cost when β increases in Fig. 4(b). However, the change in transfer cost is modest. When β=0, the recovery cost of the three methods is 0 because there are no lost packets. Moreover, we can see that the transfer cost of the multicast forest derived by our RN-based method is the lowest compared with the SR-based method and the MR-method. Our RN-based method adopts more sources than the MR-method and better selecting strategy than SR-based method.

Comparing the recovery costs of the three methods in Fig. 4(a), our RN-based and SR-based algorithms are more effective than the MR-method without the change of β. Fig. 4(b) also shows that our RN-based method gets less total cost, owing to its smaller transfer cost compared to the SR-based method. In conclusion, the recovery cost increases with the increase of β, and our RN-based method achieves the lowest total cost when β changes. Without losing generality, we will set α=1 and β=0.1 in the next evaluations.

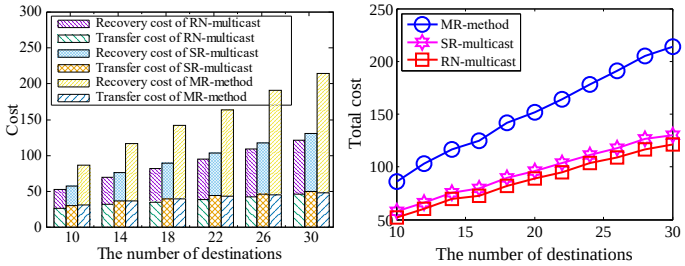
4.2.2 *Impact of the number of sources*

In this section, we change the number of sources from 1 to 10 with the interval of 1. Without changing other parameters,



(a) The recovery and transfer cost of the three methods. (b) The changing trends of total cost.

Fig. 5. The impact of the number of sources on the performance of the three methods under the Internet2 topology.



(a) The recovery and transfer cost of the three methods. (b) The changing trends of total cost.

Fig. 6. The impact of the number of destinations on the performance of the three methods under the Internet2 topology.

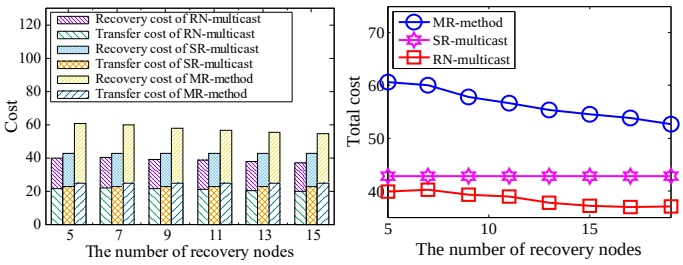
Fig. 5 shows that the impacts of increasing number of sources on both the recovery cost and the total cost are significant for our SR-based and RN-based methods.

The increasing number of sources has a slight influence on the performance of the MR-method because it only employs one source for multicast transfer. When there is only one source, the MR-method achieves a lower transfer cost than the SR-based method in Fig. 5(a). However, as the number of sources increases, the performance of our SR-based and RN-based methods improve rapidly. Furthermore, Fig. 5(b) shows that our RN-based method achieves the lowest total cost because it considers the locations of the recovery nodes and effectively reduces the recovery cost of ReMUS. For our SR-based and RN-based methods, their recovery costs reduce rapidly to a quarter of the transfer costs (respectively) when sources increase to 10 in Fig. 5(a). After that, the number of sources increases to 8 and the total costs of our RN-based and SR-based methods slowly decrease in Fig. 5(b). The facts also reflect that it is not necessary to use all sources in the multicast transfer.

4.2.3 Impact of the number of destinations

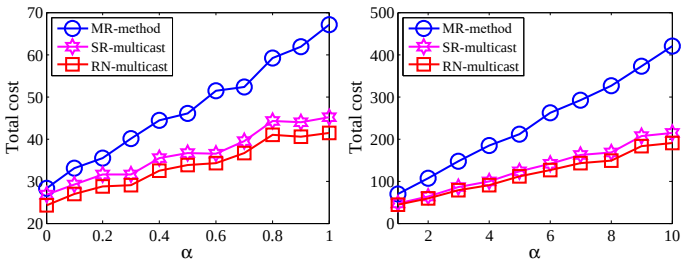
Without changing other parameters, we test the impact of the number of destinations on the performance of three methods where the number of destinations increases from 10 to 30 with the interval of 2.

Fig. 6 plots that the transfer cost, recovery cost and total cost of the three methods increase with the increase of the number of destinations. Fig. 6(a) shows that the recovery cost is significantly influenced by the quantity of destinations, especially for the MR-method. The recovery cost of the MR-method intensively changes when the number of destinations increases. Meanwhile, MR-based has always the highest total cost in Fig. 6(b). The changing trend of



(a) The recovery and transfer cost of the three methods. (b) The changing trends of total cost.

Fig. 7. The impact of the number of recovery nodes on the performance of the three methods under the Internet2 topology.



(a) α changes from 0 to 1. (b) α changes from 1 to 10.

Fig. 8. The impact of the variable α on the total cost of the three methods.

the total cost for the RN-based method is similar to the trend of the SR-based method. The transfer costs of SR-based method and RN-based method are almost the same and slowly increase in Fig. 6(a). Fig. 6 shows that our RN-based method works effectively because it achieves the minimal total cost. Additionally, the MR-method has a modest performance because it only uses one source and limits that there must be at least one recovery node in paths from the source to the destinations.

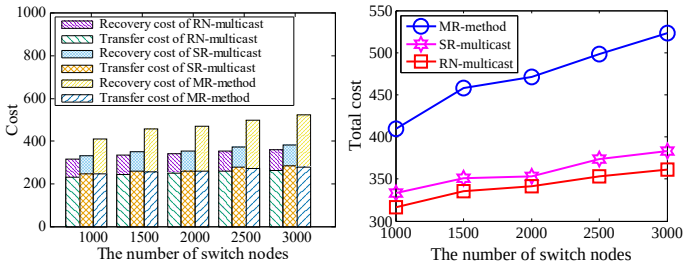
4.2.4 Impact of the number of recovery nodes

We evaluate the impact of recovery nodes on the performance of three methods where the number of recovery nodes increases from 5 to 20 with the interval of 2.

As shown in Fig. 7(a), the cost under the SR-based method remains steady when the number of recovery nodes increases because it applies to the sources-based recovery and has nothing to do with those recovery nodes. However, the MR-method and our RN-based method utilize those recovery nodes to reduce the multicast cost, hence their costs decrease when the number of recovery nodes increases. The decreasing trend for the RN-based method is not very obvious, it is because that the cost for our RN-based method is low. Meanwhile, we can see that our RN-based method achieves the lowest transfer cost, recovery cost and total cost from Fig. 7(a). Furthermore, Fig. 7(b) shows also that our RN-based method achieves the minimal total cost, and its total cost continues to decrease when the number of recovery nodes increases.

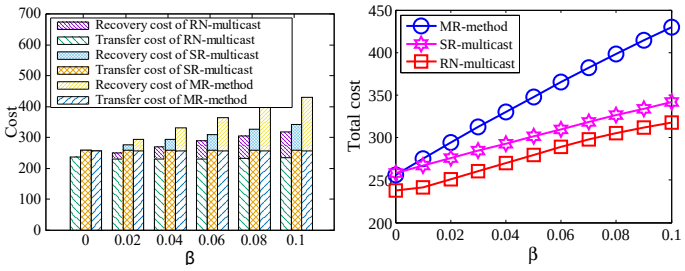
4.2.5 Impact of variable α

We evaluate the impact of the variable α on the total cost of the three methods where α changes from 0 to 1 with the interval 0.1 and from 1 to 10 with the interval 1. Fig. 8 shows that the total cost of three methods all go up



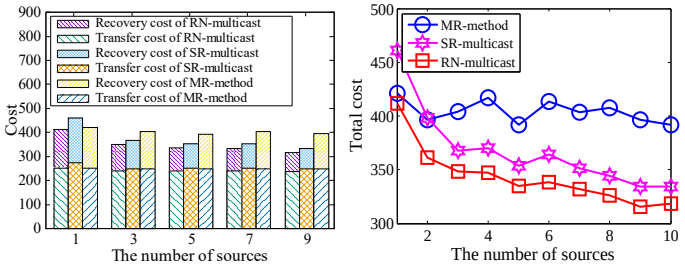
(a) The recovery and transfer cost of the three methods. (b) The changing trends of total cost.

Fig. 9. The impact of the network size on the performance of the three methods under datacenter networks.



(a) The recovery and transfer cost of the three methods. (b) The changing trends of total cost.

Fig. 10. The impact of β on the performance of the three methods under datacenter networks.



(a) The recovery and transfer cost of the three methods. (b) The changing trends of total cost.

Fig. 11. The impact of the number of sources on the performance of the three methods under datacenter networks.

with the increase of α. However, our RN-based method always incurs the lowest total cost among the three methods without the value of α. Fig. 8 shows that our SR-based method achieves a lower total cost than the MR-method. Fig. 8 also reveals that our RN-based method has a better performance than the SR-based method.

In conclusion, we evaluate three reliable multicast methods under Internet2. Our RN-based method obtains the lowest recovery-costly multicast forest, and the multicast derived by our RN-based method always incurs the lowest total cost. Although Internet2 is a real network topology, its scale is not enough large. Thus, we further evaluate the three methods under large-scale networks with the Fat-tree topology [18] in the next section.

4.3 Experiments on datacenter networks

It is well known that data centers are infrastructures of cloud computing and big data. To evaluate the performance of our RN-based and SR-based methods in data centers, we select the typical topology of data centers, the Fat-tree architecture. Unless otherwise specified, we set the number of sources, destinations, recovery nodes and switches in the network to 10, 200, 600 and 1000, respectively. Given α=1 and β=0.1, we randomly set the locations of sources, destinations and recovery nodes in each round of the experiments.

4.3.1 Impact of the network size

We evaluate the impact of the network size on the performance of the three methods. The network size is reflected by the number of switches in the network, where the number of switches increases from 1000 to 3000. Fig. 9 shows that the transfer cost, recovery cost and total cost increases as the network size expands. Especially, the recovery cost of MR-method increases fast and evidently in Fig. 9(a). the recovery cost of the MR-method is the highest. However, Fig. 9(a) reveals that our RN-based method achieves the lowest recovery cost. The transfer costs of the three methods have a small increase in Fig. 9(a). Meanwhile, Fig. 9(a) shows that our RN-based method achieves the lowest transfer cost. Hence, the total cost of our RN-based method is lower than that of SR-based method and the MR-method in Fig. 9(b).

4.3.2 Impact of the probability of packet loss β

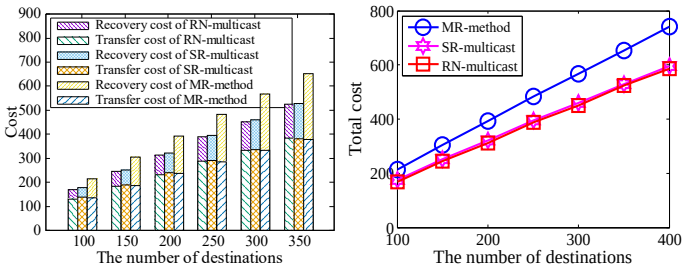
Under the previous settings, there are 1000 switches in the network with Fat-tree topology. We evaluated the impact of β on the recovery cost, transfer cost and total cost of the three methods where the probability of packet loss β increased from 0 to 0.1. Fig. 10(a) shows that our solutions are always better than MR-method in terms of recovery cost. Moreover, the recovery cost of MR-method is not only the highest but also the fastest-increasing among all recovery costs. The impact of β on the transfer costs of three methods is modest in Fig. 10(a). Meanwhile, the recovery costs of three methods increase when the probability of the packet loss β increases. The lowest total cost is achieved by our RN-based method without the value of β in Fig. 10(b).

4.3.3 Impact of the number of sources

We evaluated the impact of the number of sources on the recovery cost, transfer cost and total cost of the three methods where the number of sources increases from 1 to 10. The recovery cost of the SR-based method decreases rapidly as the number of sources increases in Fig. 11(a). Fig. 11(a) also shows that these costs of the MR-method almost keep stable with the increasing of the number of sources because it's used for single-source multicast. Fig. 11(a) shows that the MR-method achieves better performance than our SR-based method when there is only one source. However, with the increasing of the number of sources, our SR-based method has a significant improvement. Meanwhile, the performance of our RN-based method is always the best and incurs the lowest recovery cost and transfer cost in Fig. 11(a). Fig. 11(b) reveals that our RN-based method achieves the lowest total cost among the three methods without the number of sources.

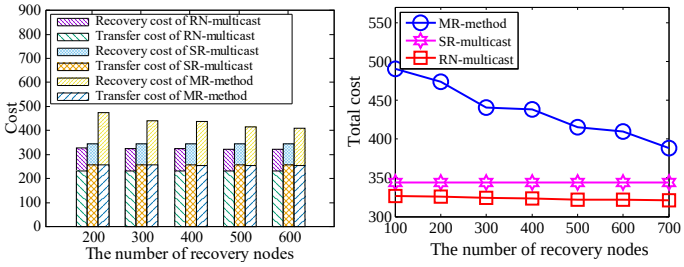
4.3.4 Impact of the number of destinations

We compare the recovery cost, transfer cost and total cost only varying the number of destinations from 100 to 400. We set the number of recovery nodes to 550 rather than 600 in order to ensure the total number of sources, destinations and recovery nodes is not larger than that of switches.



(a) The recovery and transfer cost of the three methods. (b) The changing trends of total cost.

Fig. 12. The impact of the number of destinations on the performance of the three methods under datacenter networks.



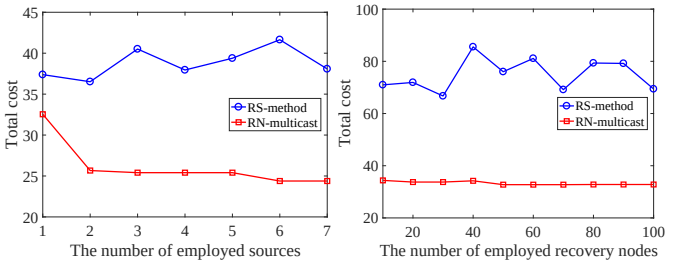
(a) The recovery and transfer cost of the three methods. (b) The changing trends of total cost.

Fig. 13. The impact of the number of recovery nodes on the performance of the three methods under datacenter networks.

Fig. 12 shows that the increase of the number of destinations causes the increase of these costs of the three methods. Fig. 12(a) reveals that the recovery costs are heavily influenced by the quantity of destinations and have faster increases than the transfer costs. Meanwhile, our RN-based method incurs the lowest recovery cost which is almost half of that of the MR-method in Fig. 12(a). Additionally, the transfer costs of the three methods are similar and are far larger than their recovery costs in Fig. 12(a). Moreover, Fig. 12(b) reveals that our RN-based and SR-based methods always achieve lower total cost than the MR-method. The transfer cost has a larger influence on the total cost than the recovery cost. In result, the advantage of the RN-based method is modest over the SR-based method in Fig. 12(b).

4.3.5 Impact of the number of recovery nodes

Without changing other parameters, we increase the number of recovery nodes from 100 to 700. We evaluate the impact of the number of recovery nodes on the recovery cost, transfer cost and total cost for the three methods. Fig. 13 shows that the recovery cost and the total cost of both the MR-method and the RN-based method decreases when the number of recovery nodes increases. Notably, the recovery cost of the MR-method efficiently decreases because of the increasing recovery nodes. However, it still has the highest recovery cost, and the recovery costs of our two methods are still lower than that of the MR-method in Fig. 13(a). Fig. 13(a) also reveals that the transfer cost and the recovery cost of the SR-based method are stable because its routing is independent with the recovery nodes. Thus, the decreasing trend for the RN-based method can also be observed by the difference of the results between the SR-based method and the RN-based method. In Fig. 13(a), the difference is more obvious when the number of recovery nodes increases. Note that the decreasing trend for the RN-based method is not obvious in Fig. 13(b), it is because that the cost for the RN-based method is low. More importantly, our RN-based method achieves not only the lowest transfer cost but also the lowest recovery cost.



(a) The impact of the locations of sources. (b) The impact of the locations of recovery nodes.

Fig. 14. The impact of the locations of sources and recovery nodes on the total cost of the multicast transfer under datacenter networks.

4.3.6 Impact of the locations of sources and recovery nodes

In this section, we evaluate the impact of the locations of sources and recovery nodes on the performance of the multicast. Given the number of employed sources and recovery nodes, we compare our RN-based method with the random select method (RS-method), which randomly select the sources and recovery nodes for each destination. Fig. 14(a) shows that our RN-based method achieves significantly less total cost for the multicast transfer than the RS-method. In this case, given the number of employed sources, different sources are employed under different multicast methods, which incur a different multicast cost. Meanwhile, we can see that more sources are not always to reduce the total cost for the RS-method from Fig. 14(a). It is because that more sources could increase more links for the multicast transfer. Furthermore, given the number of the employed recovery nodes, Fig. 14(b) indicates that our RN-based method achieves also less total cost than the RS-method. In summary, the locations of sources and recovery nodes have an influence on the performance of multicast.

5 Related Work

Traditional multicast has attracted a large amount of research, which can be roughly divided into two categories. The first one focuses on reducing the consumption of network bandwidth. Many multicast services prefer to deliver the same content to a group of destinations along a shortest-path tree, such as PIM-SM [3]. Such methods, however, are not bandwidth-efficient since each of those shortest paths is calculated independently. The Steiner minimum tree (SMT) [21] is more promising due to the minimization of the number of occupied links for a multicast group. Many approximation algorithms [22], [23], [24] have been proposed to solve the SMT problem, which is NP-hard. There are also overlay Steiner trees [25], for P2P environments. However, the design of SMT does not consider the selection of recovery nodes. Moreover, it is just exploited for one source and is not suitable for uncertain sources. Hu et al. proposed a new multicast scheme with uncertain sources for SDN, called uncertain multicast [7]. However, they did not consider the reliability problem in their work.

The second one aims to ensure the reliable transmission of multicast. Nowadays, the reliable multicast becomes crucial to provide reliable services for many important Internet and datacenter applications [4]. There are some source-based reliable multicast methods [5], which suffer from the scalability problem since only one source node serves the recovery requests from all destinations. A reliable multicast can also be established at the application level, or by using router-assist [26][27][28]. A hierarchical architecture with on-tree recovery nodes placed between the source and the destinations is proposed to facilitate local loss recovery [14][29]. However, the selection of recovery nodes has not been considered as an issue in those work. Recently, Shen et al. proposed the Recover-aware Steiner Tree (RST) problem [6]. It introduces at least one recovery node between the source and each destination to facilitate local loss recovery.

SDN brings new opportunities and challenges for multicast services. Popovic et al. [30] review existing graph-theoretical algorithms that were proposed to provide node-redundant multicast-distribution-trees for the smart-grid setting. Coronado et al. present SM-SDN@Play [31], an SDN-based solution for joint multicast rate selection and group formation in 802.11-based networks. Zhu et al. present a sender-initiated, efficient, congestion-aware and robust reliable multicast solution mainly for small groups in SDN-based data centers, called MCTCP [32]. Zhang et al. examines a video multicast orchestration scheme based on SDN, named HCM [33], for 5G networks. Those work just shows the advantage of SDN while considering the multicast with a single source. However, SDN brings more benefits for multicast with uncertain sources. When a multicast group is provided with multiple sources and recovery nodes, they can be jointly exploited to reduce the total cost of the ReMUS.

6 Conclusion

IaaS delivers Cloud Computing infrastructure to organizations, including things such as servers, network, operating systems, and storage, through virtualization technology. In Iaas, resources are available as a service. In this paper, we propose a novel reliable multicast service with uncertain sources, named ReMUS, which can efficiently reduce the network resources consumption. The uncertain sources and the recovery nodes jointly dominate the performance of the ReMUS. Although finding such a ReMUS is very challenging, we design two efficient multicast methods to solve this problem. Furthermore, we conduct extensive experiments to evaluate the performance of our methods. The results show the efficiency and effectiveness of our methods. This paper makes the first step to study the reliable services of multicast transfer under uncertain sources. We leave the ReMUS problem with the network dynamic and the latency guarantee as our future work.

Acknowledgments

This work is partially supported by the National Natural Science Foundation for Outstanding Excellent young scholars of China under Grant No.61422214, National Natural Science Foundation of China under Grant Nos.61772544 and U1536202, National Basic Research Program (973 program) under Grant No.2014CB347800, the Hunan Provincial Natural Science Fund for Distinguished Young Scholars under Grant No.2016JJ1002, and the Guangxi Cooperative Innovation Center of cloud computing and Big Data under Grant Nos.YD16507 and YD17X11.

References

[1] R. Malli, X. Zhang, and C. Qiao, "Benefit of multicasting in all-optical networks," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 3531, 2000.

[2] V. Aggarwal, R. Caldebank, V. Gopalakrishnan, R. Jana, K. K. Ramakrishnan, and F. Yu, "The effectiveness of intelligent scheduling for multicast video-on-demand," in *Proc. 17th ACM International Conference on Multimedia*, 2009.

[3] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei, "Protocol independent multicast-sparse mode (pim-sm): Protocol specification," *Draft*, 1998.

[4] D. Li, M. Xu, Y. Liu, X. Xie, Y. Cui, J. Wang, and G. Chen, "Reliable multicast in data center networks," *IEEE Transactions on Computers*, vol. 63, no. 8, pp. 2011–2024, 2014.

[5] S. Floyd, V. Jacobson, S. Mccanne, C. G. Liu, and L. Zhang, "A reliable multicast framework for light-weight sessions and application level framing," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 784 – 803, 1997.

[6] S. H. Shen, L. H. Huang, D. N. Yang, and W. T. Chen, "Reliable multicast routing for software-defined networks," in *Proc. IEEE INFOCOM*, 2015, pp. 181–189.

[7] Z. Hu, D. Guo, J. Xie, and B. Ren, "Multicast routing with uncertain sources in software-defined network," in *Proc. IEEE/ACM IWQoS*, Beijing, China, June 2016.

[8] J. Xie, D. Guo, Z. Hu, J. Wu, T. Chen, and H. Chen, "Reliable multicast routing with uncertain sources," in *Proc. 25th IEEE/ACM IWQoS*, June 2017, pp. 1–6.

[9] S. Ghemawat, H. Gobioff, and S. T. Leung, "The google file system," *Acm Sigops Operating Systems Review*, vol. 37, pp. 29–43, 2003.

[10] M. Alghamdi, "Hdfs: Hadoop distributed file system," *Wireless Communications & Mobile Computing*, 2014.

[11] B. G. Chun, P. Wu, H. Weatherspoon, and J. Kubiatowicz, "Chunkcast: An anycast service for large content distribution," in *Peer-to-Peer Systems*, 2006.

[12] A. A. Mahimkar, Z. Ge, A. Shaikh, J. Wang, J. Yates, Y. Zhang, and Q. Zhao, *Towards automated performance diagnosis in a large IPTV network*. ACM, 2009.

[13] R. M. Karp, "Reducibility among combinatorial problems. in: Complexity of computer computations," *New York: Plenum Press*, pp. 85–103, 1972.

[14] T. Speakman, J. Crowcroft, J. Gemmell, D. Farinacci, S. Lin, D. Leshchiner, M. Luby, T. Montgomery, L. Rizzo, and A. Tweedly, "Pgm reliable transport protocol specification," *Heise Zeitschriften Verlag*, 2001.

[15] J. Xie, D. Guo, Z. Hu, T. Qu, and P. Lv, "Control plane of software defined networks: A survey," *Computer Communications*, vol. 67, pp. 1 – 10, 2015.

[16] J. Xie, D. Guo, X. Li, Y. Shen, and X. Jiang, "Cutting long-tail latency of routing response in software defined networks," *IEEE JSAC*, vol. 36, no. 3, pp. 384–396, 2018.

[17] "Internet2 network infrastructure topology," https://www.internet2.edu/media/medialibrary/2014/10/23/I2-Network-Infrastructure-Topology-201410.pdf, October 2014.

[18] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," *Acm Sigcomm Computer Communication Review*, vol. 38, no. 4, pp. 63–74, 2008.

[19] H. X. Nguyen and M. Roughan, "Rigorous statistical analysis of internet loss measurements," *IEEE/ACM Transactions on Networking*, vol. 21, no. 3, pp. 734–745, June 2013.

[20] Y. Xu, C. Yu, J. Li, and Y. Liu, "Video telephony for end-consumers: Measurement study of google+, ichat, and skype," in *Proc. ACM IMC*, 2012, pp. 371–384.

[21] F. K. Hwang and D. S. Richards, "Steiner tree problem," *Networks*, vol. 22, pp. 55–89, 1992.

[22] L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for steiner trees," *Acta Informatica*, vol. 15, no. 2, pp. 141–145, 1981.

[23] M. Karpinski and A. Zelikovsky, "New approximation algorithms for the steiner tree problems," *J. Comb. Optim.*, vol. 1, no. 1, pp. 47–65, 1997.

[24] G. Robins and A. Zelikovsky, "Improved steiner tree approximation in graphs," in *Proc. of 7th Annual ACM-SIAM Symposium on Discrete Algorithms*, San Francisco, 2000, pp. 770–779.

[25] D. N. Yang and W. Liao, "On bandwidth-efficient overlay multicast," *IEEE Transactions on Parallel & Distributed Systems*, vol. 18, no. 11, pp. 1503–1515, 2007.

[26] P. Radoslavov, C. Papadopoulos, R. Govindan, and D. Estrin, "A comparison of application-level and router-assisted hierarchical schemes for reliable multicast," *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 469–482, 2004.

[27] J. C. Lin and S. Paul, "Rmtp: a reliable multicast transport protocol," in *Proc. IEEE Infocom*, 1996, pp. 1414–1424.

[28] C. Papadopoulos, G. Parulkar, and G. Varghese, "An error control scheme for large-scale multicast applications," in *Proc. IEEE Infocom*, 1998, pp. 1188 – 1196.

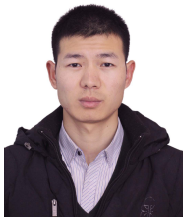
[29] B. Whetten and G. Taskale, "An overview of reliable multicast transport protocol ii," *IEEE Network*, vol. 14, no. 1, pp. 37–47, 2000.

[30] M. Popovic, R. Khalili, and J. Y. L. Boudec, "Performance comparison of node-redundant multicast distribution trees in sdn networks," in *3rd International Conference on Networked Systems (NetSys)*, March 2017, pp. 1–8.

[31] E. Coronado, R. Riggio, J. Villaln, and A. Garrido, "Efficient real-time content distribution for multiple multicast groups in sdn-based wlans," *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 430–443, 2018.

[32] T. Zhu, D. Feng, F. Wang, Y. Hua, Q. Shi, Y. Xie, and Y. Wan, "A congestion-aware and robust multicast protocol in sdn-based data center networks," *Journal of Network and Computer Applications*, vol. 95, pp. 105 – 117, 2017.

[33] X. Zhang, M. Yang, Y. Zhao, J. Zhang, and J. Ge, "An sdn-based video multicast orchestration scheme for 5g ultra-dense networks," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 77–83, 2017.

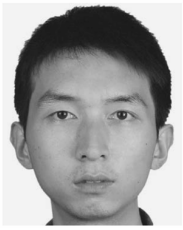


Junjie Xie Junjie Xie received the B.S. degree in computer science and technology from the Beijing Institute of Technology, Beijing, China, in 2013. He received the M.S. degree in management science and engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2015. He is currently a Ph.D. student in NUDT, from 2016. He is also a joint Ph.D. student in the University of California, Santa Cruz (UCSC), USA, from October 2017. His study in the UCSC is supported by the China Scholarship Council (CSC). His research interests include distributed systems, software-defined networking and edge computing.

Deke Guo Deke Guo received the B.S. degree in industry engineering from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 2001, and the Ph.D. degree in management science and engineering from the National University of Defense Technology, Changsha, China, in 2008. He is currently a Professor with the College of Information System and Management, National University of Defense Technology, and a Professor with the School of Computer Science and Technology, Tianjin University. His research interests include distributed systems, software-defined networking, data center networking, wireless and mobile systems, and interconnection networks. He is a senior member of the IEEE and a member of the ACM.

Jie Wu Jie Wu is the Associate Vice Provost for International Affairs at Temple University. He also serves as Director of the Center for Networked Computing and Laura H. Carnell professor in the Department of Computer and Information Sciences. Prior to joining Temple University, he was a program director at the National Science Foundation and was a distinguished professor at Florida Atlantic University. His current research interests include mobile computing and wireless networks, routing protocols, cloud and green computing, network trust and security, and social network applications. Dr. Wu regularly publishes in scholarly journals, conference proceedings, and books. He serves on several editorial boards, including IEEE Transactions on Service Computing and the Journal of Parallel and Distributed Computing. Dr. Wu was general cochair/chair for IEEE MASS 2006, IEEE IPDPS 2008, IEEE ICDCS 2013, and ACM MobiHoc 2014, as well as program co-chair for IEEE INFOCOM 2011 and CCF CNCC 2013. He was an IEEE Computer Society Distinguished Visitor, ACM Distinguished Speaker, and chair for the IEEE Technical Committee on Distributed Processing (TCDP). Dr. Wu is a CCF Distinguished Speaker and a Fellow of the IEEE. He is the recipient of the 2011 China Computer Federation (CCF) Overseas Outstanding Achievement Award.

Bangbang Ren Bangbang Ren received the B.S.degree and M.S.degree in management science and engineering from National University of Defense Technology,Changsha,China,in 2015 and 2017. He is currently a Ph.D. student in NUDT. His research interests include software-defined network,data center network and network function virtualization.



Tao Chen Tao Chen received the MS and PhD degrees in operational research from the National University of Defense Technology, Changsha, P.R. China, in 2006 and 2011, respectively. He is an assistant professor at the College of Information System and Management, National University of Defense Technology. His research interests include wireless sensor networks and data center networking. He is a member of the IEEE and the ACM.

Honghui Chen Honghui Chen received the MS degree in operational research and the PhD degree in management science and engineering from the National University of Defense Technology, Changsha, China, in 1994 and 2007, respectively. Currently, he is a professor of Information System and Management, National University of Defense Technology, Changsha, China. His research interests include information system, cloud computing and Information Retrieval.