Reliable Multicast Routing with Uncertain Sources

Junjie Xie*, Deke Guo*, Zhiyao Hu*, Jie Wu†, Tao Chen*, Honghui Chen*
*Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology, Changsha Hunan 410073, China
†Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA
Deke Guo is the corresponding author: dekeguo@nudt.edu.cn

***Abstract*—Multicast can jointly utilize the network resources when delivering the same content to a set of destinations; hence, it can effectively reduce the consumption of network resources than individual unicast. The source of a multicast, however, is not necessary to be in specific location as long as certain constraints are satisfied. This brings the multicast with uncertain sources, which could reduce more network bandwidth consumption than traditional multicast. Meanwhile, reliable multicast becomes crucial to provide reliable services for many important applications. However, prior minimal cost forest (MCF) for such a new multicast is not designed to support reliable transmissions. In this paper, we propose a novel reliable multicast routing with uncertain sources, named ReMUS. To the best of our knowledge, we are the first to study the reliable multicast under uncertain sources. The goal is to minimize the sum of the transfer cost and the recovery cost, while finding such a ReMUS is very challenging. Thus, we design a source-based multicast method to solve the problem by exploiting the flexibility of uncertain 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 to minimize the total cost of ReMUS. We conduct extensive evaluations based on the real topology of Internet2. The results indicate that our methods can efficiently realize the reliable and bandwidth-efficient multicast with uncertain sources, irrespective of the settings of networks and multicasts.**

I. Introduction

Multicast is an efficient method to deliver the same content from a given source to a group of destinations. It can considerably save the consumption of network resources 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 load of the source node and associated network links [2]. Recently, the appearance of the software defined networking (SDN) [3][4] offers opportunities to design and deploy flexible protocols, including variants of multicast methods.

A number of novel multicast methods are proposed recently and can be roughly divided into two categories. The first one focuses on reducing the consumption of network bandwidth. Many multicast routing methods prefer to deliver the same content to a group of destinations along a shortest-path tree, such as PIM-SM. The multicast tree can reduce more bandwidth consumption if those shortest paths from the same source to destinations share more links. 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 [5]. To achieve reliable transmission of multicast, the source-based reliable multicast prefers to recover the loss packets from the source directly. It, however, suffers from the scalability problem 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]. It introduces at least one recovery nodes 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], which brings new challenges and opportunities to the design of multicast methods. The root cause is the widely usage of the content replica strategy in various networks. For example, each file block in GFS and HDFS [8] has at least two replicas besides the original one across the datacenter. Furthermore, many content delivery applications have adopted the content replica design for improving the robustness and efficiency [9]. 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 is not necessary to be fixed in specific location as long as certain constraints are satisfied. This brings a new multicast with uncertain sources, which could reduce more consumptions of the network bandwidth than the traditional multicast methods.

In this paper, we reveal the reliable and bandwidth-efficient multicast with uncertain sources, abbreviated 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 transfer and recovery for the reliable multicast transfer. The ReMUS problem faces fundamental challenging issues. First, the bandwidth-efficiency and reliability are somehow conflicting with each other for a multicast transfer. The use of recovery nodes are effective to realize the reliable multicast, but directly change the design of 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 to reduce the recovery cost.

The major contributions of this paper are summarized as follows:

1) We firstly propose and characterize the reliable multicast routing with uncertain sources (ReMUS). We then rethink the source-based recovery model, and design the source-based multicast method to solve the ReMUS problem, when no recovery node exists in the network.
2) We design a general recovery model for the ReMUS problem and further propose the recovery node-based multicast method, which can jointly exploit recovery nodes and uncertain sources in networks.
3) We conduct extensive evaluations based on the real

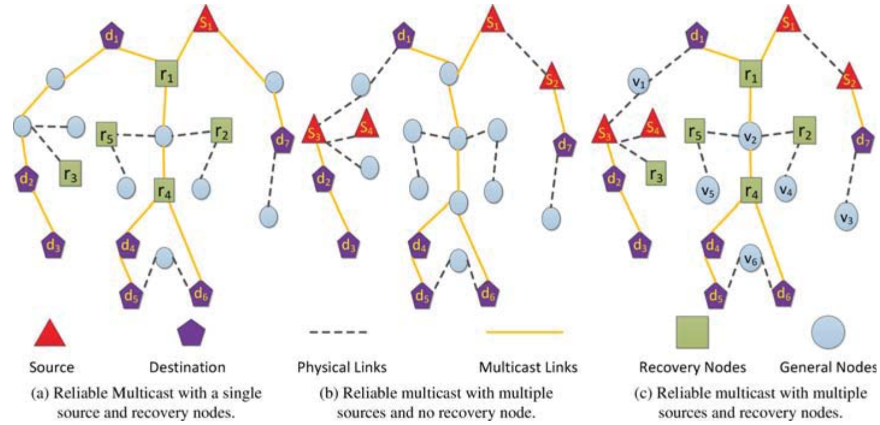


Fig. 1. Reliable multicast with the same destinations $\{d_1,d_2,d_3,d_4,d_5,d_6,d_7\}$ but different sources and recovery nodes.

topology of Internet2. The results indicate that our two methods can efficiently solve the ReMUS problem under the various settings of networks and multicasts.

II. Problem Description of ReMUS

In this section, we firstly describe the problem of ReMUS, and then give the recovery models of the ReMUS. At last, we formally characterize the ReMUS as a NLP problem and discuss its hardness.

A. Problem Statement

In this paper, we propose the problem of reliable multicast routing with uncertain sources (ReMUS). The ReMUS problem is to find a desired forest F, which employs the necessary sources and recovery nodes. A constraint is that each destination just reaches one source. Meanwhile, the sources and recovery nodes can be jointly exploited to achieve the reliable multicast routing. We firstly give the definition of recovery proxy in Definition 2.1. When there are lost packets, the recovery proxies in the multicast forest will retransfer the lost packets.

Definition 2.1 (Recovery proxy): Assume that there exits 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 can be 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 corresponding destination.

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(c). Meanwhile, Fig. 1(c) plots a desired multicast forest, which is composed of three multicast trees and 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(a) employs 13 links. This indicates that uncertain sources are very effective to reduce the transfer cost of traditional multicast. Moreover, the recovery proxy of destination d_6 changes as recovery node r_4 in Fig. 1(c) from S_1 in Fig. 1(b) after introducing recovery nodes. In this way, the recovery cost of destination d_6 is decreased since its recovery path is shortened. This indicates that recovery nodes have potential to reduce the recovery cost.

B. 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.

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. 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 as Equation (1), which is equal to the probability of packet loss on at least one link in the path.

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

In this case, the recovery cost of destination d is given by Equation (2) where $c(e_i)$ denotes the transfer cost of link e_i.

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

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

2) Designing a general recovery model: To characterize this general recovery model, let $R(d)$ denote the set of all recovery proxies of 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 recovering destination d from a recovery proxy $r(d)_i$, $1\leq i\leq|R(d)|$, which is calculated as Equation (3).

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

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 no packet loss in the path from s to $r(d)_i$. Further, $\beta_{s,r(d)_i}$ 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.

C. Problem Formulation

We use an undirected graph $G(V,E)$ to denote the network topology. F denotes the set of links employed by a desired forest for the ReMUS problem, i.e, a multicast forest. 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 receive data from 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 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 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 from u to v.

To calculate the recovery cost of multicast forest, κ_d denotes the recovery cost of any destination d in F, which is calculated as 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, it has multiple feasible recovery paths, resulting from those corresponding recovery proxies of destination d. The recovery cost of destination d is equal to the sum recovery cost from all its recovery proxies based on the recovery model in Section II-B2.

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

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

In Formula 5, α is a regulative parameter. 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. The above formulation is a Non-Linear Programming (NLP) model, because $c_{u,v}$ is not sure to be 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 constrains should be satisfied.

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

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

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

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

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

Algorithm 1 SR-based building method of multicast forest.

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

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

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

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

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 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, ρ_u=1. Constraint (13) indicates that $e_{u,v}, \tau_{u,v}, \phi_{d,s}, \pi_{d,(u,v)}, \rho_u, \mathcal{P}_{u,v}$ 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 traditional multicast with one source is NP-hard in graph theory [10]. The multicast problem with uncertain sources is more difficult than that with 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.

III. Efficient solutions of ReMUS

The settings and usage of uncertain sources and recovery nodes jointly dominate the performance of the ReMUS problem.

A. Source-based building method of multicast forest

Given a network, modeled as an undirected graph, we set the transfer cost of each link as its weight. Although the shortest path can ensure the least transfer cost, the associated recovery cost is not always the least since the probability of packet loss may be non-minimal. Even though we suppose each link has a 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

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.
1:  for i=1 to |C| do
2:      calculate shortest paths P(A) from c_i to S;
3:  end for
4:  for i=1 to |D| do
5:      calculate shortest paths P(B) from d_i to S;
6:      for j=1 to |S| do
7:          C(B)_j ← addedCost(P(B)_j, C, F_2, α, β);
8:      end for
9:      calculate shortest paths P(C) from d_i to C;
10:     for k=1 to |C| do
11:         P(A + C)_k ← P(A)_k ∪ P(C)_k;
12:         C(A + C)_k ← addedCost(P(A + C)_k, C, F_2, α, β);
13:     end for
14:     P(D) ← find(P(B) ∪ P(A + C), min{C(B) ∪ C(A + C)});
15:     F ← F ∪ P(D);
16: end for
17: return multicast forest, F;
```

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. We first compute the shortest paths from the destination to all sources. Second, for each of such shortest path, we calculate the total cost, including the transfer cost and the recovery cost. Third, we select the shortest path with the least 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 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 lost packets to the destination based on the recovery model in Section II-B1. Therefore, the SR-based method can achieve the reliable and bandwidth-efficient multicast transfer.

Additionally, when the transfer cost 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 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 be still efficient, which is to find the shortest path for each destination. The entire time complexity of Algorithm 1 is $O(|D| \times |V|^2)$.

B. A general building method of multicast forest

When there are recovery nodes in networks, we can adopt the recovery model in Section II-B2. Assume that recovery n-odes are deployed in advance and their locations are fixed. The general building method of multicast forest needs to consider the locations of recovery nodes and sources. Meanwhile, it also takes the shared links between multiple destinations into account. In this section, we further design a general building method of multicast forest with as low cost as possible. The building process of multicast forest needs to consider the recovery nodes, and thus the method is also called 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 is to find the shortest paths $P(B)$ from d to all sources. Then, RN-based method will compute all shortest paths $P(C)$ from d to all candidate recovery nodes. To ensure that a selected source is responsible to deliver data to destination d, 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.

The main idea of Algorithm 2 is to select the path for each destination from $P(B)$ and $P(A+C)$. It is worth noting that there are some common links among different paths for these destinations. Therefore, the selecting strategy of Algorithm 2 is to pick the path that has minimal added total cost instead of the path that has minimal total cost for each destination. Algorithm 2 adds the selected paths for destinations into multicast forest F. When adding a path into F, the total cost of multicast forest F increases, and that will produce the added total cost. For the paths in $P(B)$, it is necessary to calculate the added total cost by calling the function $addedCost()$, which is to calculate the sum of the added transfer cost and the added recovery cost. Similarly, for each path in $P(A + C)$, we will compute the added total cost. The path that has the minimal added total cost will be selected. Finally, all selected paths between destination side and source side constitute the multicast forest F.

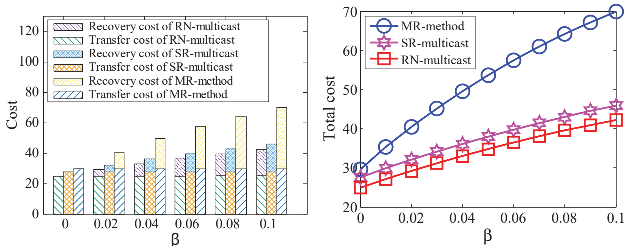
Time Complexity. The time complexity of calculating shortest paths from a node to all other nodes in a network is $O(|V|^2)$. Therefore, the time complexity of the first loop in Algorithm 2 is $O(|C|\times|V|^2)$. The time complexity of function $addedCost(P, C, F, \alpha, \beta)$ is $O(|V|)$. The time complexity of steps 6 and 10 are $O(|S|\times|V|)$ and $O(|C|\times|V|)$, respectively. 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 $O((|C| + |D|) \times |V|^2)$.

IV. Performance Evaluation

In this section, we conduct massive experiments to evaluate the performance of our SR-based and RN-based methods comparing with the modified RAERA method (MR-method), under varied settings of networks and multicast transfers. We simulate our algorithm in the real backbone network of Internet2 [11]. Based on statistic data shown in Nguyen et al. [12] and Xu et al. [13], we set the packet loss rate of each link from 1% to 10%. The source, destinations, and candidate recovery nodes are chosen randomly from each network. To eliminate the influence of random, each simulation result is averaged over 100 samples. Without loss of generality, we suppose that the transfer cost of each link is 1. However, our model and algorithms can apply to the situation where each link has different transfer cost. Unless otherwise specified, we set that the number of sources, destinations and recovery nodes are 3, 10 and 10, respectively.

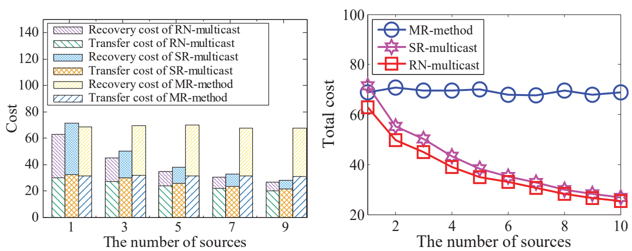
A. Impact of the probability of packet loss β

Fig. 2 shows the impact of β on the transfer cost, recovery cost and total cost of three methods given α=1. In Fig. 2(a), the recovery cost go up 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



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

Fig. 2. The impact of β on the performance of three methods under Internet2 topology.



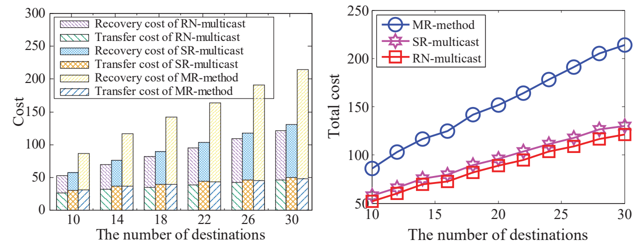
(a) The recovery and transfer cost of three methods.
(b) The changing trends of total cost.

Fig. 3. The impact of the number of sources on the performance of three methods under Internet2 topology.

of the total cost when β increases in Fig. 2(b). However, the changing of transfer cost is modest. When β=0, the recovery cost of three methods is 0 because there are no lost packets. Moreover, we can see that the transfer cost of multicast forest derived by our RN-based method is the least comparing with the SR-based method and the MR-method. Comparing recovery costs of three methods in Fig. 2(a), our RN-based and SR-based algorithms are more effective than MR-method without the change of β. Fig. 2(b) also shows that our RN-based method gets less total cost, owing to its smaller transfer cost than SR-based method. In conclusion, the recovery cost increases with the increase of β, and our RN-based method achieves the least total cost when β changes. Without losing generality, we will set α=1 and β=0.1 in next evaluations.

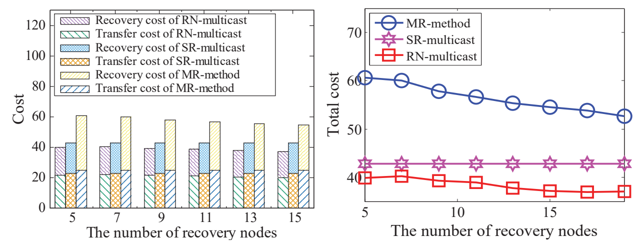
B. Impact of the number of sources

The increasing number of sources has a slight influence on the performance of MR-method because it just employs one source for multicast transfer. When there are only one source, MR-method achieves less transfer cost than the SR-based method in Fig. 3(a). However, as the increase of the number of sources, the performance of our SR-based and RN-based methods are improved rapidly. Furthermore, Fig. 3(b) shows that our RN-based method achieves the least total cost because it considers the locations of recovery nodes and reduces effectively the recovery cost of ReMUS. For our SR-based and RN-based methods, their recovery costs reduce rapidly to a quarter of transfer costs respectively when sources increase to 10 in Fig. 3(a). After that the number of sources increases to 8, the total costs of our RN-based and SR-based methods slowly decrease in Fig. 3(b). The facts also reflects that it is not necessary to use all sources for multicast transfer.



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

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



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

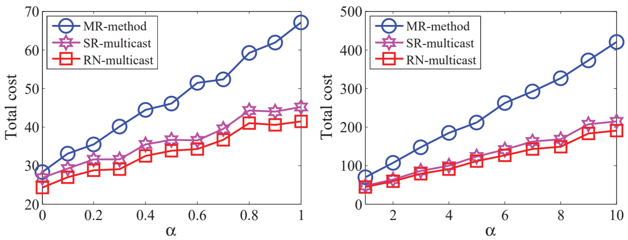
Fig. 5. The impact of the number of recovery nodes on the performance of three methods under Internet2 topology.

C. Impact of the number of destinations

Fig. 4 plots that the transfer cost, recovery cost and total cost of three methods increase with the increase of number of destinations. Fig. 4(a) shows that the recovery cost is significantly influenced by the quantity of destinations, especially for MR-method. The recovery cost of MR-method intensively changes when the number of destinations increases. Meanwhile, MR-based has always the highest total cost in Fig. 4(b). The changing trend of total cost for RN-based method is similar to the trend of SR-based method. The transfer costs of SR-based method and RN-based method are almost the same and slowly increase in Fig. 4(a). Fig. 4 shows that our RN-based method works effectively because it achieves the minimal total cost. Additionally, 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 destinations.

D. Impact of the number of recovery nodes

Fig. 5 plots that our RN-based method achieves the least transfer cost, recovery cost and total cost. MR-method and our RN-based method are both recovery-node-aware, and hence their costs decrease when the number of recovery nodes increases. However, the recovery and transfer cost of SR-based method are not influenced by the amount of recovery nodes in Fig. 5(a). The SR-based method applies to sources-based recovery and has nothing to do with recovery nodes. Thus, its cost remains steady when recovery nodes increase in Fig. 5(a). When few nodes are selected as recovery nodes, we can see that MR-method has higher recovery cost from Fig. 5(a). When recovery nodes increase, the performance of MR-method has a significantly improvement. Furthermore, Fig. 5(b) shows that our RN-based method achieves the minimal total cost, and its performance shows improvements when the number of recovery nodes increases.



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

Fig. 6. The impact of the variable α on the total cost of three methods.

E. Impact of variable α

Fig. 6 shows that the total cost of three methods all go up with the increase of α. However, our RN-based method always incurs the least total cost among the three methods without the value of α. Fig. 6 shows that our SR-based method achieves less total cost than the MR-method. Fig. 6 also reveals that our RN-based method has a better performance than SR-based method. In conclusion, we evaluate three reliable multicast methods under Internet2. Our RN-based method obtains the least recovery-costly multicast forest, and the multicast derived by our RN-based method always incurs the least 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 in the future.

V. Related Work

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 routing methods prefer to deliver the same content to a group destinations along a shortest-path tree, such as PIM-SM. Such methods, however, are not bandwidth-efficient since each of those shortest paths is calculated independently. The Steiner minimum tree (SMT) is more promising due to minimize the number of occupied links for a multicast group. Many approximation algorithms have been proposed to solve the SMT problem, which is a NP-hard problem. There are also overlay Steiner trees [14], for P2P environments.

The second one aims to ensure the reliable transmission of multicast. Recently, Shen et al. propose the Recover-aware Steiner Tree (RST) problem [6]. It introduces at least one on-tree recovery nodes between the source and each destination to facilitate local loss recovery. Additionally, many content delivery applications have utilized the content replica design for improving the robustness and efficiency [15], [9]. Each replica of a given file has the capability to serve as a source node for multicast transfers. That is, the source of a multicast is not necessary has to be fixed in specific location as long as certain constraints are satisfied. This brings a new multicast with uncertain sources, which could reduce more network bandwidth consumption than traditional multicast. When multicast group is provided multiple sources and recovery nodes, they can be jointly exploited to reduce the total cost of reliable multicast transfer.

VI. Conclusion

In this paper, we propose a novel reliable multicast routing with uncertain sources, named ReMUS. The settings of uncertain sources and recovery nodes dominate the performance of this new reliable multicast. The goal is to minimize the sum of the transfer cost and the recovery cost, while finding such a ReMUS is very challenging. Thus, we design the source-based multicast method 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 to minimize the total cost of ReMUS. We conduct extensive evaluations under the topologies of Internet2. The results indicate that our methods can efficiently realize the reliable and bandwidth-efficient multicast with uncertain sources, irrespective of the settings of networks and multicasts.

Acknowledgment

The authors thank all the anonymous reviewers for their insightful feedback. Besides, this work is supported by the National 973 Basic Research Program under grant No. 2014CB347800, the National Natural Science Foundation of Outstanding Youth Fund under grant No. 61422214.

References

[1] R. Malli, X. Zhang, and C. Qiao, "Benefit of multicasting in all-optical networks," *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. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[4] 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.

[5] 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.

[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.

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

[9] 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.

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

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

[12] 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.

[13] Y. Xu, C. Yu, J. Li, and Y. Liu, "Video telephony for end-consumers: Measurement study of google+, ichat, and skype," in *Proc. ACM Conference on Internet Measurement Conference*, 2012.

[14] 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.

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