



2019

Minimum-Cost Forest for Uncertain Multicast with Delay Constraints

Bangbang Ren

the College of Systems Engineering, National University of Defense Technology, Changsha 410072, China.

Geyao Cheng

the College of Systems Engineering, National University of Defense Technology, Changsha 410072, China.

Deke Guo

the College of Systems Engineering, National University of Defense Technology, Changsha 410072, China.

Follow this and additional works at: <https://tsinghuauniversitypress.researchcommons.org/tsinghua-science-and-technology>



Part of the [Computer Sciences Commons](#), and the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Bangbang Ren, Geyao Cheng, Deke Guo. Minimum-Cost Forest for Uncertain Multicast with Delay Constraints. *Tsinghua Science and Technology* 2019, 24(2): 147-159.

This Research Article is brought to you for free and open access by Tsinghua University Press: Journals Publishing. It has been accepted for inclusion in *Tsinghua Science and Technology* by an authorized editor of Tsinghua University Press: Journals Publishing.

Minimum-Cost Forest for Uncertain Multicast with Delay Constraints

Bangbang Ren, Geyao Cheng*, and Deke Guo

Abstract: The use of multicast transmission can efficiently reduce the consumption of network resources by jointly serving multiple destinations with a single source node. Currently, many multicast applications impose the constraint wherein multicast flows must be processed by a series of Virtual Network Functions (VNFs) before reaching their destinations. Given a multicast transmission, there are usually multiple server nodes, each of which is able to host all the required VNFs. Thus, the multicast flow should be initially steered to one or a few selected server nodes that act as pseudo sources, and the destinations will then retrieve new flow from any of these pseudo sources. In this paper, we model this kind of multicast as an uncertain multicast with multiple pseudo sources, whose routing structure is usually a forest consisting of multiple isolated trees. We then characterize and construct the Delay-guaranteed Minimum Cost Forest (D-MCF) such that each path from the source to the destination satisfies the end-to-end delay constraint. To tackle this NP-hard problem, we design two efficient methods, the Partition Algorithm (PA) and the Combination Algorithm (CA), to approximate the optimal solution. Theoretical analyses and evaluations indicate that these two methods can generate the desired routing forest for any multicast transfer. Moreover, the PA method achieves a better balance between performance and time consumption than the CA method. The evaluation results show that PA- $(\Omega + 20)$ can reduce total cost by 49.02% while consuming 12.59% more time, thus significantly outperforming the CA- $(\Omega + 20)$ method.

Key words: uncertain multicast; network function virtualization; delay guaranteed

1 Introduction

The concept of network softwarization promotes the reform of existing networks and the development of next-generation networks. Its key technologies include Network Function Virtualization (NFV), Software-Defined Networking (SDN), and network virtualization^[1]. Flexible networks can adapt to flows with special requirements that must traverse a series of network functions. In the traditional network, such network functions are implemented as middleboxes, which are often proprietary hardware, e.g., firewalls, WAN optimizers, and intrusion detection systems, to guarantee the quality of service for data

transfer. The large number of middleboxes means high up-front investment costs in hardware and hardware-based appliances, which constrain innovation in an increasingly network-centric connected world^[2].

NFV is emerging as a new networking paradigm for shaping existing networks and promoting innovation^[3,4]. NFV substitutes proprietary middleboxes with software programs running on commodities servers. Network functions in the NFV environment are called virtual network functions. The virtualization of network functions offers many potential benefits, such as reducing the extent of equipment investment, speeding innovation, enabling the provision of customized services, and so on. Network flows must always traverse some VNFs in a particular order before reaching their destinations. This requirement is known as the Service Function Chain (SFC)^[5]. Virtualization technology brings flexibility and scalability. All the VNFs of an SFC can be

• Bangbang Ren, Geyao Cheng, and Deke Guo are with the College of Systems Engineering, National University of Defense Technology, Changsha 410072, China. E-mail: {renbangbang11, chenggeyao13, dekeguo}@nudt.edu.cn.

* To whom correspondence should be addressed.

Manuscript received: 2017-05-13; accepted: 2017-05-18

deployed in a server node to form an SFC instance. In practice, the network should deploy many SFCs to improve its scalability and performance. Although many efforts have focused on the usage of SFCs in unicast transfers, there is little evidence of similar attempts with respect to multicast transfers. Recently, the authors in Ref. [6] proposed the NFV-enabled multicast, in which all VNFs are placed in a server node.

Multicasts are designed to deliver the same content from a single source node to a group of destination nodes. The current approach for supporting a multicast session is to establish a multicast tree, along which data can flow to all destinations. It has been successfully deployed in internet protocol television networks^[7], P2P networks^[8], and data center networks^[9,10]. Compared with unicasts, multicasts can significantly save bandwidth consumption and relieve the load of the source server. This is mainly because that multicasts avoid the unnecessary duplicated transmissions among a set of independent unicast paths towards destinations after multiplexing a shared multicast tree. However, when considering the NFV-enabled multicast, the resultant routing may be a pseudo-multicast tree^[6]. This property makes previous designing methods for multicast routing become infeasible.

To support an NFV-enabled multicast, we can first find a path from the single source node to a specific server node, which hosts an entire SFC, and then construct a multicast tree to span this server node and all destination nodes of the multicast transfer. In this way, we can tackle an NFV-enabled multicast as a traditional multicast through two steps. The first step is to find an unicast path from the source node to a server node hosting the required SFC. The second step is to treat this server node as a pseudo source, and then find a multicast tree spanning the pseudo source and all destinations. Currently, many methods have been designed to construct a desired multicast tree for a given multicast transfer. They, however, assume prior knowledge about the multicast characteristics, i.e., the source of multicast group is fixed and there exists only one source node. This kind of multicast is also called the deterministic multicast. Routing algorithms for deterministic multicasts have been classified in Ref. [11] into three categories: source-based tree algorithms, center-based tree algorithms^[12], and Steiner tree-based algorithms^[13].

The flexibility of NFV makes it easy to deploy

an entire SFC in any server node as long as certain constraints are satisfied. These server nodes can be denoted as a pseudo source set S . For any NFV-enabled multicast, the multicast flow can be firstly steered to any pseudo source in S from the source node, and then be delivered to destinations. That is, destination nodes can be served by any server node in S . Such multicasts with flexible pseudo source nodes are defined as uncertain multicast^[14–16]. Such a new multicast paradigm brings high expectation for more robust and flexible network services but incurs some essential technical challenges. The routing structure for an uncertain multicast is usually a forest, which spans all destination nodes and must satisfy the following constraints. In the resultant forest, each destination node can reach only one pseudo source node and any two utilized pseudo source nodes are not reachable with each other. Additionally, it is not necessary that all pseudo source nodes appear in the forest. This constraint makes it impossible to treat all pseudo source nodes as a whole virtual source node. Thus, the uncertain multicast problem cannot be simply reduced as the deterministic multicast problem.

On the one hand, it is well known that there exists different optimization principles when designing desired routing trees for a deterministic multicast. One representative principle is to minimize the total cost of all links on the constructed routing tree. The cost can be the bandwidth or expenditure of each link. On the other hand, many multicast applications impose a constraint on the end-to-end delay between the source and destination sides, especially those delay-sensitive applications. In this setting, a multicast tree is infeasible if the end-to-end delay constraint cannot be satisfied, even though it incurs the lower cost. In this paper, we aim to reconsider the construction of the desired forest for any uncertain multicast, so as to not only satisfy but also optimize the aforementioned principles.

In this paper, we characterize the routing problem of a multicast requiring SFC, with a constraint on the end-to-end delay, as the Delay-guaranteed Minimum Cost Forest (D-MCF). The D-MCF problem captures two well-known NP-hard optimization problems, the Steiner tree problem and the restricted shortest path, each of which has received much attention in past years^[17,18]. For this reason, we prove that building such a forest for the D-MCF problem is NP-hard and cannot be solved in polynomial time. Thus, we further propose two dedicated building algorithms, the Partition Algorithm (PA) and the Combination Algorithm (CA),

to approximate the optimal solution. The objective is to minimize the total cost of all edges in the resultant forest and ensure that the end-to-end delay in the forest does not violate the delay constraint. Each destination only connects to one selected source, and it is not necessary that all sources in the uncertain multicast are utilized by the forest.

The rest of this paper is organized as follows. In Section 2, we introduce related works. Section 3 describes the observation and formulation of our D-MCF problem. In Section 4, we present two efficient methods for the D-MCF problem. In Section 5, we present the results of our experiments, and we give some discussions in Section 6. In Section 7, we state our conclusion.

2 Related Work

Multicast transmission is designed to deliver the same content from a single source node to a group of destination nodes^[19]. Compared to unicast transmission, multicast can significantly reduce bandwidth consumption and relieve the load on the source server. The reason for this is that by multiplexing a shared multicast tree, multicast avoids unnecessary duplicate transmissions among a set of independent unicast paths to their destinations. The emergence of SDN has served to boost multicast application because of its central control ability^[20]. SDN separates the data plane and control plane, and its central computation approach can find the optimal multicast routing^[21].

Although multicast transmission brings particular benefits to SDN, it faces challenges when considering NFV requirements. Recently, several studies have been conducted on NFV-enabled multicasting in SDN^[6, 22, 23]. These works considered NFV requirements as a special function node, with the real routing for their solution being walks, which increase the complexity of the routing table. In our work, we first identify a number of server nodes that can deploy VNFs and then treat these nodes as pseudo source nodes. By connecting destinations with these pseudo source nodes, we can avoid repeatedly walking certain links.

Given a multicast group, several proposals have been made for constructing a multicast tree at minimum cost under given delay constraints. Note that the shortest path and minimum spanning tree are two classic problems in graph theory that are well addressed by many efficient and exact algorithms^[24, 25]. However,

determining the shortest path and the minimum spanning tree become NP-hard problems once we consider delay constraints^[17, 26]. The authors in Ref. [27] proposed a KPP algorithm, wherein the basic idea is to find a spanning tree for the closure graph of the constrained shortest path between the source and destination nodes. Parsa et al.^[28] proposed a heuristic algorithm based on a feasible search space, which starts with a minimum delay tree and then iteratively decreases the cost of the delay-bounded tree.

In all of the above studies, the authors investigated deterministic multicasts in which the desired routing structure for any multicast group is a specific tree. In contrast, the best routing structure for an uncertain multicast is usually a forest consisting of multiple disconnected trees. The authors in Ref. [14] proposed two approximation algorithms for constructing a minimum cost forest, and those in Ref. [29] conducted a study to minimize the maximum bandwidth utilization of congested links. In summary, very few works have considered the routing problem of an uncertain multicast from the aspect of delay constraints.

3 Problem Formulation

In this section, we describe the D-MCF problem in details. Section 3.1 presents an example of D-MCF problem and explains how the delay constraint influences the routing forest. In Section 3.2, the D-MCF problem is further characterized as an Integer Programming (IP) model. Section 3.3 proves that D-MCF problem is NP-hard.

3.1 Problem observation

For simplicity, we use an undirected graph $G = (V, E, c(e), d(e))$ to represent a network, where V denotes the node set and E denotes the edge set. Each edge $e \in E$ is associated with a cost $c(e) : E \rightarrow \mathbf{Z}^+$ and a delay $d(e) : E \rightarrow \mathbf{Z}^+$, where \mathbf{Z}^+ is a set of positive integers. Before giving the formal model of D-MCF, we first give the definition of the pseudo source node.

Definition 1 Pseudo source nodes are those selected servers that host an entire SFC instance. Pseudo source nodes get flow from the source and transmit them to destinations to ensure flow be processed by SFC before reaching to destinations.

In Section 1, we point out that a multicast with NFV requirements can be regarded as an uncertain multicast which has multiple pseudo source nodes, and

any destination node can pull data from any pseudo source node. Thus, in our network model, an uncertain multicast group can be denoted by $\delta = (S, D)$, where S represents the set of pseudo source nodes with $1 \leq |S| \leq |V|$ and D represents the set of destination nodes with $1 \leq |D| \leq |V|$. We assume that each network link has adequate bandwidth. Thus, there is no need to consider the size of transferred data in a multicast session.

Figure 1 shows a graph $G = (V, E, c(e), d(e))$. We need to find a routing topology for a multicast with NFV requirements. That is, the flow must be processed by the VNFs before reaching destinations. In Section 1, we have pointed out that we can first place VNFs in some servers. In Fig. 1, $S = \{F, L\}$ are two selected servers hosting an entire SFC instance. Through this way, an uncertain multicast $\delta = \{S, D\}$, where the pseudo source nodes $S = \{F, L\}$ and the destinations $D = \{A, B, C, G, H, I, J, K\}$, is constructed. As previously mentioned, routing for a deterministic multicast is exhibited as a tree. However, routing for an uncertain multicast usually forms a forest F which consists of some isolated trees and satisfies the following constraints:

- (1) Each tree $T_i \in F$ has one single pseudo source in S .
- (2) In F , the pseudo source nodes in a pair cannot reach to each other.
- (3) Each destination node connects with only one pseudo source node.

Given an uncertain multicast, many feasible routing forests exist, each of which can satisfy the above constraints. For example, Fig. 2 reports three feasible forests, each of which satisfies the aforementioned constraints. However, we must select the best one according to our optimization objective. In this study,

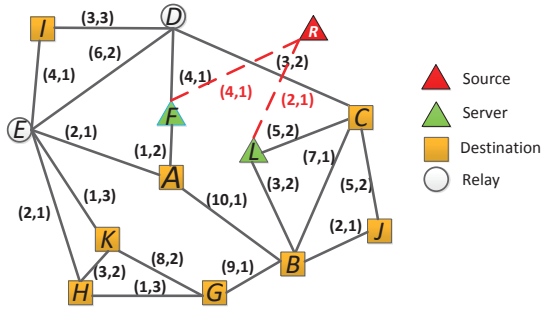
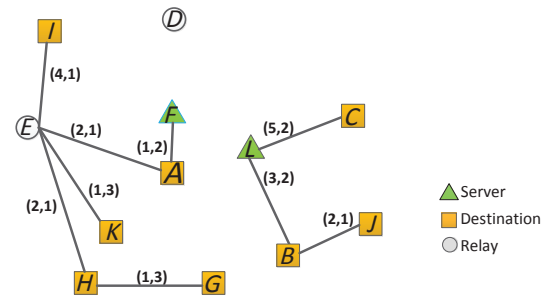
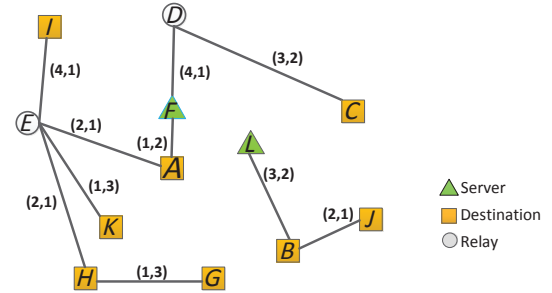


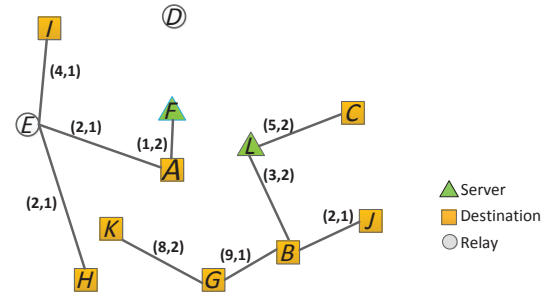
Fig. 1 An illustrative example of an uncertain multicast group $\delta = (S, D)$ in a network $G = (V, E, c, d)$, with source node R , pseudo source nodes $S = \{F, L\}$, and destinations $D = \{A, B, C, G, H, I, J, K\}$.



(a) A forest with a delay bound $\Delta = 8$ offers the total cost of 21



(b) A forest with delay bound $\Delta = 8$ offers the total cost of 23



(c) A forest with delay bound $\Delta = 6$ offers the total cost of 36

Fig. 2 Three routing forests for an uncertain multicast under different delay bounds.

we desire to find a forest F which results in the minimum total cost and satisfies the end-to-end delay constraint. Given a delay bound Δ , we must ensure that the delay from each destination node $d \in D$ to its corresponding pseudo source node $s \in S$ is less than Δ . The path between the pair of pseudo source and destination nodes is labeled as P_{sd} . The end-to-end delay along the path must satisfy the delay constraint.

$$d(P_{sd}) = \sum_{e \in P_{sd}} d(e) < \Delta.$$

In the resultant routing forest, all end-to-end delay between pseudo source and destination must obey the delay constraint. Otherwise, the solution becomes infeasible. The quality of a feasible forest for an uncertain multicast problem is evaluated by the total cost of all edges. The optimizing function for an uncertain multicast is as follows:

$$\text{Min} \sum_{e \in F} c(e).$$

For example, if $\Delta = 8$, Fig. 2 reports three feasible forests with different total costs and the best solution is shown in Fig. 2a. However, if $\Delta = 6$, then the forests in Figs. 2a and 2b will be infeasible because $d(P_{FG}) = 7 > \Delta$ violates the end-to-end delay constraint.

3.2 Integer programming

After presenting our D-MCF problem, we further characterize it as an integer programming model. Let N_v denote the set of neighbor nodes of v in G , and u is in N_v if $e_{u,v}$ is an edge from u to v in E . The output forest F needs to ensure that there is only one path in F from S to every node in D . To achieve this goal, our problem includes the following binary decision variables. The binary variable $\pi_{d,u,v}$ is used to indicate that edge $e_{u,v}$ is in the path from S to a destination node d , and another binary variable $\varepsilon_{u,v}$ indicates that edge $e_{u,v}$ is in F . The cost and delay of edge $e_{u,v}$ are denoted by $c_{u,v}$ and $d_{u,v}$, respectively. Intuitively, when we are able to find the path from S to each destination node d with $\pi_{d,u,v} = 1$ on every edge $e_{u,v}$ in the path, the routing of the forest with $\varepsilon_{u,v} = 1$ for every edge $e_{u,v}$ in F can be constructed via the union of the paths from S to all destination nodes in D . The integer programming formulation is as follows:

$$\begin{aligned} & \text{Min } \sum_{\varepsilon_{u,v} \in E} c_{u,v} \varepsilon_{u,v}, \\ (1) \text{ s.t. } & \sum_{s \in S} \sum_{v \in N_s} \pi_{d,s,v} = 1, \forall d \in D; \\ (2) \text{ s.t. } & \sum_{v \in N_s, s \in S} \pi_{d,s,v} - \sum_{v \in N_s, s \in S} \pi_{d,v,s} = 1, \forall d \in D; \\ (3) \text{ s.t. } & \sum_{u \in N_d} \pi_{d,u,d} - \sum_{u \in N_d} \pi_{d,d,u} = 1, \forall d \in D; \\ (4) \text{ s.t. } & \sum_{v \in N_u} \pi_{d,v,u} = \sum_{v \in N_u} \pi_{d,u,v}, \forall d \in D, \forall u \in V, \\ & u \notin S, u \notin D; \\ (5) \text{ s.t. } & \sum_{u \in V, v \in V, u \neq v} \pi_{d,u,v} d_{u,v} < \Delta, \forall d \in D; \\ (6) \text{ s.t. } & \pi_{d,u,v} \leq \varepsilon_{u,v}, \forall d \in D, \forall s \in S. \end{aligned}$$

For each destination node $d \in D$, Constraint (1) indicates that there is only one path from d to S . That is, each destination node in D can reach only one pseudo source node. Thus, no path exists among pseudo source nodes in the forest. The next three constraints, i.e., Constraints (2), (3), and (4), are the flow-continuity constraints to find the path from S to each destination node d in D . Constraint (2) states that the net outgoing

flow from S to a destination node is one, implying that one pseudo source s from S is selected and one edge $e_{s,v}$ from s to any neighbor node v needs to be selected with $\pi_{d,s,v} = 1$. Constraint (3) ensures that the net incoming flow to d is one, implying that at least one edge $e_{u,d}$ from any neighbor node $u \in N_d$ must be selected with $\pi_{d,u,d} = 1$.

For every other node u , Constraint (4) guarantees that the incoming and outgoing flows for u are equal. Constraint (5) ensures that the total delay of path between d and s in the output forest F never violates the delay constraint Δ . Constraint (6) is formulated to find the routing of the forest, i.e., $\varepsilon_{u,v}$. It states that $\varepsilon_{u,v}$ must be 1 if edge $e_{u,v}$ is included in the path from S to at least one d , i.e., $\pi_{d,u,v} = 1$. The forest F is the union of paths from all destinations to the pseudo source side.

3.3 Hardness of D-MCF

In Ref. [18], a Steiner tree problem was proved to be NP-hard by being reduced to an exact cover by 3-sets. D-MCF also can be reduced to exact cover by 3-sets. Here, we prove it by using reduction to absurdity. Both minimum spanning tree and shortest path with delay bound can be a special case of D-MCF, which have been proved NP-hard. In addition, suppose that there exists an accurate polynomial algorithm $A(m, \Delta)$ where m denotes the number of source nodes and Δ denotes delay bound to solve D-MCF. Then we can use $A(m = 1, \Delta = \infty)$ to solve Steiner tree problem in polynomial time, which violates that Steiner tree problem is NP-hard. Thus, D-MCF problem is also NP-hard.

4 Design of the Forest Building Methods

In this section, we propose two efficient algorithms to solve the D-MCF problem: the PA and CA. The PA basically involves partitioning an uncertain multicast into several independent groups, each of which has one pseudo source. Some groups may have one single pseudo source. In this way, we can construct delay constrained Steiner trees for such independent groups, and then combine such trees into a forest. On the other hand, CA first finds the minimum cost path that satisfies the delay constraint from the pseudo source side to each destination and then combines these unicast paths into a feasible forest.

4.1 Partition algorithm

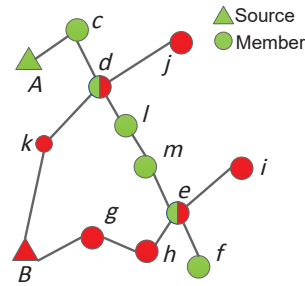
Deterministic multicast is the special case of the general uncertain multicast. The resultant routing forest of an

uncertain multicast usually contains some isolated trees, each of which can be seen as the multicast routing of a small deterministic multicast group. This fact inspires us to partition an entire uncertain multicast group into several independent groups, each of which contains only one pseudo source node and can be seen as a deterministic multicast. PA first builds a routing tree for each independent deterministic multicast and then combines these routing trees to get a resultant routing forest. Before discussing the details of PA, we first give a formal definition of the *constrained cheapest path*, which is very important in the processes of group partition and tree construction.

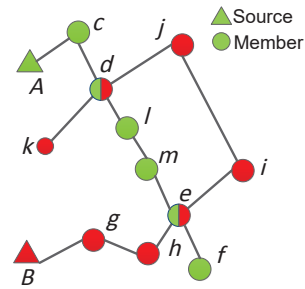
Definition 2 For any pair of nodes u and v , a **constrained cheapest path** between them is defined as the least cost path, $\Gamma_{u,v}$, whose end-to-end delay is less than Δ . The cost and delay of such a path are denoted by $c(\Gamma_{uv})$ and $d(\Gamma_{uv})$, respectively.

The constrained cheapest path plays a key role in the group partition. Reference [17] proposed an algorithm to derive the constrained cheapest paths between every pair of nodes using the dynamic programming. The computational complexity of this algorithm is $O(|V|^3\Delta)$. For a destination node d_k , we first find the set of constrained cheapest paths towards each pseudo source node $\{s_1, s_2, \dots, s_m\}$, denoted as $\{\Gamma_{d_k s_1}, \Gamma_{d_k s_2}, \dots, \Gamma_{d_k s_m}\}$. Among such paths of d_k , we choose the one with the least cost. Without loss of generality, we assume that $\Gamma_{d_k s_j}$ has the least cost, and then put d_k and s_j in a group. Repeating this procedure can divide $|D|$ destination nodes into at most $|S|$ isolated groups, since some pseudo sources may not be selected by any destination node. That is, an uncertain multicast group can be divided into at most m deterministic multicast groups.

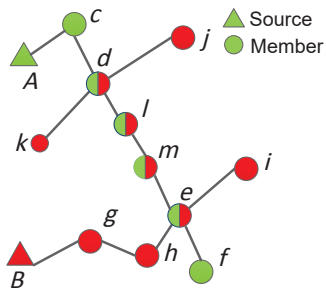
However, the routing forest cannot be constructed directly as the simple combination of the involved routing trees. Figure 3 shows different situations when combining two trees. The red tree has nodes $\{B, g, h, e, i, j, d, k\}$ and the green tree has nodes $\{A, c, d, l, m, e, f\}$, in which triangles denote pseudo source node along with circles denote other non-root nodes of trees. Combining such two multicast trees directly may violate constraints of routing forest mentioned in Section 3.1. Figures 3a and 3b show that the two trees have some common nodes. The only difference is that Fig. 3b exhibits a circle (e, i, j, d, l, m, e) , which does not contain any pseudo source node. Figure 3c describes a situation whereby



(a) Case 1: Combination of two multicast trees exhibits some common nodes



(b) Case 2: Combination of two multicast trees forms a circle



(c) Case 3: Combination of two multicast trees exhibits some common edges

Fig. 3 Illustrative examples of the combination of two multicast trees.

the two trees share a common path (d, l, m, e) . Such observations require pruning some edges to break the connectivity among pseudo source nodes and avoid circles, so as to derive the final routing forest. Before designing an efficient pruning principle, we first discuss **common nodes**.

Definition 3 When adding a new tree into an existing forest, some nodes may exist in the new tree as well as the existing forest; hence, they are called **common nodes**.

Common nodes play a key role in the process of pruning edges in the resultant routing topology after combining the involved multicast trees. We take Fig. 3a as an example to illustrate our edge pruning principle in detail. The red part denotes a current forest F_{k-1} , while the green part denotes a newly added tree T_k . First, we check whether there exists a circle when combining F_{k-1} and T_k . There is no single circle in F_{k-1} and T_k ,

so the formed circle must have common nodes, which are $\{d, e\}$, as shown in Fig. 3a.

Here, we show how to break the circle $(d, k, B, g, h, e, m, l, d)$. We calculate the delay of all paths from each node in this circle to the two pseudo sources. Without loss of generality, we assume that path P_{Am} has the highest delay. We then delete the last hop of P_{Am} , i.e., edge (l, m) , so as to break the circle. After breaking the circle, we still have to break the connectivity of the two pseudo sources. That is, we must delete an edge of path $P_{AB} = \{A, c, d, k, B\}$. We compare the delay of paths between common node d and two pseudo sources, i.e., P_{Ad} and P_{Bd} . Assuming that $d(P_{Ad}) > d(P_{Bd})$, then we delete edge (c, d) which is the last hop of path P_{Ad} . After deleting (c, d) and (l, m) , we can get a forest F_k that satisfies the constraints. Proposition 1 indicates that this pruning principle ensures a feasible solution.

Proposition 1 The **Pruning principle** involves two steps. In Step 1, find circles in forest F_k . For each circle C , calculate the delay of all shortest delay paths between each member node of C and each pseudo source node. Select the path with the highest delay, and then delete the link of this shortest delay path which connects to the member node. In Step 2, find the shortest path between any pair of pseudo source nodes and compare the delay from one common node in this source-to-source path to two pseudo source nodes. Delete the link that connects to the common node and lies in the higher delay side. Executing such pruning principle ensures a feasible routing forest.

Proof To ease the description, we take Fig. 3a as an example. In Fig. 3a, the red forest F_{k-1} spans nodes $\{g, h, e, i, j, d, k\}$ while the green tree T_k contains nodes $\{A, c, d, l, m, e, f\}$. It is easy to find that such two parts share a common node set $\{d, e\}$.

In Step 1, we find a circle $C = \{d, k, B, g, h, e, m, l, d\}$ in Fig. 3a. Then we find the shortest delay paths between all nodes in C and pseudo source nodes, and denote them as P_{ij} , where $i \in \{A, B\}$ and $j \in \{d, k, B, g, h, e, m, l\}$. We compare the delay of such paths, and assume that P_{Am} has the highest delay, and then accordingly delete the last edge of P_{Am} , i.e., (l, m) . To depict that this method is feasible without violating the delay constraint, we take details of multicast transfer as an example. In a time t , $\{A, B\}$ send packet, and nodes $j \in \{d, k, g, h, e, m, l\}$ need this packet. All j will get packet before time $t + \Delta$, since F_{k-1} and T_k both satisfy the delay bound. Since P_{Am} has the highest delay to get packet from $\{A, B\}$, m can get packet from

e instead of l under our assumption. Thus, we can delete edge (l, m) to break the circle without violating the delay constraint. This procedure is repeated until there is no circle in the combined graph.

After breaking all circles, we proceed to Step 2, where we need to break the connectivity of pseudo sources. According to our pruning principle, Step 1 ensures that there exists only one path P_{AB} between A and B because no circle exists. Additionally, there must be a common node in this path. For example, the common node is d in Fig. 3a. Then we calculate the delay of P_{Ad} and P_{Bd} , and assume that P_{Ad} has a higher delay. With this assumption, we delete the last hop of P_{Ad} , i.e., (c, d) . This can ensure a feasible solution since all nodes reaching to A through d can now reach to B through d with the lower delay. Step 2 is repeated until there is no path between any pair of pseudo sources.

In summary, the above procedures about dealing with common nodes are very effective for deriving a desired uncertain multicast forest, without violating the delay and routing forest constraints. Thus, Proposition 1 is proved. ■

After pruning the edges, some new leaf nodes may be introduced, which are not destination nodes. Such leaf nodes should be deleted to further reduce the total cost, and then the PA can finally produce a routing forest. The details of the PA are shown in Algorithm 1.

Algorithm 1 Partition Algorithm (PA)

Require: An undirected network $G = (V, E, c(e), d(e))$ and an uncertain multicast session $\delta = (S, D)$ with source nodes set $S = \{s_1, s_2, \dots, s_m\}$ and destination nodes set $D = \{d_1, d_2, \dots, d_n\}$.

Ensure: A routing forest F

- 1: Using constrained cheapest path to divide δ and let $\{\Phi_1, \Phi_2, \dots, \Phi_m\}$ denote the partition result. That is, each Φ_i include one pseudo source node s_i and some destination nodes.
 - 2: **for all** $\Phi_i \in \{\Phi_1, \Phi_2, \dots, \Phi_m\}$ **do**
 - 3: Construct a complete graph C of all member nodes in Φ_i via exploiting those constrained cheapest paths
 - 4: Find a spanning tree in C under the delay constraint.
 - 5: Map the resultant spanning tree in C into T_i in G ,
 - 6: $F = T_1$
 - 7: **for** $k = 2$ to m **do**
 - 8: **Prune** (F, T_k)
 - 9: **Deleteextranode** (F)
 - 10: **return** F
- Prune** (F, T)
- 1: $F = F + T$
 - 2: Delete potential edges using the designed pruning rational
 - 3: **Return** F
- Deleteextranode** (F)
- 1: **while** There exists leaf node q that is not a destination node in F **do**
 - 2: delete q
 - 3: **Return** F
-

Theorem 1 The time complexity of PA is $O(|V|^3\Delta)$.

Proof As shown in Algorithm 1, it first takes $O(|V|^3\Delta)$ operations to calculate all constrained cheapest paths and need $O(|S||D|)$ operations to finish the partition task. After partitioning, $O(|V|^3\Delta)$ operations are again used to generate a multicast tree for each group. Compared to the generation of constrained cheapest paths, the time required to find spanning trees and map it into the original network can be negligible. A total of $O(|V|^2)$ operations is required to combining these trees because there are $O(|V|^2)$ edges in the network. In addition, $O(|V|^3)$ operations are needed to remove loops using Dijkstra's algorithm. The last step, i.e., deletion of non-destination leaf nodes, requires $O(|V|^2)$ operations. Overall, we can conclude that the complexity of PA is $O(|V|^3\Delta)$. ■

4.2 Combination algorithm

Given an uncertain multicast, we aim to find a minimum cost forest without violating the delay constraint. Since the delay constraint focuses on verifying each destination node, we just need to ensure that each path from a pseudo source to a destination is feasible. Thus, another way to tackle this problem is to first find feasible paths, and then combine them into a feasible forest. We propose the CA to solve the D-MCF problem with two phases and the details of the CA are shown in Algorithm 2.

- **The single path phase** aims to find constrained cheapest paths for each destination node.
- **The whole forest phase** devotes to combining all cheapest paths into a feasible forest.

In the single path phase, we aim to find constrained cheapest paths for each destination node, and these paths ensure that destinations can receive data from pseudo sources under the delay bound. The principle to break loops in the second phase is shown in Proposition 1. The combination of such paths incurs a feasible solution, which can be proved by Proposition 2.

Algorithm 2 Combination algorithm

Require: An uncertain multicast transfer in an undirected network $G = (V, E, c, d)$, where the pseudo source node set is $S = \{s_1, s_2, \dots, s_m\}$ and the destination node set is $D = \{d_1, d_2, \dots, d_n\}$.

Ensure: A routing forest F .

- 1: **for** $i = 1$ to n **do**
 - 2: Find the constrained cheapest path P_i between d_i and S
 - 3: $F = F + P_i$
 - 4: Find loops in F and break them without violating the delay constraint.
-

Proposition 2 In the whole forest phase, all constrained cheapest paths produced by the single path phase are combined, and this combination ensures a feasible solution.

Proof First, the constrained cheapest paths ensure that destinations can get data from pseudo sources without violating the delay constraints. Some loops may occur when all these single paths are combined and we can break them by deleting an edge. In the principles followed to delete this edge, the delay constraints are never violated for the concerned destinations. Finally, we can conclude that the combination of all shortest paths produces a feasible routing forest. Thus, Proposition 2 is proved. ■

Theorem 2 The complexity of CA is $O(|V|^3\Delta)$.

Proof In the single path phase, it takes $O(|V|^3\Delta)$ operations to find all constrained cheapest paths. In the whole forest phase, it takes $O(|V|^2)$ operations to combine all paths. To remove the loops by the method shown in Proposition 1, we use Dijkstra's algorithm^[30] to decide the edge to be delete at the cost of $O(|V|^3)$ operations consumption. Thus, the total complexity of the CA is $O(|V|^3\Delta)$. ■

Theorem 3 The loose approximation ratio of CA is $|D|$. If we denote F^* as the optimal solution and F' as the resultant forest of CA, we can state that their approximation ratio is less than $|D|$.

Proof For a clear proof, we first give two cost metrics: the total cost for the whole forest F , $AC(F)$, and the average cost of each correspondence path from pseudo sources to each destination, $BC(F)$. We denote the optimal forest as F^* and the feasible forest resulting from CA as F' . Here, $AC(F') \leq |D|BC(F')$ since $AC(F')$ counts $c(l)$ for every $l \in F'$ once and $|D|BC(F')$ counts each $c(l)$ at least once. In addition, $|D|BC(F') \leq |D|BC_{\max}(F')$, where $BC_{\max}(F')$ is the cost of the unicast path which has the maximum cost in F' . Furthermore, $BC_{\max}(F') \leq AC(F^*)$, since $BC_{\max}(F')$ is the least costly way of reaching the most costly destination. The insight of this property is that CA construct forest with constrained cheapest path, and F^* gets to that destination (among other things). Thus, $AC(F^*)$ must be at least $BC_{\max}(F')$. Then we can get $AC(F') \leq |D|BC(F') \leq |D|BC_{\max}(F') \leq |D|AC(F^*)$. Thus, the approximation ratio is less than $|D|$, Theorem 3 is proved. ■

5 Experiments

Comprehensive evaluations were conducted to quantify

the performance of the proposed PA and CA methods. Given any uncertain multicast, we compared our two methods with Enhanced-Minimum Cost Forst (E-MCF), a representative method for building a minimum cost forest regardless of the delay bound. To ensure the existence of a feasible D-MCF solution, we relax the delay bound to $\Omega + 20$, where Ω denotes the maximum delay of the shortest delay paths between pseudo sources and destinations. Without loss of generality, we generated a series of Erdos-Renyi random graphs^[23] on a rectangle grid with the size $[50 \times 50]$ as the underlying network topology. The cost of each link was decided by the distance between two nodes, and the caused delay of each link in the network was a random integer from 1 to 3.

We evaluated these methods under different network sizes, different scales of uncertain multicast transfers, and a varied number of pseudo sources in uncertain multicast transfers. The following three metrics were studied: the total cost of the forest, the maximum delay path in the resultant forest, and the time consumption of constructing the forest. In addition, we compare the performance of our two algorithms with different delay bounds under various network sizes.

5.1 Impact of network size

We studied the impact of the network size on the aforementioned three metrics under given setting of uncertain multicasts. For each uncertain multicast transfer, 3 pseudo source nodes and 45 destination nodes were selected randomly from the whole topology. The results generated when the network size was changed from 100 to 300 (Figs. 4a–4c) show that both the total cost and the consumption significantly increased with the network size. This is mainly because a multicast forest in a larger network possesses more links. Figure 4a further indicates that more cost will be introduced into the derived multicast forest to meet the delay bound constraint. Because of the delay bound

relaxation, E-MCF always generates the least-cost forest, and CA always incurs the highest cost. As previously explained, the CA focuses on optimizing each destination dependently; however, this local optimization strategy fails to achieve the global target.

On the contrary, as shown in Fig. 4b, E-MCF leads to the highest maximum delay. This result keeps consistent with our inference in Fig. 4a, that is, a stricter delay bound incurs a higher cost. In addition, the time consumption curve of E-MCF lies below that of PA and CA. This is because both PA and CA spend more time in calculating constrained cheapest paths. As for PA and CA perform differently in the term of time-consumption, this is because that PA has additional procedure to partition groups. We can clearly see that the time-consumption of PA in Fig. 4c is slightly more than that of CA; hence, the PA algorithm achieves a balance between the total cost and the time-consumption, when the underlying network is expanded.

5.2 Impact of group size

We further measured the impact of the group size, which is the total number of multicast member nodes, on the metrics. Given $|S| = 3$, we varied the number of destinations from 45 to 105. The changes of the performance metrics with the increase of group size and consequently the transfer size (Figs. 5a–5c) show that the cost and delay increased with the group size, but the time consumption remained nearly stable. This is because when the group size increases, more links are employed to construct the multicast forest, and thus increases cost. As for time-consumption, we have pointed out that complexity of PA and CA is all decided by the number of all nodes in network. Thus, we can see that the time-consumption was nearly stable with increase in transfer size.

5.3 Impact of number of pseudo sources

In this paper, we further evaluate the impact of the

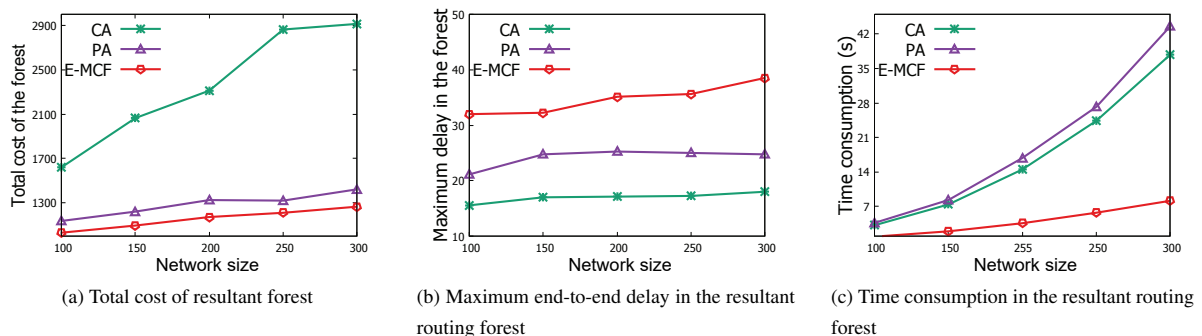


Fig. 4 Changing trends of three metrics when n ranges from 100 to 300 under different forest building methods.

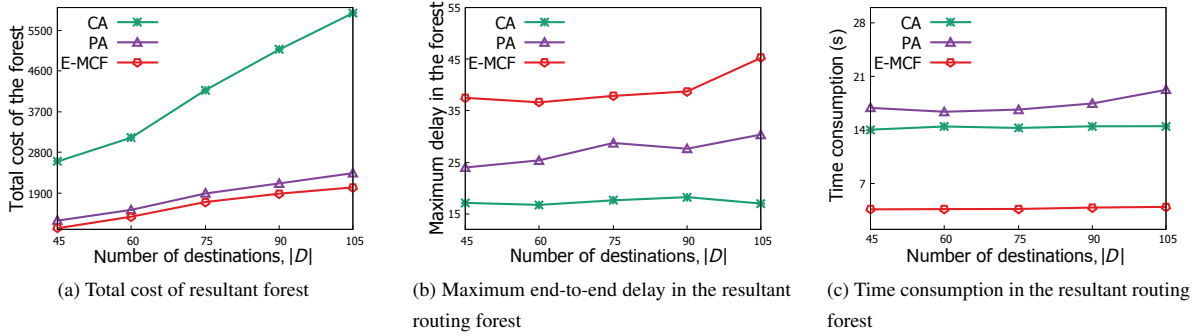


Fig. 5 Changing trends of three metrics when the number of destination ranges from 45 to 105 with $|S| = 3$ under different forest building methods.

number of pseudo sources $|S|$ while keeping other parameters the same. Figures 6a, 6b, and 6c show plots of the results when $|S|$ varies. Given $|S| + |D| = 120$, we varied the number of pseudo sources from 1 to 5. As shown in Fig. 6a, the costs of the multicast forests of three methods all decrease with an increase in the number of pseudo sources, but this trend is more obvious for the CA. The reason for this is that having more pseudo sources provides more alternative paths to reduce the cost, and the CA provides the richest space for optimization. As shown in Fig. 6b, the generated maximum delay has an obvious decreasing tendency as $|S|$ increases, which indicates that the destinations have more opportunities to obtain data from the nearest pseudo source.

In Fig. 6c, we can see that although all these methods consume more time as the number of pseudo source increases, their slopes change; the slopes increase slightly from $|S| = 1$ to $|S| = 3$, but then increase sharply from $|S| = 3$ to $|S| = 5$. This observation and the trend shown in Fig. 6a reveal that simply increasing the number of pseudo sources may not yield a proportional improvement in the performance. Therefore, having a large number of replicas may not be a good choice for building a robust distributed system.

5.4 Impact of delay bound

In above experiments, we compared our proposed

methods considering delay bound with the E-MCF, which does not consider delay limitations. We made a preliminary conclusion that a loose delay bound may incur a low-cost forest, but does so at the cost of consuming more time before all the destinations receive data from the pseudo sources. Figure 7 shows a more comprehensive result regarding the impact of a delay bound.

In this experiment, the number of pseudo sources and the multicast group size are stable, and we focus on the performance of PA and CA with two delay bounds, $\Omega + 5$ and $\Omega + 20$. Figure 7a shows a plot in which we can see that the PA always generates forests with a greater maximum delay than the CA under the same condition. The average delay shows the similar trend, as shown in Fig. 7b. The underlying reason for this is that the CA optimizes the forest from the local viewpoint by combining unicast paths, whereas PA always combines trees. Since our optimization goal is to minimize total cost, the PA may relax the delay bound to reduce cost. This can be proved by the fact that PA- $(\Omega + 20)$ and CA- $(\Omega + 20)$ always have higher values than PA- $(\Omega + 5)$ and CA- $(\Omega + 5)$ in terms of their maximum and average delays, respectively.

On the other hand, Fig. 7c indicates that forests generated by PA- $(\Omega + 20)$ have a lower cost than

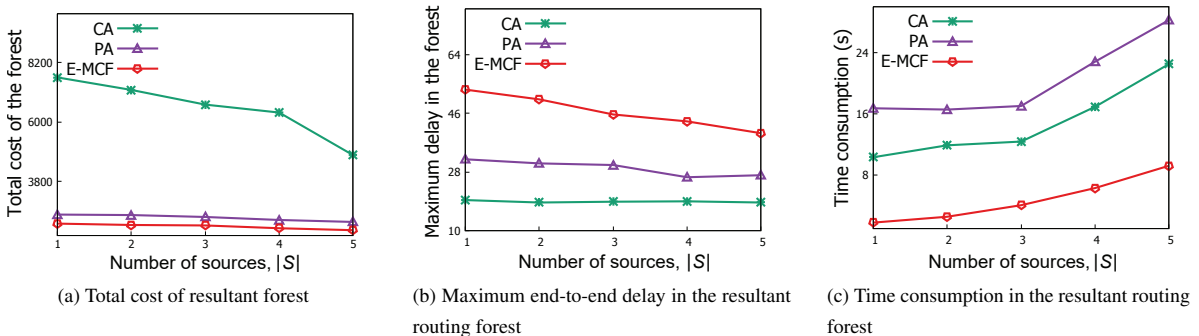
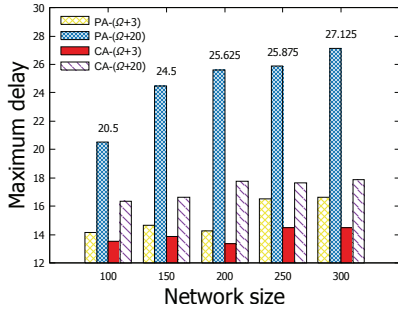
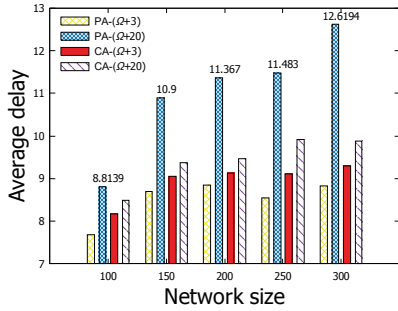


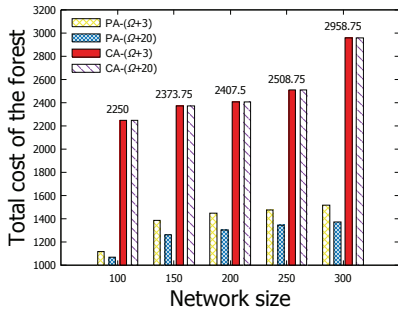
Fig. 6 Changing trends of three metrics when the number of pseudo sources ranges from 1 to 5 with $|S| + |D| = 120$ under different forest building methods.



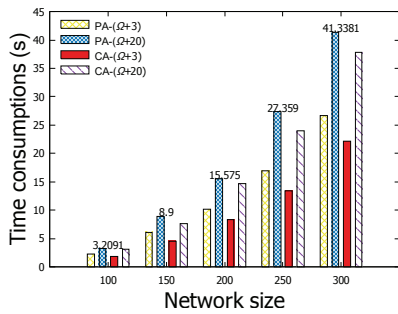
(a) Maximum delay of resultant forest under different delay bounds



(b) Average delay of resultant forest under different delay bounds



(c) Total cost of resultant forest under different delay bounds



(d) Time consumption of generating resultant forest under different delay bounds

Fig. 7 Comparisons of routing forests under different settings of the delay bounds.

PA-($\Omega + 5$). This trend fits the intuitive conclusion that a loose delay bound yields a low-cost forest. Additionally, Fig. 7d shows the time consumptions of the PA and CA, in which we can see that for the same delay bound, PA always consumes more time than CA. However, this extra time cost is acceptable,

because when we set the delay bound to $\Omega + 5$, the PA can reduce the total cost by 44.32% while consuming 24.90% more time than the CA. If the delay bound is $\Omega + 20$, then the PA can reduce the total cost by 49.02% and consume only 12.59% more time.

In summary, our comparisons of the PA and CA results indicate that the delay bound is an essential factor to consider when constructing a multicast forest for any uncertain multicast. Our proposed PA method can balance the total cost, the end-to-end delay, and the time consumption.

6 Discussion

In previous sections, we discussed the uncertain multicast model and solved it using the two proposed approximation algorithms. Here, we briefly discuss our design choices and introduce some interesting issues for future research.

Selection of pseudo source nodes. In our work, we proposed the D-MCF problem and concentrated on designing methods to solve it. However, to find routings for NFV-enabled multicast transmission, the method used to select the server nodes for use as pseudo sources has great impact. Our experimental results show that the number of pseudo sources can influence the quality of the solutions. Also, the selected pseudo sources will influence the value of the delay bound.

Same delay bound value for all pseudo sources. We use a fixed delay bound for all pseudo sources in the D-MCF. Considering the background of the D-MCF, the delays from different pseudo sources and source nodes differ. Thus, pseudo sources that generate less delay to the source can be assigned a looser delay bound. Additionally, even destinations of the multicast group may have different delay tolerances, which can be classified as a problem of delay variation.

Order of VNFs in SFC. We assume that all required VNFs can be deployed in a single server node. This assumption simplifies the real routing for NFV-enabled multicast transmission. In fact, all VNFs of the SFC are typically deployed in different nodes, and this situation will reshape the D-MCF problem, whereby the pseudo sources may be transformed into multi-level pseudo sources.

7 Conclusion

Network function virtualization is emerging as a new networking paradigm for shaping existing networks.

Virtual network functions separate network functions from proprietary hardware. This deployment flexibility influences the pattern of multicast transmission, and in this paper, we proposed the uncertain multicast as a novel and general model of multicast transfer. We characterized and addressed the D-MCF problem to ensure the quality of service of multicast applications. This motivated us to search for the minimum cost forest for an uncertain multicast with delay constraints. We formulated the D-MCF problem as an integer programming model and proved that it is an NP-hard problem. On this basis, we proposed two efficient algorithms, the PA and CA, to approximate the optimal solution. The evaluations indicate that our two methods perform better than the traditional MCF method at different evaluation settings for three metrics. The comprehensive evaluation results reveal that our method can effectively solve the proposed D-MCF problem.

To the best of our knowledge, our work is one of the first to address constraint routing for the uncertain multicast. From the theoretical perspective, the modeling of our setting introduces a new type of optimization problem. Flexibility in the choice of pseudo sources servers to expand the scope of multicast use in traditional networks and exploits the benefits of NFV.

Acknowledgment

This work was partially supported by the National Natural Science Foundation for Outstanding Excellent Young Scholars of China (No. 61422214), the National Natural Science Foundation of China (No. 61772544), the National Key Basic Research and Development (973) Program of China (No. 2014CB347800), the Hunan Provincial Natural Science Fund for Distinguished Young Scholars (No. 2016JJ1002), the Guangxi Cooperative Innovation Center of Cloud Computing and Big Data (Nos. YD16507 and YD17X11), and the NUDT Research Plan (No. ZK17-03-50).

References

- [1] A. Galis, S. Clayman, L. Mamas, J. R. Loyola, A. Manzalini, S. Kuklinski, J. Serrat, and T. Zahariadis, Softwarization of future networks and services-programmable enabled networks as next generation software defined networks, in *Proc. 2013 IEEE SDN for Future Networks and Services*, Trento, Italy, 2013, pp. 1–7.
- [2] Network functions virtualisation—Introductory white Paper, <https://portal.etsi.org/nfv/nfv-white-paper.pdf>, 2012.
- [3] B. Han, V. Gopalakrishnan, L. S. Ji, and S. Lee, Network function virtualization: Challenges and opportunities for innovations, *IEEE Commun. Mag.*, vol. 53, no. 2, pp. 90–97, 2015.
- [4] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, Network function virtualization: State-of-the-art and research challenges, *IEEE Commun. Surv. Tutor.*, vol. 18, no. 1, pp. 236–262, 2016.
- [5] S. Mehraghdam, M. Keller, and H. Karl, Specifying and placing chains of virtual network functions, in *Proc. 2014 IEEE 3rd International Conference on Cloud Networking*, Luxembourg, 2014, pp. 7–13.
- [6] Z. C. Xu, W. F. Liang, M. T. Huang, M. K. Jia, S. Guo, and A. Galis, Approximation and online algorithms for NFV-enabled multicasting in SDNs, in *Proc. 2017 IEEE 37th Int. Conf. Distributed Computing Systems*, Atlanta, GA, USA, 2017, pp. 625–634.
- [7] A. Mahimkar, Z. H. Ge, A. Shaikh, J. Wang, J. Yates, Y. Zhang, and Q. Zhao, Towards automated performance diagnosis in a large IPTV network, in *Proc. of ACM SIGCOMM 2009 Conf. Data Communication*, Barcelona, Spain, 2009, pp. 231–242.
- [8] D. K. Guo, J. Wu, Y. H. Liu, H. Jin, H. H. Chen, and T. Chen, Quasi-kautz digraphs for peerto-peer networks, *IEEE Trans. Parall. Distrib. Syst.*, vol. 22, no. 6, pp. 1042–1055, 2011.
- [9] X. O. Ding, H. Z. Wang, Y. T. Gao, J. Z. Li, and H. Gao, Efficient currency determination algorithms for dynamic data, *Tsinghua Sci. Technol.*, vol. 22, no. 3, pp. 227–242, 2017.
- [10] D. K. Guo, C. L. Li, J. Wu, and X. L. Zhou, DCube: A family of network structures for containerized data centers using dual-port servers, *Comput. Commun.*, vol. 53, pp. 13–25, 2014.
- [11] C. Diot, W. Dabbous, and J. Crowcroft, Multipoint communication: A survey of protocols, functions, and mechanisms, *IEEE J. Select. Areas Commun.*, vol. 15, no. 3, pp. 277–290, 1997.
- [12] T. Ballardie, P. Francis, and J. Crowcroft, Core Based Trees (CBT), in *Proc. Conf. on Communications Architectures, Protocols and Applications*, San Francisco, CA, USA, 1993, pp. 85–95.
- [13] K. Bharath-Kumar and J. Jaffe, Routing to multiple destinations in computer networks, *IEEE Trans. Commun.*, vol. 31, no. 3, pp. 343–351, 1983.
- [14] Z. Y. Hu, D. K. Guo, J. J. Xie, and B. B. Ren, Multicast routing with uncertain sources in software-defined network, in *Proc. 2016 IEEE/ACM 24th Int. Symp. Quality of Service (IWQoS)*, Beijing, China, 2016, pp. 1–6.
- [15] D. K. Guo, X. Q. Teng, Z. Y. Hu, J. J. Xie, and B. B. Ren, Source selection problem in multi-source multi-destination multicasting, *Comput. Netw.*, vol. 127, pp. 43–55, 2017.
- [16] B. B. Ren, D. K. Guo, J. J. Xie, W. X. Li, B. Yuan, and Y. F. Liu, The packing problem of uncertain multicasts, *Concurr. Comput.: Pract. Exp.*, vol. 29, no. 16, p. e3985, 2017.
- [17] F. Ergun, R. Sinha, and L. S. Zhang, An improved FPTAS for restricted shortest path, *Inf. Process. Lett.*, vol. 83, no. 5, pp. 287–291, 2002.

- [18] F. K. Hwang, D. S. Richards, and P. Winter, *The Steiner Tree Problem*. Elsevier, 1992.
- [19] S. E. Deering and D. R. Cheriton, Multicast routing in datagram internetworks and extended LANs, *ACM Trans. Comput. Syst.*, vol. 8, no. 2, pp. 85–110, 1990.
- [20] M. M. Wang, J. W. Liu, J. Mao, H. S. Cheng, J. Chen, and C. Qi, RouteGuardian: Constructing secure routing paths in software-defined networking, *Tsinghua Sci. Technol.*, vol. 22, no. 4, pp. 400–412, 2017.
- [21] S. H. Shen, L. H. Huang, D. N. Yang, and W. T. Chen, Reliable multicast routing for software-defined networks, in *Proc. 2015 IEEE Conf. Computer Communications (INFOCOM)*, Hong Kong, China, 2015, pp. 181–189.
- [22] S. Q. Zhang, Q. Zhang, H. Bannazadeh, and A. Leon-Garcia, Network function virtualization enabled multicast routing on SDN, in *Proc. 2015 IEEE Int. Conf. Communications (ICC)*, London, UK, 2015, pp. 5595–5601.
- [23] J. J. Kuo, S. H. Shen, M. H. Yang, D. N. Yang, M. J. Tsai, and W. T. Chen, Service overlay forest embedding for software-defined cloud networks, in *Proc. 2017 IEEE 37th Int. Conf. Distributed Computing Systems (ICDCS)*, Atlanta, GA, USA, 2017, pp. 720–730.
- [24] R. L. Graham and P. Hell, On the history of the minimum spanning tree problem, *IEEE Ann. Hist. Comput.*, vol. 7, no. 1, pp. 43–57, 1985.
- [25] R. W. Floyd, Algorithm 97: Shortest path, *Commun. ACM*, vol. 5, no. 6, p. 345, 1962.
- [26] H. F. Salama, D. S. Reeves, and Y. Viniotis, The delay-constrained minimum spanning tree problem, in *Proc. 2nd IEEE Symp. Computers and Communications*, Alexandria, Egypt, 1997, pp. 699–703.
- [27] V. P. Kompella, J. C. Pasquale, and G. C. Polyzos, Multicast routing for multimedia communication, *IEEE/ACM Trans. Network.*, vol. 1, no. 3, pp. 286–292, 1993.
- [28] M. Parsa, Q. Zhu, and J. J. Garcia-Luna-Aceves, An iterative algorithm for delay-constrained minimum-cost multicasting, *IEEE/ACM Trans. Network.*, vol. 6, no. 4, pp. 461–474, 1998.
- [29] Y. R. Chen, S. Radhakrishnan, S. Dhall, and S. Karabuk, On multi-stream multi-source multicast routing, *Comput. Netw.*, vol. 57, no. 15, pp. 2916–2930, 2013.
- [30] E. W. Dijkstra, A note on two problems in connexion with graphs, *Numerische Mathematic*, vol. 1, no. 1, pp. 269–271, 1959.



Bangbang Ren received the BS and MS degrees from National University of Defense Technology (NUDT) in 2015 and 2017, respectively. He is currently a PhD student in NUDT. His research interests include software-defined network, data center network, and network function virtualization.



Deke Guo received the BS degree from Beijing University of Aeronautics and Astronautics in 2001, and the PhD degree from National University of Defense Technology in 2008. He is currently a professor in the College of Systems Engineering at National University of Defense Technology. 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.



Geyao Cheng received the BS degree from National University of Defense Technology in 2016. She is currently working towards the MS degree at National University of Defense Technology. Her research interests include datacenter network, mobile computing, and crowdsensing.