Source Selection Problem in Multi-Source Multi-Destination Multicasting

Xiaoqiang Teng, Deke Guo*, Zhiyao Hu, Junjie Xie, Bangbang Ren

School of Information System and Management
National University of Defense Technology, Changsha Hunan 410073, China

Abstract

Multicast is designed to jointly deliver content from a single source to a set of destinations; hence, it can efficiently save the bandwidth consumption and reduce the load on the source. In many important applications, the appearance of multiple sources brings new opportunities and challenges to reduce the bandwidth consumption of a multicast transfer. In this paper, we focus on such type of multi-source multicast and construct an efficient routing forest with the minimum cost (MCF). MCF spans each destination by one and only one source, while minimizing the total cost (i.e. the weight sum of all links in one multicast routing) for delivering the same content from the source side to all destinations. Prior approaches for single source multicast do not exploit the opportunities of a collection of sources; hence, they remain inapplicable to the MCF problem. Actually, the MCF problem for a multi-source multicast is proved to be NP-hard. Therefore, we propose two $(2+\varepsilon)$-approximation methods, named P-MCF and E-MCF. We conduct experiments on our SDN testbed together with large-scale simulations under the random SDN network, regular SDN network and scale-free SDN network. All manifest that our MCF approach always occupies fewer network links and incurs less network cost for any multi-source multicast than the traditional Steiner minimum tree (SMT) of any related single source multicast, irrespective of the used network topology and the setting of multicast transfers.

Keywords: Multicast routing, multi-source multicast, the minimum-cost forest, Steiner tree

1. Introduction

Multicast is a natural approach to deliver the same content to a group of destinations not a single destination. It benefits in efficiently saving the bandwidth consumption and reducing the load on the sender, compared with the unicast. The root cause is that multicast can avoid unnecessary duplicated transmissions among a set of independent unicast paths towards destinations, after multiplexing a shared multicast tree. The tree spans the source and all destinations. Flows from the same source to destinations form a multicast transfer. Despite such bandwidth superiority, multicast in the Internet suffers many deploying obstacles during the preceding decades. Until recently, it achieves a few successful network-level deployments in IPTV networks [1], enterprise networks, and data center networks [2, 3].

Among existing multicast routing protocols, PIM is the most widely used one [2], which connects the source and destination via a shortest-path tree. An inherent drawback of this method is that those independent shortest paths, from each destination to the same receiver, lack opportunities to share more common links. Thus, the formed shortest-path tree fails to minimize the amount of utilized links; hence, multicast traffic along it cannot considerably save the bandwidth consumption. For this reason, many efforts focus on solving the Steiner minimum tree (SMT) [4], a NP-hard problem. It aims to minimize the total number of edges in a tree, spanning all members of any given multicast group. The SMT is not adopted by current networks, due to the huge computation overhead even under heuristic methods and the challenge of distributed deployment [5].

Such multicast methods, however, assume priori knowledge of multicast characteristics, i.e., the source of each multicast group is fixed in advance. We call this case as the *single-source multicast*. Unfortunately, in many cases, the characteristics of a multicast transfer are unknown a priori. That is, it is not necessary that the source of a multicast transfer has to be placed in a specific location as long as certain constraints are

*Deke Guo is the corresponding author.

satisfied. A major reason is the widely used content replica designs for improving the robustness and efficiency in various networks, such as the content distribution network [6], IPTV networks, and datacenter networks [7, 8]. When delivering a content file to multiple destinations, the source of such a multicast transfer can be any one replica in theory. This causes the *multi-source multicast*, due to the source flexibility.

In this paper, we focus on a new type of multicast transfer, the multi-source multicast. Given a set of destinations and a collection of sources, together with the network topology, the multi-source multicast problem aims to construct a minimum cost forest (MCF). It ensures that each destination connects to just one source through a path in the forest. The proposed multi-source multicast is a general scheme, which is equivalent to the single-source multicast in the special setting of a single source. The objective is to minimize the cost of the constructed forest. Compared with the SMT, finding the MCF for any multi-source multicast is more challenging, due to the flexible use of multiple sources and the impact on routing. We prove that the MCF problem is NP-hard. A potential approach is to divide multi-source multicast as a set of single-source multicasts, each with a distinct source. The SMT with the least cost among such multicasts just acts as the MCF of the multi-source multicast. This way, however, suffers from the complexity of solving a set of NP-hard SMT problems. Moreover, the MCF under non-single sources may cause less total cost than the picked best SMT with any single source. Thus, prior approaches, relying on traditional multicast, remain inapplicable to the multi-source multicast proposed in this paper.

To solve the MCF problem efficiently, we propose two $(2+\varepsilon)$-approximation methods, P-MCF and E-MCF. Such methods use a part of sources to establish routing paths towards all destinations. They can seek shared nodes among such paths to enhance the possibility of aggregating more links. Thus, they can reduce the number of employed links while ensuring that the delivered content can reach each destination within an acceptable delay. We conduct small-scale experiments on our testbed and large-scale simulations to compare the multi-source multicast and traditional multicast under the random network, regular network and scale-free network respectively. The evaluation results indicate that the MCF of multi-source multicast occupies less network links and incurs less network cost, irrespective of the used network topology.

In the remainder of this paper, we present our work as follows. In Section 2, we summarize related multicast methods. In Section 3, we present the problem formulation and model of the multi-source multicast. We design two efficient approximation algorithms in Section 4, and evaluate the performance of our algorithms through emulations and simulations in Section 5. Finally, we discuss and conclude this paper in Sections 6 and 7.

2. Related Work

2.1. Approximation methods for the SMT problem

Given any single-source multicast transfer, much research work focus on the construction of multicast trees to satisfy various constraints. The SMT problem formulates a common constraint on the multicast tree, i.e., the amount of occupied links for spanning all members of a multicast group. Many algorithms for the SMT problem have been proposed to approximate the optimal solution. A fast and effective algorithm achieves a $(2+\varepsilon)$-approximation, which approximates the SMT problem using a minimum spanning tree (MST) in [9]. Several methods, based on greedy strategies achieve better approximation ratio, such as 1.746 [10], 1.693 [11], 1.55 [12], 1.39 [13], but considerably incur high time complexity than the $(2+\varepsilon)$-approximation algorithm. The *k-restricted loss-contracting* algorithm achieves a 1.55-approximation. The time complexity, however, is too high due to many rounds of iterative computation, which is hard to be terminated within the acceptable time. Actually the round of iterations is unpredictable for large-scale networks.

Although the fast SMT algorithm does not achieve the optimal approximation ratio, it significantly saves more computation time than other algorithms. Hence, it is more suitable than other algorithms for large multicast groups and large-scale networks. A LP-based approximation algorithm presented in [13] can achieve a better approximation ratio of 1.39. It adopts iterative randomized rounding technique and is designed on the basis of *k*-restricted algorithm. Hence, it suffers high computation complexity. Such SMT algorithms, however, are still inapplicable to the MCF problem proposed in this paper, due to the introduction of multiple sources for each multicast group. Those multicast algorithms have not been adopted by current networks, due to the huge computation overhead and the challenge of distributed deployment [5]. Fortunately, the emergence of SDN makes it possible to realize novel multicast protocols.

2.2. The minimum-residual-bandwidth multicast

Given a multicast transfer, the desired multicast tree may need to satisfy other essential constraints, such as the end-to-end delay constraint, the reliability constraint, and the optimal residual bandwidth. The most related research work revolves around the MMForest algorithm [14]. For a collection of multi-source multicasts, MMForest aims to establish a multicast forest for each multi-source multicast to span its all destinations and sources, such that the shared network exhibits the maximal residual bandwidth. Actually, MMForest is designed based on the existing Widest-Path Forest algorithm, which prefers to select those links with higher residual capacity. It means that the knowledge about the collection of multi-source multicasts is known in advance and the construction order of their forests has serious impact on the final residual capacity. That is, all multi-source multicasts have to design their forests cooperatively.

The MMForest method focuses on optimizing the residual bandwidth under a set of multi-source multicast streams. On the contrary, the MCF problem proposed in this paper aims to tackle the widely used constraint, i.e., minimizing the amount of occupied links for each multicast stream. Thus, for a multi-source multicast, our MCF method establishes a forest with less amount of links than the MMForest method; hence, our MCF consumes less amount of total bandwidth than the MMForest. When many streams need to be delivered along different multi-source multicasts, more links would be saved by our MCF rather than the MMForest. Thus, our MCF method can further improve the level of residual bandwidth, and make the network support more multicast forests concurrently.

Moreover, MMForest remains inapplicable to our MCF problem. The root cause is that its basic idea is similar to the most straightforward multicast tree. MMForest makes each destination select one path with the largest residual bandwidth towards all sources, and then merges those paths selected by all destinations. Obviously, this method loses non-trivial opportunities to maximize the number of shared links among individual paths. MMForest is actually one type of the shortest-path forest; hence, it incurs higher total link cost than our MCF, due to the lack of shared links.

3. Problem Statement of MCF

We start with the important observation of multi-source multicast, and then present the problem statement of multi-source multicast. Finally, we give a mixed integer linear programming model for the multi-source multicast problem.

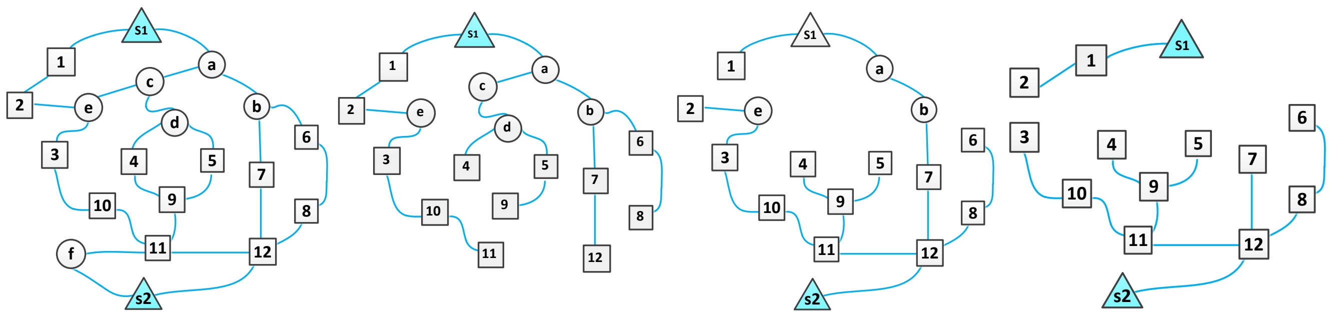
3.1. Observations

Multicast is a natural approach to deliver the same content to a group of destinations not a single destination. It benefits in efficiently saving the bandwidth consumption and reducing the load on the sender, compared with unicast. The total cost of a multicast tree, which spans all destinations and the single source, is widely used as the performance metric of a multicast. Without considering the diversity among network links, the cost of the multicast tree mostly depends on the amount of occupied links. To minimize the tree cost, given a multicast group many approximation methods [15] have been proposed to construct the SMT. Given the same set of destinations, the cost of a multicast group is sensitive to the selection of the source.

In many cases, due to the widely used designs of content replica, the source of a multicast transfer is not necessary to be placed in specific locations as long as certain constraints are satisfied. A major reason is the widely used design of content replica designs for improving the robustness and efficiency in various networks. Thus, any replica can participate in the multicast transmission by acting as the single source. The resultant different multicast transfers exhibit the diversity of tree cost, as shown in Figure 1(b) and Figure 1(c).

Figure 1 plots an illustrative example of the diversity of SMT for multicast with the same destinations $\{1, 2, ..., 12\}$. Triangle nodes denote all potential sources, square nodes denote the destinations, and cycle nodes denote intermediate forwarding nodes. Figure 1(b) and Figure 1(c) indicate two multicast trees with s_1 and s_2 as the single source, respectively. We can see that the multicast tree with the source s_1 is of cost 17, while the multicast tree with the source s_2 is of cost 16, even node $s1$ also participates in the multicast routing as a forwarding node.

Such facts motivate us to resolve the problem of multicast with multi-source sources, i.e., the multi-source multicast problem. It offers new opportunities to reduce more bandwidth consumption than existing multicast with a single source. A simple method for exploiting such opportunities is to calculate the SMTs for the multicast groups with the same destinations but varied sources by addressing a set of NP-Hard SMT problems. The SMT with the least tree cost should be the preferred one. For example, the multicast tree in Figure 1(c) is more desired one than that in Figure 1(b). However, we find that the picked best SMT with any single source may not be the optimal selection. A more efficient multicast may exist when multiple sources participate simultaneously. For example, multicast transmission in Figure 1(d) is just of cost 12 and outperforms



(a) Random network topology. (b) Multicast with a source s_1. (c) Multicast with a source s_2. (d) Multicast with sources s_1 and s_2.

Figure 1: An illustrative example of multi-source multicast with the same destination set $\{1, 2, ..., 12\}$ but different sources. The total cost of multicast transmission is 17, 16, and 12 in Figure 1(b), Figure 1(c), and Figure 1(d), respectively.

any individual SMT. Such a preferred design, however, cannot be achieved by prior approaches, relying on traditional multicast. That is, they remain inapplicable to the multi-source multicast problem proposed in this paper.

3.2. Problem statement

In the case of multiple sources, the traditional multicast problem can be naturally generalized as the multi-source multicast. More precisely, we consider a network $G(V,E)$, where V and E denote the set of nodes (switches and servers) and edges (links), respectively. Each edge $e \in E$ is associated with a cost and is normalized as one for the easy of presentation. We formalize the multi-source multicast problem in Definition 1.

Definition 1. *Given a destination set $D \in N$ and a source set $S \in N$, a multi-source multicast means to select a subset $S_0 \in S$ and partition set D into number of $|S_0|$ subsets D_i for $0 \le i < |S_0|$. Each source node in subset S_0 is responsible to deliver the same content to all destinations in one of $|S_0|$ partitioned subsets of set D. A constraint is that the intersection of any pair of D_i and D_j is empty for $0 \le i, j < |S_0|$. Thus, each destination is served by one and only one source in set S.*

The multi-source multicast is a general scheme of multicast communication. Actually, it is equivalent to the single-source multicast when the set S contains only one source. Even in the case of $|S|>1$, a multi-source multicast actually acts as a single-source multicast when all destinations employ any common source $s \in S$. It is clear that the usage strategy of such sources dominates the transmission cost of multi-source multicast. Therefore, we present the minimum cost forest (MCF) problem in Definition 2 to minimize the transmission cost of a multi-source multicast.

Definition 2. *Given a multi-source multicast, the problem of MCF is to find a forest spanning each destination $d \in D$ with only one source $s \in S$ such that the total cost of utilized edges in the forest is minimized. Note that each edge $e \in E$ is associated with a cost and is normalized as one for the easy of presentation.*

Accordingly, we can infer that the MCF does not necessarily contain all sources. Any pair of sources s_1 and s_2, however, should be separated if they were connected in the MCF. Otherwise, some destinations may reach more than one sources, increasing the cost of the resultant forest. The multicast topology in Figure 1(c) is not desirable since two sources s_1 and s_2 are linked through a path.

The multi-source multicast becomes the single-source multicast, when the source set contains only one element. Thus, the MCF problem is equivalent to constructing a SMT, which is NP-hard in a general graph. In general setting, the MCF problem involves more challenges than the SMT problem, due to the flexible use of multiple sources.

Theorem 1. *Given a multi-source multicast with the source set S and destination set D in a network $G(V,E)$, the problem of calculating its MCF is NP-Hard.*

Proof 1. *We prove the NP-hardness of MCF by giving a polynomial time reduction from SMT, a NP-hard problem, to MCF. We first examine the SMT problem of a multi-source multicast. It means to find a SMT for spanning all destinations and sources of that multi-source multicast. Clearly, the optimal solution of this SMT problem cannot be found within polynomial time. Note that $|S|^2$ source pairs exist in the optimal SMT and each source pair is connected by only one path. The path between any two sources has at least one redundant links.*

A link is redundant only if each destination still reaches at least one source after removing that link. For example, links $a \rightarrow b$ and $s_1 \rightarrow a$ in Figure 1(c) are redundant. Therefore, the SMT problem of a multi-source multicast would be reduced to the MCF problem, by removing all redundant links in the optimal SMT. The process of removing potential redundant links should be completed within polynomial time. The remaining links in the original optimal SMT form a forest F, where each destination connects to only one source. Thus, the MCF problem of multi-source multicast is NP-Hard; hence, Theorem 1 is proved.

A natural generalization of the Steiner Tree Problem is the Steiner Forest Problem (SFP) [16]. Given a collection of disjoint subsets, $S_1, S_2, ..., S_k$, of V in an undirected graph $G =< V, E >$, SFP aims to find a minimum cost subgraph in G, in which any two vertices lie in the same subset S_i are connected for $1 \leq i \leq k$. Note that for our MCF problem, it is not necessary that all sources will be employed and each destination just needs to connect to only one source in the resultant forest. It is not clear how to reasonably divide all destinations and sources of a multi-source multicast into a collection of disjoint subsets, such that each subset contains at least one source node and at least one destination node. Thus, our MCF problem is different from the SFP problem and the SFP problem cannot be simply reduced to our MCF problem. For this reason, we prefer to prove the NP-hardness of MCF by reducing the SMT problem to our MCF problem within polynomial time.

3.3. Mixed integer linear programming

We further present a Mixed Integer Linear Programming (MILP) formulation for the proposed MCF problem. It is practical to find the optimal solutions for small instances of the problem via any commercial MILP tool.

Let N_v denote the set of all neighbor nodes of node v in G, and u is in N_v if $e_{u,v}$ is an edge from u to v. Let S denote the source set of the multi-source multicast group, and D denote the destination set. The output minimal cost forest F needs to ensure that there is only one path in F from every node $d \in D$ to only one source $s \in S$. To achieve this goal, our problem includes the following binary decision variables. Let binary variable $\varpi_{d,s}$ denote whether a destination node d selects a source node s as the root node. That is, there is a path from d to s if $\varpi_{d,s}$=1. In this setting, let binary variable $\pi_{d,u,v}$ denote whether an edge $e_{u,v}$ is in the path from the destination d to the source s. Let binary variable $\varepsilon_{u,v}$ denote whether edge $e_{u,v}$ is in the output T. Intuitively, we should find the path from each destination d to just one source node s with $\varpi_{d,s}$=1. Thus, every edge $e_{u,v}$ in the path has $\pi_{d,u,v}$=1. The routing of the resultant forest with $\varepsilon_{u,v}$=1 for every edge $e_{u,v}$ in F can be achieved by the union of the paths from all destination nodes in D to at least one source node in S.

The objective function of the MCF problem is as follows.

$$\min \sum_{e_{u,v} \in E} c_{u,v} \times \varepsilon_{u,v},$$

where the weight $c_{u,v}$ denotes the cost of edge $e_{u,v}$.

To find $\varepsilon_{u,v}$, our MILP formulation includes the following constraints, which explicitly describe the routing principles for the multi-source multicast.

$$\sum_{s \in S} \varpi_{d,s} = 1, \forall d \in D \tag{1}$$

$$\sum_{v \in N_s} \pi_{d,s,v} = 1, \varpi_{d,s} = 1, \forall d \in D, \exists s \in S \tag{2}$$

$$\sum_{u \in N_d} \pi_{d,u,d} = 1, \varpi_{d,s} = 1, \forall d \in D, \exists s \in S \tag{3}$$

$$\sum_{v \in N_u} \pi_{d,u,v} = \sum_{v \in N_u} \pi_{d,v,u} = 1, \ \varpi_{d,s} = 1, \tag{4}$$

$$\forall d \in D, \exists s \in S, \forall u \in V, u \neq d, u \neq s$$

$$\pi_{d,u,v} \leq \varepsilon_{u,v}, \forall d \in D, \exists s \in S, \forall e_{u,v} \in E \tag{5}$$

For each destination node $d \in D$, the first constraint ensures that there exists only one source node $s \in S$ such that there exists at least one path from d to s in the output F. Formulas (2), (3), and (4) impose constraints on finding the path from every destination d in D to its source s. More precisely, given any destination node d, s is the source of the path towards d, and constraint (2) implies that only one edge $e_{s,v}$ from s to any neighbor node v needs to be selected with $\pi_{d,s,v} = 1$. At the same time, it ensures that there exists only one path from s to d and any pair of source nodes in the output F is isolated. On the other hand, every destination node d is the flow destination, and constraint (3) ensures that only one edge $e_{u,d}$ from any neighbor node u to d must be selected with $\pi_{d,u,d} = 1$. Furthermore, for any other node u in G, we can infer from constraint (4) that it is either located in the path from s to d or not. If u locates in the path, u has one incoming flow in the path with one binary variable $\pi_{d,v,u} = 1$ and one outgoing flow in the path with one binary variable $\pi_{d,u,v} = 1$. Otherwise, $\pi_{d,v,u}$ as well as $\pi_{d,u,v}$ are 0. Note that, according to the objective function, $\pi_{d,v,u} = 1$ is set for just one neighbor node v so as to minimize the cost of output forest F.

The last constraint desires to find the routing of the output F, i.e., $\varepsilon_{u,v}$. More specifically, $\varepsilon_{u,v}$ must be 1 if edge $e_{u,v}$ is in the path between at least one pair of source s and destination d, i.e., $\pi_{d,u,v} = 1$. The output forest F is the union of the paths from all destinations to their corresponding sources.

4. Efficient building methods of MCF

We start with designing a fundamental approximation method, P-MCF, to find the minimum-cost forest, and then present a more efficient method, E-MCF.

4.1. Primary approximation method

As aforementioned, the MCF problem for a multi-source multicast is NP-hard and cannot be solved in polynomial time. Thus, we focus on designing efficient methods to approximate the optimal forest [17] for the multi-source multicast.

A straightforward method is to treat a multi-source multicast as a collection of single-source multicasts, each with a different source. Accordingly, we select the SMT with the least cost as the MCF of the multi-source multicast. This method suffers the complexity of solving a set of NP-hard SMT problems. Moreover, the MCF involving multiple sources may cause less total cost than the picked best SMT, as discussed in Section 3.1. Thus, previous approaches for traditional multicast remain inapplicable to the proposed multi-source multicast. For this reason, we design an approximation MCF method, called P-MCF, as illustrated in Algorithm 1.

Given a network, its undirected graph model is $G{=}(V,E)$. For a multi-source multicast with the source set S and destination set D, we derive the complete graph $G_1{=}(V_1,E_1)$ from the original graph D and two nodes sets, S and D. The node set V_1 is the union of the source set S and destination set D. According to the definition of the complete graph, there exists an edge between any pair of nodes in V_1. For every edge $\{v_i,v_j\}\in E_1$, $d(\{v_i,v_j\})$ denotes its cost, i.e., the length of the shortest path from node v_i to node v_j in the original graph G. That is, each edge in G_1 refers a shortest path in G.

Algorithm 1 P-MCF method

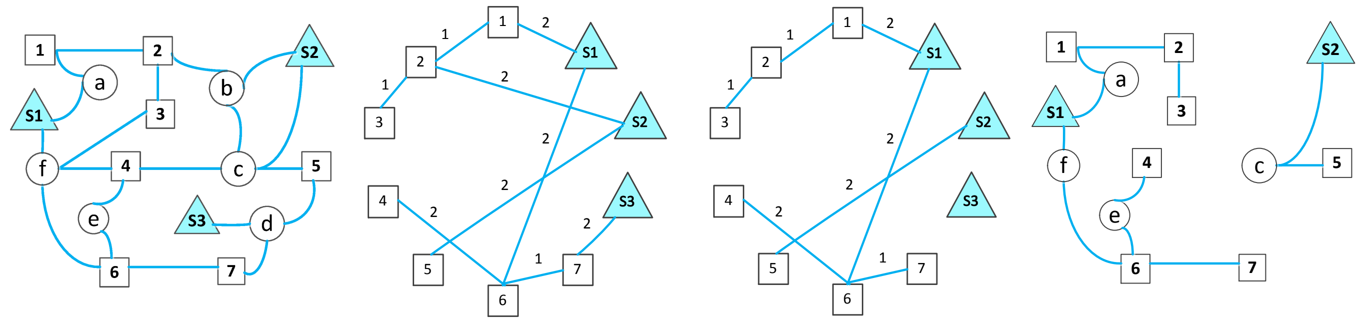
```
Require: An undirected graph $G = (V,E)$, the set of destinations $D$, and the set of sources $S$.
Ensure: A minimum cost forest $F$.
1: Construct the complete graph $G_1 = (V_1,E_1)$ from $G$ and two sets, $D$ and $S$.
2: Find a minimum spanning tree $T$ for the graph $G_1$.
3: Construct the subgraph $T_1$ from $T$ by deleting some edges, if necessary, so as to isolate all source nodes.
4: Construct the subgraph $F$ of $G$ by replacing each edge in $T_1$ by its corresponding shortest path in $G$. Note that those isolated source nodes in $T_1$ are also removed.
```

We then construct a minimum spanning tree T for the graph G_1 such that any pair of nodes in V is connected through only one path. Accordingly, any pair of source nodes in S is also connected in the tree T. However, any two sources should be separated if they are employed in the MCF, as discussed in Sections 3.2 and 3.3. For this cause, any pair of connected sources in T need to be separated by removing the maximum-cost edge along their only path in T. If there are several such edges in the path, it is reasonable to pick an arbitrary one. The challenge issue is the processing order of those connected sources in the tree T. Thus, we record the maximum-cost edge in the path of each pair of connected source nodes, and sort such edges in the descending order of their costs. We delete the first edge from T and update the sorted edges. Cutting edges is finished until the updated T_1 no longer contains any connected sources. Finally, we can build a MCF for the multi-source multicast by removing those isolated nodes in T_1 and replacing each edge in T_1 by its corresponding shortest path in G.

Figure 2(a) illustrate an example of multi-source multicast in a small scale network. Let $S{=}\{s_1,s_2,s_3\}$ be the set of source nodes and $D{=}\{1,2,3,4,5,6,7\}$ be the set of destination nodes. Figure 2(b) shows a minimal spanning tree T of the complete graph consisting of all sources and destinations. Figure 2(c) shows the minimum spanning tree T_1 after removing edges $\{s_2,2\}$ and $\{s_3,7\}$ from T. Note that we may remove different edges to ensure that any two sources are disconnected. For example, it's reasonable to delete edges $\{s_1,6\}$ and $\{s_1,1\}$.

Theorem 2. *Given any multi-source multicast in a network G, the output MCF of our P-MCF method obtains an approximation ratio of* $(2+\varepsilon)$.

Proof 2. *Let a node set P denote the union of the source set S and the destination set D. Let $T(P)$ denote the generated MCF of the multi-source multicast through our P-MCF method. $T(P)$ utilizes fewer links than the minimum spanning tree $mst(P)$, since $T(P)$ is derived from $mst(P)$ after removing some edges. We can infer that the cost of $T(P)$ is less than that of $mst(P)$, i.e., $|T(P)|\le|mst(P)|$. Let $smt(P)$ denote a Steiner minimum tree spanning all nodes in P, which uses fewer links than $mst(P)$ because of high link overlap rate [12]. There exists an Euler tour of $smt(P)$, called T_E, which passes each edge in $smt(P)$ two times. The distance*



(a) A multi-source multicast in a small-scale random network (b) T, the minimal spanning tree of G_1 (c) T_1, where sources are disconnected (d) The MCF of cost 11

Figure 2: An illustrative example of our P-MCF method, which constructs a MCF for the multi-source multicast in Figure 2(a).

function of the Steiner tree problem conforms to the triangle inequality. The cost of any Euler tour is less than that of the related minimum spanning tree. Hence, $2|smt(P)| \geq |T_E| \geq |mst(P)| \geq |T(P)|$ *[18]. Let opt be the optimal MCF of the multi-source multicast. If additional d edges are added into opt to connect each trees as a new tree* T', T' *utilizes number of* $|opt|+d$ *links, which is more than* $smt(P)$. *Thus, we have* $2(opt+d) \geq 2smt(P) \geq mst(P) \geq T(P)$. *The approximation ratio of our P-MCF is* $\frac{T(P)}{opt} \leq 2+\frac{2d}{opt}$. *Specially, the ratio of d to opt is* ε *and is considerably less than 1 in most real networks. Thus, Theorem 2 is proved.*

Time Complexity. We analyze the time complexity of our P-MCF method in the following way. At the beginning, the number of $(|S|+|D|)^2$ shortest paths have to be calculated for achieving the complete graph G_1. The time complexity of calculating one shortest path is $O(V^2)$ [19]. Thus, the time complexity of the complete graph is $O((|S|+|D|)^2 \times |V|^2)$. The time complexity of finding the minimal spanning tree T from G_1 is $O((|S|+|D|)^2)$ [20]. The time complexity of finding and removing the maximum cost edge in paths between sources will cost at most $O(|S|)$. Thus, the total time complexity of the proposed P-MCF method is $O((|S|+|D|)^2 \times |V|^2)$.

4.2. Enhanced approximation method

In this section, we design an enhanced approximation method, called E-MCF, to build the MCF for any multi-source multicast. The basic idea is originated from the observation about the shared nodes in MCF. The major difference between the E-MCF and P-MCF methods is the construction of the complete graph G_1 from G and the multi-source multicast. The two methods share the same process after deriving out the complete graph G_1.

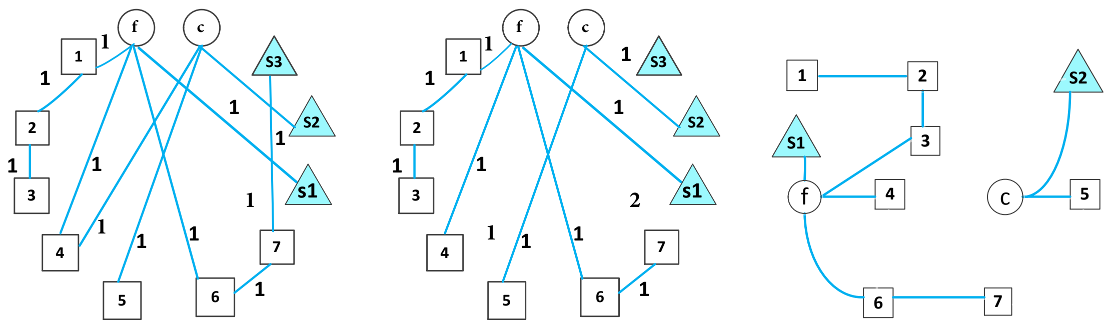
Multicast destinations share one source node; hence, their routing paths towards the source node may overlap. Those overlapped links have common nodes, which are called shared nodes. Those shared nodes frequently appear in the shortest paths from sources to destinations.

In the E-MCF method, the complete graph G_1 contains not only nodes in sets D and S but also some shared nodes. Those shared nodes are helpful to find which links are more beneficial if they are involved in the constructed minimum spanning tree T. Without such shared nodes, T usually involves more edges, which can't be aggregated along paths from destinations to such sources.

4.2.1. Observation of shared nodes.

Note that, in the case of our P-MCF method, all nodes in the complete graph G_1 are just those source nodes and destination nodes. Thus, it's hard to distinguish which edges/paths in G_1 are more possible to aggregate if they appear in the minimum spanning tree T. We find that if some shared nodes are appended into the complete graph G_1, the formed minimal spanning tree and the final MCF are dramatically changed since more paths can be aggregated. As a result, the total transmission cost of a multi-source multicast can be considerably reduced. For example, as shown in Figure 3(a), we add shared nodes f and c into G_1. Figure 3(b) reports the resultant minimum spanning tree of G_1, where more edges can be aggregated. As shown in Figure 3(c), the output MCF for the same multi-source multicast is of cost 9, which is much less than that of formed MCF in Figure 2(d).

Accordingly, we aim to introduce some shared nodes into the complete graph G_1. Thus, more edges with high overlap rate will be added into the complete graph. Finally, the output MCF can save more links and reduce



(a) T, the minimal spanning tree of G_1 (b) T_1, where sources are disconnected (c) The MCF of cost 9

Figure 3: An illustrative example of our E-MCF method, which constructs a MCF for the multi-source multicast in Figure 2(a).

its total cost.

4.2.2. Identification of shared nodes.

As aforementioned, the introduction of shared nodes in E-MCF method can considerably reduce the total cost of resultant forest. However, it is non-trivial to identify and select shared nodes in graph G. To address this problem, we propose Algorithm 2 as follows.

In Algorithm 2, the function *ShortestPath* calculates the shortest path between any two nodes. Note that Algorithm 2 calculate those shortest paths between not only any pair of source and destination but also any pair of two destinations. It, however, does not consider the shortest path between any pair of two sources since those sources selected by the MCF should be disconnected. Accordingly, we extract intermediate nodes in each shortest path and count the times it is shared with other shortest paths via the function *IntermediateNode*. The function $IsConnected(u,v)$ is used to judge whether node u is connected with node v.

As an example, we compare Figure 2(b) with Figure 3(a). It is clear that nodes f and c are two branching point; hence, they are more feasible to aggregate edges if they are included in the complete graph G_1. We can see shared nodes should be connected to many other nodes such that flows from such nodes will be immediately aggregated at a shared node.

4.2.3. Analysis of E-MCF.

The major difference between P-MCF and E-MCF is the design of the complete graph G_1. As shown in Figure 3, such a novel design is very effective to reduce the transmission cost of the same multi-source multicast. Figure 3(a) shows the minimum spanning tree T of the new complete graph G_1 with the node set $\{1,2,3,4,5,6,7,s1,s2,s3,f,c\}$. Notably, two shared nodes f and c are also included in G_1. To disconnect each pair of sources, two edges $c{\rightarrow}4$ and $s_3{\rightarrow}7$ are removed from T as shown in Figure 3(b). After replacing each edge in T_1 with the corresponding shortest path in G, we achieve the output MCF of cost 9, as shown in Figure 3(c).

As illustrative an example in Figure 3(b), shared node f divides the edge in Figure 2(c) from node 3 to s_1 into two edges $\{3,f\}$ and $\{f,s_1\}$. The path from 6 to s_2 is also divided by f. Some aggregated links which are repeatedly used can be saved. Consequently, the total weight decreases.

Theorem 3. *Given any multi-source multicast in a network $G=(V,E)$, E-MCF is more effective than P-MCF.*

Proof 3. *Recall that the P-MCF method first constructs the complete graph G_1 of all members of the multi-source multicast, and then derives a minimum spanning tree T_1 from G_1. We collect all intermediate nodes located on the shortest paths. The most frequent intermediate nodes among all shortest path are selected as shared nodes. In the E-MCF method, those shared nodes will be added into G_1. Thus, a new complete graph G_2 and a minimum spanning tree T_2 are achieved. We will prove that T_2 might exhibit less total weight than T_1, due to the introduction of those shared nodes.*

If adding a shared node v, some edges in T_1 might change according to two situations as follows. For each node u in T_1, T_2 will consider whether the new edge $e_{u,v}$ should be adopted.

- *For an edge $e_{u,w}$ in T_1, if the related shortest path in G traverses the shared node v, T_2 prefers to replace the edge $e_{u,w}$ with edges $e_{u,v}$ and $e_{v,w}$. This will not increase the total weight of T_2 since the weight of $e_{u,w}$ is equal to that of $e_{u,v}$ plus that of $e_{v,w}$. If other edges of node u in T_1, such as e_{u,w_1}*

Algorithm 2 Identification of shared nodes

```
Require: An undirected graph $G = (V, E)$, the set of destinations $D$, and the set of sources $S$.
Ensure: A set of shared nodes $P'$.
1: $P' \leftarrow \emptyset, L \leftarrow \emptyset$;
2: for $\forall s_i \in S$ and $\forall d_j \in D$ do
3:     $L \leftarrow ShortestPath(s_i, d_j)$
4: end for
5: for $\forall d_i \in D$ and $\forall d_j \in D$ and $d_i \neq d_j$ do
6:     $L \leftarrow ShortestPath(d_i, d_j)$
7: end for
8: for $\forall l_i \in L$ do
9:     $Candidate \leftarrow IntermediateNode(l_i)$
10: end for
11: for $\forall u \in Candidate$ do
12:     $NumberofNeighbor \leftarrow 0$
13:     for $\forall v \in Candidate$, $u \neq v$ do
14:         if $IsConnected(u, v)$ then
15:             $NumberofNeighbor \leftarrow NumberofNeighbor + 1$
16:         end if
17:     end for
18:     if $NumberofNeighbor > 3$ then
19:         $P' \leftarrow u$
20:     end if
21: end for
```

and e_{u,w_2}, exhibit the same property of $e_{u,w}$, T_2 will update edge e_{u,w_1} with edges $e_{u,v}$ and e_{v,w_1}, and update e_{u,w_2} with edges $e_{u,v}$ and e_{v,w_2}. Thus, the common edge $e_{u,v}$ is shared repeatedly and can be aggregated. This fact will reduce the total weight of T_2.

- *Otherwise, let $e_{u,w}$ be the edge with the highest weight among all adjacent edges of u in T_1. If edge $e_{u,w}$ has higher weight than edge $e_{u,v}$, T_2 will adopt edge $e_{u,v}$ rather than edge $e_{u,w}$. The total weight of T_2 will be decreased by the weight of $e_{u,w}$ minus that of $e_{u,v}$. On the contrary, if the weight of edge $e_{u,w}$ is less than that of $e_{u,v}$, T_2 still utilizes edge $e_{u,w}$ even a shared node v is used.*

In summary, the total weight of T_2 from E-MCF is less than that of T_1 from P-MCF. We have proved that P-MCF is $(2+\varepsilon)$-approximation in Theorem 2. Thus, E-MCF will achieve better approximation ratio than P-MCF.

Time complexity. We analyze the time complexity of the E-MCF method in the following way. At the beginning, we need to calculate $(|S||D| + \frac{|D||D-1|}{2})$ shortest paths from all destinations to all source and among destinations. The time complexity of calculating one shortest path is $O(|V|^2)$ [20]; hence, the time complexity of finding the complete graph is $O((|S||D| + \frac{|D||D-1|}{2}) \times |V|^2)$. The time complexity of finding all shared nodes is $O(|V|)$. The time complexity of forming the minimum spanning tree T from G_1 is $O((|S| + |D| + |SharedNode|^2)$, where $|SharedNode|$ is the number of shared nodes. At last, some edges are removed by discovering and cutting the highest-weight edge in the path between any two sources in T. If each edge in T is checked, its complexity is $O((|S| + |D| + |SharedNode| - 1)$. In total, the complexity of our E-MCF method is $O((|S||D| + \frac{|D||D-1|}{2}) \times |V|^2)$.

5. Performance Evaluation

We begin with the implementation method of multi-source multicast in actual networks. We then measure the performance of multi-source multicast based on small-scale experiments. Finally, we conduct large-scale simulations to evaluate the multi-source multicast under different network settings. In this section, we also use the term P-MCF and E-MCF to refer the forests resulting from the P-MCF and E-MCF methods, respectively.

5.1. Implementation of the multi-source multicast in a SDN testbed

We execute the SMT method for traditional multicast and our methods for multi-source multicast in a real testbed of software-defined networks (SDN) [21, 22]. SDN is a flexible architecture for the network resource management and network application innovation. After decoupling the control plane from the data plane, the controller is responsible for a programmable control plane. It collects users' network requirements and jointly manages the underlying physical network in a centralized way. The response to users' requirements will be translated into a series of control policies, which are delivered to involved network devices for updating their flow tables. Accordingly, SDN provides new opportunity for deploying various flexible protocols, such as the single-source multicast and the multi-source multicast. We report the data plane and the control plane of our testbed as follows.

Data Plane Our testbed includes 16 ONetSwitch20 Openflow switches [23], each of which has four 1G data ports and one 1G management port. Each ONetSwitch20 switch consists of the Zynq SoC and other components, so as to enable the line-rate packet switching. We select ONetSwitch20 as our data plane since it can flexibly support the management and customization of the flow table. Such ONetSwitch20 switches are connected randomly and the available data ports of each switch are reserved to connect computing nodes, PcDuino nodes. Figure 2(a) plots the topology of our testbed, where all computing nodes are omitted. In this section, we vary the number of source and destination nodes, which connect to different switches. When we mention the source and destination nodes of a multicast or a multi-source multicast, we just replace each multicast member with the related switch it appends in this section.

Control Plane We deploy the RYU controller [24] in the testbed since it is flexible to support our evaluation. We realize three multicast applications, including the SMT, P-MCF, and E-MCF, on the RYU controller. The first task of the control plane is to generate and maintain the SMT for any given multicast and the MCF for any given multi-source multicast. The second task is to deploy the generated SMT and MCF into the physical SDN network, according to the Openflow specification [25]. The packet forwarding procedure should be transparent such that switches need not to be aware whether a packet belongs to a multicast session or not. One simple method is to configure special addresses for multicast sessions and configure multicast flow entries, each of which contains a set of output instructions. Once a multicast flow entry in an involved switch matches packets with the address, the packet will be replicated and forwarded out from a given set of ports. Our flow table is shown in Table 1.

Table 1: Multicast Flow Entry

Match	Actions
multicast address	output:port 1 output:port 2

5.2. *Evaluation based on small-scale experiments*

In this section, we evaluate not only our P-MCF and E-MCF methods for multi-source multicast, but also the widely used SMT method for traditional single-source multicast on our testbed. The testbed keeps the topology among 16 SDN switches, but changes the configurations of multi-source multicasts.

Given each setting of a multi-source multicast, we first construct the desired forest using our P-MCF and E-MCF methods. At the same time, we derive a single-source multicast by randomly picking a source from all potential sources. The SMT method establishes the minimum-cost tree for spanning the picked source and all destinations. For any multi-source multicast, we evaluate the resultant SMT or MCF using two performance metrics, including the total number of links and the flow completion time (FCT). FCT means the duration that one destination completely receives one multicast stream from a source. The longest, shortest, and average FCTs indicate the maximum, minimum, and average completion time among all flows in a multi-source multicast, respectively. Note that each source delivers 10MB data towards related destinations.

5.2.1. *Impact of the number of sources*

We randomly fix 7 destination nodes while vary the number of sources from 2 to 6. Figure 4 reports the evaluation results about the three methods, in terms of four metrics under varied number of sources.

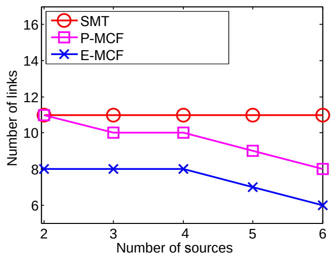
Figure 4(a) indicates that E-MCF always incurs lower cost than the SMT, when the number of sources increases from 2 to 6. This evidence proves the feasibility and effectiveness of the proposed multi-source multicast in this paper. Such benefit is more notable when increasing number of sources are utilized. Additionally, the E-MCF also outperforms the P-MCF, irrespective of the number of sources.

Figure 4(b) and Figure 4(c) show that the E-MCF and P-MCF of the multi-source multicast outperform the SMT of the traditional multicast, in terms of the shortest and the average FCTs. Additionally, we find that the shortest and the average FCTs of P-MCF are less than that of E-MCF. The reason is that the P-MCF prefers to use the shortest path; hence, it contains the most amount of shortest paths from destinations to their nearest sources among three building algorithms. The longest FCT records the time duration until the last destination receives the data from sources; hence, it dominates the users' experience of the multicast routing. In our small-scale testbed, the three methods exhibit similar value of the longest FCT, as shown in Figure 4(d).

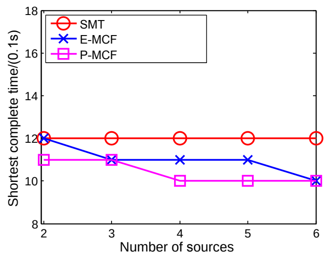
In experiments, no matter a multi-source multicast contains how many sources, we consider a dedicated multicast with one given source. Accordingly, the four performance metrics of the resultant SMT do not vary with the increasing number of sources. When a multi-source multicast employs more sources, our E-MCF and P-MCF perform better than the SMT. Among the three methods, the E-MCF is the desired one since it incurs the least cost and lower FCT.

5.2.2. *Impact of the number of destinations*

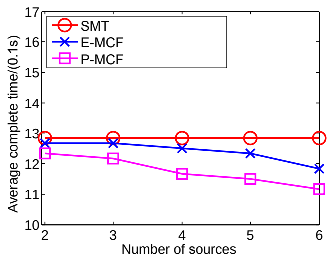
To evaluate the impact of the number of destinations, we randomly fix two source nodes and vary the number of destinations from 3 to 9. Figure 5 reports the results



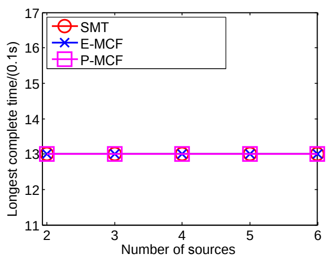
(a) Total link cost.



(b) Shortest FCT.

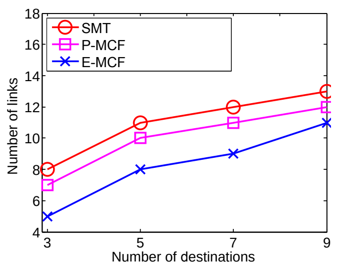


(c) Average FCT.

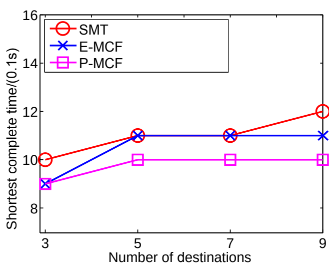


(d) Longest FCT.

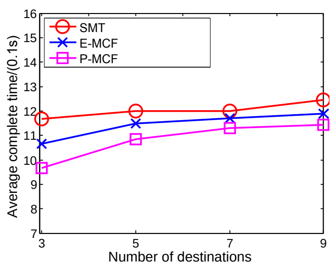
Figure 4: Impact on total link cost and FCTs of multi-source multicast when varying the numbers of sources from 2 to 6 in the testbed, while 7 destinations are fixed.



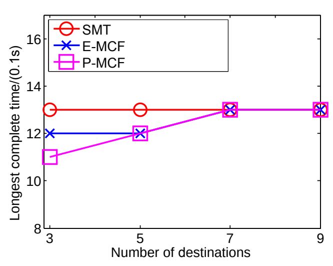
(a) Total link cost.



(b) Shortest FCT.



(c) Average FCT.



(d) Longest FCT.

Figure 5: Impact on total link cost and FCTs of multi-source multicast with varied number of destinations in the testbed, while two source nodes are fixed.

of the three methods, in terms of four metrics under varied number of destination nodes.

Figure 5(a) shows that total cost of the three methods increases along with increasing destinations. However, E-MCF and P-MCF always incur lower cost than the SMT. Figure 5(b), Figure 5(c) and Figure 5(d) report the shortest, average, and the largest FCTs under varied multicast methods. We can see that the P-MCF method achieves the best performance in terms of the three FCT metrics, irrespective of the number of destinations. On the other hand, the E-MCF method incurs the lowest cost among the three methods as shown in Figure 5(a), and achieves acceptable FCTs.

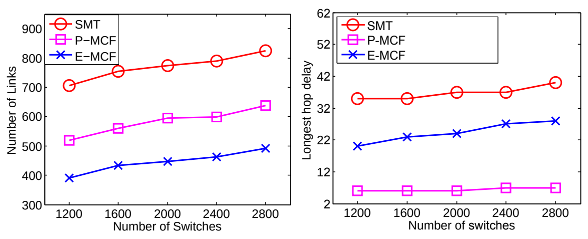
5.3. Evaluation based on large-scale simulations

The above small-scale experimental results have demonstrated the benefits of our multi-source multicast and proved that the E-MCF achieves the best performance than P-MCF and SMT. It is necessary to evaluate the performance of our methods under large-scale networks, where the multi-source multicast involves more sources and destinations. Thus, we further conduct large-scale simulations to evaluate the performance of our MCF problem in random networks, regular networks, and scale-free networks. We focus on two performance metrics, including the total link cost and the longest hop delay of each multicast routing structure. To ease the presentation, we use the largest hop length between a pair of source and destination to refer the longest hop delay.

5.3.1. Simulation settings of large-scale networks

We first realize three types of network topologies, including random networks, regular networks, and scale-free networks, as the adjacent matrix in JAVA language. A random network is obtained by starting with a set of n isolated vertices and adding successive edges between them at random. Actually, regular and scale-free networks originate from the regular and scale-free graphs in graph theory. A regular graph means that all nodes in the graph have the same degree. However, the degree distribution of nodes in a scale-free network obeys to the power law; hence, small proportion of nodes has most connections but large proportion of nodes does not [**? ?**].

In this way, we implement our two MCF algorithms as well as the SMT algorithm. In our simulations, we also need to derive a related multicast for any multi-source multicast before comparing our MCF methods and prior SMT method. Thus, we initialize a multi-



(a) Total link cost. (b) The longest hop delay.

Figure 6: The changing trends of the cost and hop delay under random networks covering 1200 to 2800 switches. The multi-source multicast has 10 sources and 300 destinations.

source multicast as a single-source multicast by randomly picking a source from all potential sources. The SMT method then establishes the minimum-cost tree for spanning the picked source and all destinations.

Moreover, we measure the performance of the three methods under random networks, regular networks and scale-free networks. In the case of each network topology, we change the network scale from 1200 switches to 2800 switches, the number of sources from 3 to 21, and the number of destinations from 500 to 1000. The resultant network consists of 4800 to 10000 hosts if each switch connects with four hosts.

5.3.2. Evaluation under random networks

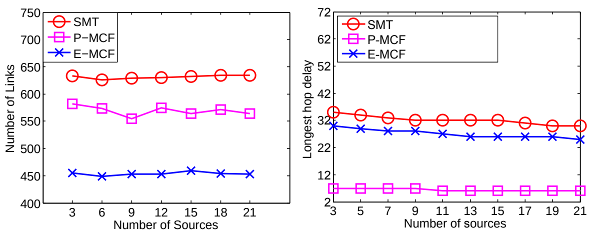
We first measure the performance of E-MCF, P-MCF, and SMT in large-scale random networks. We start with multi-source multicasts with 10 sources and 300 destinations in varying scale random networks.

Impact of the network scale. Figure 6 reports the evaluation results of the three methods, when the number of switches increases in a random network. We can infer from Figure 6(a) that the SMT of traditional multicast causes the largest link cost among the three methods. The benefits of our P-MCF and E-MCF methods for the multi-source multicast are more noticeable when the network scale becomes 2800. More precisely, our E-MCF method can save the total cost of traditional SMT by 39.23%. Figure 6(b) shows that our P-MCF is the best one among three methods in terms of the longest hop delay, since any destination just selects the shortest path to its nearest source.

Although the P-MCF outperforms the E-MCF in terms of the longest hop delay, the E-MCF and SMT always occupy the least and most number of links, respectively. In summary, our E-MCF method always incurs the least cost than others, irrespective of the network scale.

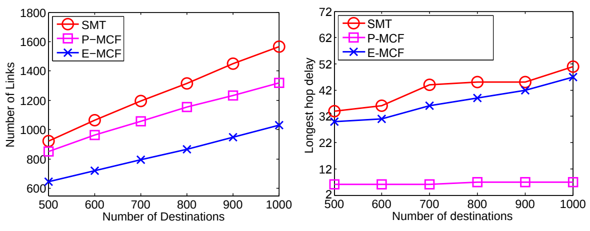
Impact of the number of sources. As shown in Figure 7(a), with the increasing of the number of sources, the number of links based on the traditional SMT remains relatively stable because it just randomly utilizes one given source. Meanwhile, the E-MCF always incurs fewer links than the P-MCF and the SMT. Additionally, newly added sources will not considerably reduce the link costs of E-MCF method. We find the link cost of each building method fluctuates along the increasing number of sources. The cost of SMT may grow when the number of sources increases. The root cause is that the single source for establishing the SMT is selected randomly. The average cost of resultant SMT, however, is relatively stable. Despite the fact that the cost of the resultant P-MCF forest also faces fluctuation, it exhibits a decreasing trend along with the increasing number of sources. We can see from the evaluation results that three data copies (three sources) are somehow efficient and sufficient for multi-source multicast in practice. Figure 7(b) shows that the P-MCF causes the least hop delay than other two methods. It means that destinations can early receive data from sources in P-MCF than in E-MCF. In conclusion, E-MCF cannot only reduce the hop delay of the SMT but also cause the least link cost among the three methods.

Impact of the number of destinations. Figure 8 in-



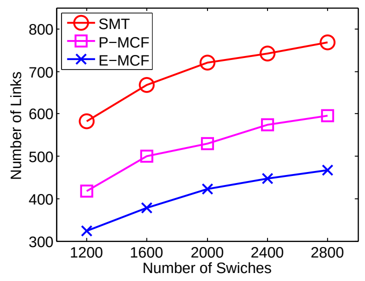
(a) Total link cost. (b) The longest hop delay.

Figure 7: The impact of the number of sources on the link cost and the hop delay under a random network of 2000 switches. We set 300 fixed destinations but variable sources for multi-source multicasts.

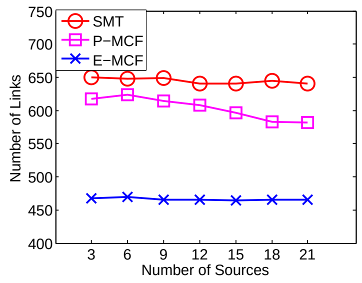


(a) Total link cost. (b) The longest hop delay.

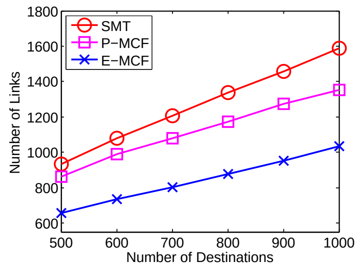
Figure 8: The impact of the number of destinations on the link cost and the hop delay under a random network with 2000 switches. We set 10 fixed sources but select variable quantity of destinations for multi-source multicasts.



(a) Impact of the network scale.

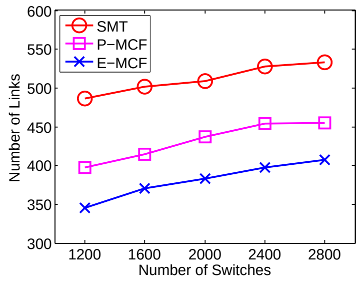


(b) Impact of number of sources.

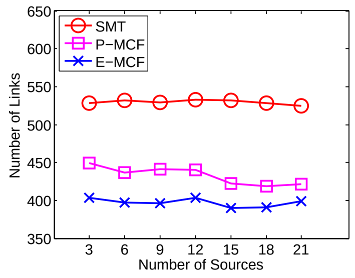


(c) Impact of number of destinations.

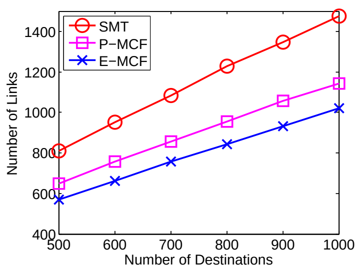
Figure 9: The changing trends of the link cost under different environments in regular networks.



(a) Impact of the network scale.



(b) Impact of number of sources.



(c) Impact of number of destinations.

Figure 10: The changing trends of the link cost under different environments in scale-free networks.

dicates that the addition of more destinations will naturally increase the link cost and the longest hop delay under the three multicast methods. Additionally, our two methods P-MCF and E-MCF always outperform the traditional SMT method in terms of the two performance metrics. In summary, our E-MCF causes the lowest link cost while achieves the acceptable hop delay.

5.3.3. Evaluation under regular networks

In this section, we estimate the performance of the three multicast methods under the scale-increasing regular networks. The regular network means that all nodes have the same value of node degree [26]. Our regular networks contain 1200 to 2800 switches. Under each setting of network scale, we set the multi-source multicast with 10 sources and 300 destinations. We then measure the total link cost of our P-MCF, E-MCF, and the traditional SMT. As shown in Figure 9(a), it is clear that the total link cost caused by the three methods increases along with the increasing network scale. Additionally, the E-MCF and P-MCF always cause less cost than the traditional SMT, irrespective of the network scale. Note that our E-MCF is the best one among those methods in terms of the total link cost.

Secondly, we select 300 destinations in a regular network with 2000 switches. In this setting, we change the number of sources from 3 to 21 and evaluate the performance of multi-source multicast. We can see from Figure 9(b) that the E-MCF with multiple sources is obviously better than the traditional SMT with only one source. The impact of more sources on the total link cost becomes weak when the number of sources exceeds 3. That is, the total link cost in the E-MCF slowly decreases when the number of sources increases. In this evaluation setting, the E-MCF still occupies fewer links than the P-MCF and the traditional SMT.

Thirdly, we select 10 sources under the regular network which includes 2000 switches. The number of destinations for multi-source multicast changes from 500 to 1000. Figure 9(c) reports the evaluation results of three methods under the varied number of destinations. Similarly, our E-MCF always uses fewer links than other two methods, irrespective of the number of destinations.

5.3.4. Evaluation under scale-free networks

In this section, we conduct simulations under scale-free networks because the distribution of nodes in the Internet follows the power laws. A network whose vertex connectivity follows a scale-free power-law distribution is known as the scale-free network [?].

Firstly, we construct scale-free networks with varying scales. The number of switches varies from 1200 to 2800. We randomly choose 10 sources and 300 destinations under each setting of network scale. We then record the total link cost of our P-MCF, E-MCF, and the traditional SMT. As shown in Figure 10(a), it is clear that the total link cost of each method grows up along with the increasing network scale. It's noteworthy that the link cost of the SMT increases rapidly because it always utilizes a single source. The increased link cost for our E-MCF and P-MCF is slow. It is clear that our E-MCF incurs the least number of links.

In a scale-free network with 2000 switches and 300 destinations, we change the number of sources from 3 to 21. We can see from Figure 10(b) that the SMT multicast is not significantly affected by the number of sources. The link cost of P-MCF becomes less when the number of available sources increases. The impact of more sources on the E-MCF algorithm becomes weak when the number of sources exceeds 3. Additionally, our E-MCF method can incur the least number of links than other two methods, irrespective of the number of sources.

In order to evaluate the impact of the number of destinations, we randomly choose 10 nodes as sources under scale-free networks with 2000 switches. The number of destinations changes from 500 to 1000. As shown in Figure 10(c), three multicasts need more links owing to the increase of receivers, and our E-MCF multicast always occupys fewer links than the P-MCF method and the SMT algorithm for a single source.

In conclusion, our MCF methods are adaptive to different network topologies. Our E-MCF method always incurs fewer links than our P-MCF method and the traditional SMT multicast method, irrespective of the network topology, the network scale, the number of sources, and the number of destinations.

5.4. The running time of involved algorithms

Note that the time complexity of our P-MCF and E-MCF methods have been analyzed in Sections 4.1 and 4.2.3, respectively. It is clear that the time complexity depends not only the network scale $|V|$ but also the number of sources and destinations of a given multi-source multicast. In general, the number of sources and destinations is far fewer than the network scale $|V|$.

In this section, we further evaluate the running time of our MCF algorithms and the SMT algorithm in small-scale random networks. The network scale varies from 50 to 500. The number of destinations of evaluated multi-source multicasts ranges from 10 to 100. The default number of sources is 3. The running environment of algorithms is equipped with a processor of Intel Core i7. We collect the running time of our algorithms under varied settings of evaluations, and report the average results in Table 2. It is clear that our two MCF methods consume some more time than the traditional SMT algorithm, due to the appearance of multiple sources. Note that the approximate method for SMT problem just consider the case of only one source node in a multicast transfer. Additionally, we can see that E-MCF consumes additional time than P-MCF because it has to search out those shared nodes after establishing one multicast forest.

6. Discussion

In this section, we extend the multi-source multicast by discussing several important issues, including the feasibility of sufficient sources, rethinking the SMT method, the scalability and capability of a multi-source multicast, and the virtual source multicast.

6.1. The feasibility of sufficient sources for MCF

For any multi-source multicast, both theoretical and evaluation results indicate that multiple sources would considerably reduce the total cost of resulting multicast forest under our MCF methods. In reality, the number of sources for a multi-source multicast is usually constrained by the real applications. For example, only three duplications are deployed for each file block in GFS. The evaluation results in Figure 7(a), Figure 9(b) and Figure 10(b) indicate that the cost of our resultant MCF becomes relatively stable when the number of sources exceeds a threshold, for example three. That is, three sources are efficient and usually sufficient for a multi-source multicast routing in practice. This fact would ease the precondition of realizing the multi-source multicast routing in various network applications.

6.2. Rethinking prior SMT methods

In the above evaluations, we initialize a MCF as a single-source multicast by randomly picking a source from all potential sources. The SMT method then establishes the minimum-cost tree for spanning the picked source and all destinations. The evaluation results indicate that our MCF methods always incur less cost than the SMT method of the initialized multicast. However, the total cost of the resultant SMT significantly depends on the selection of the source. It is clear that each picked

Table 2: The running time of our algorithms and SMT algorithm under varied settings of random networks and multi-source multicasts.

	$\lvert D\rvert$=10, $\lvert V\rvert$=50	$\lvert D\rvert$= 20, $\lvert V\rvert$=75	$\lvert D\rvert$=30, $\lvert V\rvert$= 150	$\lvert D\rvert$=40, $\lvert V\rvert$=200	$\lvert D\rvert$=50, $\lvert V\rvert$=250
SMT	0.0049s	0.01224s	0.02392s	0.03474s	0.04942s
P-MCF	0.00866s	0.0204s	0.03508s	0.05336s	0.07476s
E-MCF	0.01028s	0.02192s	0.03866s	0.05772s	0.08304s
	$\lvert D\rvert$=60, $\lvert V\rvert$= 300	$\lvert D\rvert$=70, $\lvert V\rvert$=350	$\lvert D\rvert$=80 , $\lvert V\rvert$= 400	$\lvert D\rvert$=90 , $\lvert V\rvert$=450	$\lvert D\rvert$=100 , $\lvert V\rvert$=500
SMT	0.068042s	0.08796s	0.1141s	0.13614s	0.16756s
P-MCF	0.0926s	0.11862s	0.14486s	0.17018s	0.2044s
E-MCF	0.11032s	0.1484s	0.18804s	0.23674s	0.29706s

source would cause a distinct multicast tree given the same multi-source multicast. However, it is not necessary that the randomly picked source would incur the multicast tree with the lowest cost. Thus, we need to demonstrate the gains of multiple sources for any multi-source multicast by rethinking and comparing the SMT method.

More previously, given any multi-source multicast we establish a single-source SMT for each potential source. Accordingly, we can find the SMT with the least cost among all generated SMTs and compare it with the MCF derived from our MCF methods. For this reason, we conduct experiments under a random topology with 2000 switches, 300 destinations, where the number of sources varies from 3 to 21. We can see from Figure 11 that the revised SMT method still performs worse compared with our two MCF methods in terms of the total link cost. The fundamental reason is that the SMT method and its variants cannot find and fully exploit the benefits of multiple sources. That is, the SMT method can't efficiently address the multi-source multicast proposed in this paper.

Table 3 further summarizes the cost of each SMT resulting from each available source. We can see that those SMTs under the same settings indeed exhibit varied link costs. The link cost difference among related SMTs, however, is not very remarkable.

Table 3: The link cost of all SMTs, under varied settings of MCF.

$\lvert S\rvert$	The link cost of each SMT
3	652 654 649
6	652 655 648 654 651 655
9	648 652 651 653 652 648 655 650 648
12	655 646 652 651 649 649 655 655 652 649 652 650
15	653 648 652 654 652 652 647 654 650 646 654 650 652 652 651
18	649 652 652 649 649 652 648 650 648 647 652 653 649 651 652 655 653 644
21	652 653 646 652 653 650 652 649 652 651 655 649 647 652 652 651 652 654 653 652 649

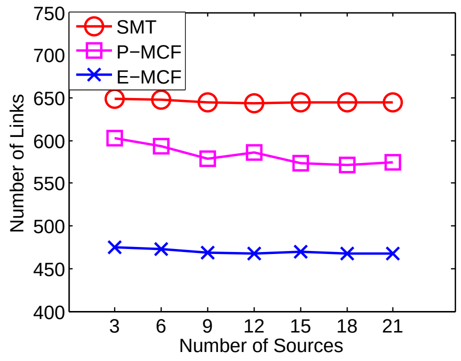


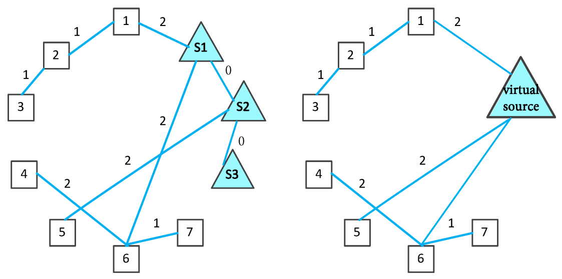
Figure 11: The changing trends of the link cost of the best SMT and our MCF, under a random topology with 2000 switches and 300 destinations.

6.3. The scalability and capacity of MCF

Figure 8(a), Figure 9(c) and Figure 10(c) indicate that a multi-source multicast session incurs growing cost of the resultant forest, when it serves increasing destinations and the number of sources is fixed. Thus, we further analyze the scalability of our MCF solutions. We can see that the total link cost of each resultant MCF is always less and increases more slowly than that of the SMT for the single-source multicast, when the amount of destinations increase. This manifests that a multi-source multicast is better than a related single-source multicast, in terms of the scalability. The root cause is that numerous destinations are not necessary to retrieve data from only one source. Additionally, our algorithms are also scalable in terms of the computation complexity as we have proved in Section 4.

The capacity problem of a multi-source multicast refers to the ability of accommodating as more multi-source multicast sessions as possible in a given network. It mainly depends on the available network resources and forest building methods. To improve the capacity of multi-source multicasts, the constructed forest for each multi-source multicast session should cause the least link cost. The saved link resources by each multi-source multicast session would be essential to support other multi-source multicast sessions. Therefore, the proposed MCF solutions theoretically perform better than existing SMT solutions for single-source multicast, when they're applied to support uncertain multicast streams. We will study the packing problem of multi-source multicasts to evaluate the multi-source multicast capacity of a given network in our future work.

6.4. Comparison with the virtual source multicast



(a) The minimum spanning tree T_1 resulting from our P-MCF method. (b) The minimum spanning tree T_2 of the virtual-source multicast.

Figure 12: An illustrative example of the virtual-source based multicast method for the MCF in Figure 2(a).

In theory, the multi-source multicast problem can be converted as the multicast problem by abstracting all sources into a virtual source. That is, a link of cost 0 is added between each pair of sources. Hence, we can construct the minimum spanning tree T_2 for the virtual source multicast in Figure 12(b). Let G_{VS} include only one virtual source (instead of any real source) and all destinations. For any destination node in G_{VS}, the distance towards the virtual source is just the minimum length among all shortest paths towards all real sources. That is, the edge from any destination to the virtual source represents the shortest path towards the closest source.

The virtual source multicast can achieve a SMT via any existing building method, as the single source multicast does. However, the resultant SMT would be further embodied as a forest structure after converting the virtual node as corresponding sources. The resultant forest can achieve an approximation ratio of $(2+\varepsilon)$. Actually, the forest building process for the virtual source multicast is equivalent to that in our P-MCF building method. That is, the virtual source multicast is just another representation of our multi-source multicast. The SMT for the virtual source multicast is equivalent to the forest from our P-MCF method. However, the SMT for a virtual source multicast does not achieve the benefit of shared nodes. Our E-MCF method can achieve better performance than the virtual source multicast, due to fully exploit the benefits of potential shared nodes.

7. Conclusion

Compared with unicast, multicast can naturally save the bandwidth consumption and reduce the load on the source. Although the appearance of SDN offers a good opportunity to implement multicast and its variants, multicast for SDN still attracts much less attention in literature. In this paper, we put forward a novel multi-source multicast with multiple potential sources. We concentrate on constructing a forest with the minimum cost (MCF). We further propose two $(2+\varepsilon)$-approximation methods, named P-MCF and E-MCF. The small-scale experiments and large-scale simulations demonstrate that our MCF approaches always occupy less network links than the traditional SMT, irrespective of the number of sources, destinations and scale in multicast transmissions.

Acknowledgements

This work is supported in part by the National Natural Science Foundation for Outstanding Excellent young scholars of China under grant No.61422214, National Basic Research Program (973 program) under Grant No. 2014CB347800, the Program for New Century Excellent Talents in University, the Distinguished Young Scholars of NUDT under grant No.JQ14-05-02, the Preliminary Research Funding of NUDT under grant NO. ZDYYJCYJ20140601, and the Guangxi Cooperative Innovation Center of cloud computing and Big Data under grant No.YD16507.

References

[1] 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," in *Proc. of the ACM SIGCOMM*, Barcelona, Spain, 2009, pp. 231–242.

[2] D. Li, Y. Li, J. Wu, S. Su, and J. Yu, "ESM: efficient and scalable data center multicast routing," *IEEE/ACM Transactions on Networking*, vol. 20, no. 3, 2012.

[3] 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.

[4] G. Robins and A. Zelikovsky, "Tighter bounds for graph steiner tree approximation," *SIAM J. Discrete Math*, vol. 19, no. 1, pp. 122–134, 2005.

[5] D. N. Y. Shan Hsiang Shen, Liang Hao Huang and W. T. Chen, "Reliable multicast routing for software-defined networks," in *Proc. of the IEEE INFOCOM*, Hong Kong, 2015.

[6] B. Chun, P. Wu, H. Weatherspoon, and J. Kubiatowicz, “Chunkcast: An anycast service for large content distribution,” in *Proc. of 5th International IPTPS, Santa Barbara, CA, USA*, 2006.

[7] D. Guo, J. Xie, X. Zhou, X. Zhu, W. Wei, and X. Luo, “Exploiting efficient and scalable shuffle transfers in future data center networks,” *IEEE Transactions on Parallel Distributed Systems*, vol. 26, no. 4, pp. 997–1009, 2015.

[8] L. Luo, D. Guo, W. Li, T. Zhang, J. Xie, and X. Zhou, “Compound graph based hybrid data center topologies,” *Frontiers of Computer Science*, vol. 9, no. 6, pp. 860–874, 2015.

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

[10] P. Berman and V. Ramaiyer, “Improved approximations for the steiner tree problem,” *J. Algorithms*, vol. 17, no. 3, pp. 381–408, 1994.

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

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

[13] J. Byrka, F. Grandoni, T. Rothvoß, and L. Sanità, “An improved lp-based approximation for steiner tree,” in *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, 2010, pp. 583–592.

[14] Y.-R. Chen, S. Radhakrishnan, S. Dhall, and S. Karabuk, “On multi-stream multi-source multicast routing,” *Computer Networks*, vol. 57, no. 15, pp. 2916–2930, 2013.

[15] G. Robins and A. Zelikovsky, “Minimum steiner tree construction,” in *Handbook of Algorithms for Physical Design Automation.*, 2008.

[16] E. Gassner, “The steiner forest problem revisited,” *Journal of Discrete Algorithms*, vol. 8, no. 2, pp. 154–163, 2010.

[17] X. Zheng, C. Cho, and Y. Xia, “Content distribution by multiple multicast trees and intersession cooperation: Optimal algorithms and approximations,” *Computer Networks*, vol. 83, pp. 100–117, 2015.

[18] D. Dingzhu, K. Ker-I, and H. Xiaodong, *Design and Analysis of Approximation Algorithms*. Higher Education Press, 2011.

[19] D. P. Sidhu, T. Fu, S. Abdallah, R. Nair, and R. Coltun, “Open shortest path first (OSPF) routing protocol simulation,” in *Proc. of SIGCOMM*, 1993, pp. 53–62.

[20] C. Zhong, M. I. Malinen, D. Miao, and P. Fränti, “A fast minimum spanning tree algorithm based on k-means,” *Inf. Sci.*, vol. 295, pp. 1–17, 2015.

[21] D. Kreutz, F. M. V. Ramos, P. J. E. Veríssimo, 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.

[22] 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.

[23] “ONetSwitch20,” http://www.meshsr.com/product/onetswitch20, [Online].

[24] S. Hub, “RYU Controller,” http://sdnhub.org/tutorials/ryu/, [Online].

[25] N. McKeown, T. Anderson, H. Balakrishnan, G. M. Parulkar, L. L. Peterson, J. Rexford, S. Shenker, and J. S. Turner, “Openflow: enabling innovation in campus networks,” *Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[26] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey, “Jellyfish: Networking data centers randomly.” in *Proc. USENIX NSDI*, San Jose, CA, USA, April 2012.