Theoretical Computer Science 410 (2009) 5415–5424

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

A 3.4713-approximation algorithm for the capacitated multicast tree routing problem

Zhipeng Cai[a], Zhi-Zhong Chen[b], Guohui Lin[a,*]

[a] Department of Computing Science, University of Alberta. Edmonton, Alberta T6G 2E8, Canada
[b] Department of Mathematical Sciences, Tokyo Denki University. Hatoyama, Saitama 350-0394, Japan

ARTICLE INFO

Keywords:
Capacitated multicast tree routing
Approximation algorithm
Steiner minimum tree
Tree partitioning

ABSTRACT

Given an underlying communication network represented as an edge-weighted graph $G = (V, E)$, a source node $s \in V$, a set of destination nodes $D \subseteq V$, and a capacity k which is a positive integer, the *capacitated multicast tree routing problem* asks for a minimum cost routing scheme for source s to send data to all destination nodes, under the constraint that in each routing tree at most k destination nodes are allowed to receive the data copies. The cost of the routing scheme is the sum of the costs of all individual routing trees therein. Improving on our previous approximation algorithm for the problem, we present a new algorithm which achieves a worst case performance ratio of $\frac{\sqrt{2089}+77}{80} + \frac{5}{4}\rho$, where ρ denotes the best known approximation ratio for the Steiner minimum tree problem. Since ρ is about 1.55 at the writing of the paper, the ratio achieved by our new algorithm is less than 3.4713. In comparison, the previously best ratio was $\frac{8}{5} + \frac{5}{4}\rho \approx 3.5375$.

1. Introduction

Multicast is a point-to-multipoint communication that a source node sends data to multiple destinations [1,16,10,8,15]. In computer and communication networks supporting multimedia applications, such as news feed and video distribution, multicast is an important service. Implementing multicast on local area networks (LANs) is easy because nodes connected to a LAN usually communicate over a broadcast network. In contrast, implementing multicast on wide area networks (WANs) is yet quite challenging [17,6] because nodes connected to a WAN communicate via a switched/routed network. Basically, to perform multicast in WANs, the source node and all the destination nodes must be interconnected. So, finding a multicast routing in a WAN is equivalent to finding a multicast tree T in the network such that T spans the source and all the destination nodes. The objective is to minimize the *cost* of T, which is defined to be the total weight of edges in T.

In certain networks such as WDM optical networks with limited light-splitting capabilities, only a limited number of destination nodes can be assigned to receive the data copies sent from the source node during each transmission. A routing model for such networks, called the *multi-tree model* [11,7], has been introduced in the literature. Under this model, we are interested in the problem of finding a set of routing trees such that each tree spans the source node and a limited number of destination nodes that is assigned to receive data copies, and every destination node must be designated to receive a data copy in one of the routing trees. We call this problem the *capacitated multicast routing problem*. In particular, when the number of destination nodes in each routing tree is limited to a pre-specified number k, we call it the *multicast k-tree routing* (kMTR) problem.

* Corresponding address: Department of Computing Science, University of Alberta, Athabasca Hall 2-21, Edmonton, Alberta T6G 2E8, Canada. Tel.: +1 780 492 3737; fax: +1 780 492 1071.
E-mail addresses: zhipeng@cs.ualberta.ca (Z. Cai), chen@r.dendai.ac.jp (Z.-Z. Chen), ghlin@cs.ualberta.ca (G. Lin).

0304-3975/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2009.05.013

We next formally define the problems. For a graph G, we denote its node set by $V(G)$. The underlying communication network is modeled as a triple (G, s, D), where G is a simple, undirected, and edge-weighted complete graph, $s \in V(G)$ is the *source* node, and $D \subseteq V(G) - \{s\}$ is the set of *destination* nodes. The weight of each edge e in G, denoted by $w(e)$, is nonnegative and represents the routing cost of e. The *weight* (or *cost*, used interchangeably) of a subgraph T of G, denoted by $w(T)$, is the total weight of edges in T.

A subgraph T of G is said to be a *D-marked Steiner tree* if (1) T is a tree, and (2) at least one destination node in T is marked (to receive a data copy). For each D-marked Steiner tree T, we use $D \cap T$ to denote the set of marked destination nodes in T. Note that some destination nodes in T may not be marked, and they are not allowed to receive data copies but serve as Steiner nodes. The *size* of T is defined to be the number of marked destination nodes in T, i.e., $|D \cap T|$. A set $\mathcal{T}$ of D-marked Steiner trees are *disjointly-D-marked* if $(D \cap T_1) \cap (D \cap T_2) = \emptyset$ for any two trees T_1 and T_2 in $\mathcal{T}$.

Given a positive integer k, a *k-tree routing* in network (G, s, D) is a set $\mathcal{T} = \{T_1, \dots, T_\ell\}$ of disjointly-D-marked Steiner trees such that each T_i $(1 \leq i \leq \ell)$ contains s and is of size at most k, and $D = \bigcup_{i=1}^{\ell}(D \cap T_i)$. The *weight* (or *cost*, used interchangeably) of a k-tree routing is the total weight of the D-marked Steiner trees in the routing. Given a network (G, s, D) and a number k, the *capacitated multicast routing problem* asks for a k-tree routing in (G, s, D) whose weight is minimized over all k-tree routings in (G, s, D). When k is fixed, the problem is called the *multicast k-tree routing* problem, denoted as kMTR for short.

When $k \geq |D|$, kMTR reduces to the well-known *Steiner minimum tree* (SMT) problem: Given a network (G, s, D), it asks for a minimum-weight tree T in G that spans $\{s\} \cup D$. The SMT problem is NP-hard, and its current best approximation ratio is $\rho \approx 1.55$ [5,14]. On the other hand, when $k \leq 2$, kMTR can be solved efficiently [7].

The algorithmically most interesting case is when $3 \leq k < |D|$, kMTR differs from the SMT problem yet remains NP-hard [2,12]. Two groups of researchers [2,4,9] independently designed $(2 + \rho)$-approximation algorithms, where ρ is the approximation ratio for the SMT problem. Later, Morsy and Nagamochi presented a new approximation algorithm with a worst-case performance ratio of $(\frac{3}{2} + \frac{4}{3}\rho)$, which leads to an improvement over the $(2 + \rho)$-approximation algorithms only when $\rho < 1.5$ [13]. Recently, we presented an $(\frac{8}{5} + \frac{5}{4}\rho)$-approximation algorithm which is based on the weight averaging technique introduced in [2,4] and an advanced tree partitioning technique [3]. This is a true improvement over the $(2 + \rho)$-approximation algorithms for the current ρ and its future values.

In this paper, we examine more carefully two cases where our previous algorithm does not perform well, and design better routing schemes for them respectively. The result is an improved approximation algorithm for kMTR, which achieves a performance ratio of $(\frac{\sqrt{2089}+77}{80} + \frac{5}{4}\rho)$. Given the fact that $\rho \approx 1.55$, the achieved improvement in performance ratio is $0.0662 = 3.5375 - 3.4713$, more than 5 times the last improvement of $0.0125 = 3.55 - 3.5375$.

2. Preliminaries

Throughout the rest of paper, fix a communication network (G, s, D) and a positive integer k. For ease of explanation (to avoid dealing with floors and ceilings), we assume that k is a multiple of 12. Recall that G is a simple, undirected, and edge-weighted complete graph, $s \in V(G)$ is the source node, and $D \subseteq V(G) - \{s\}$ is the set of destination nodes. The non-destination nodes in $V(G) - (D \cup \{s\})$, as well as destination nodes when unmarked, can be used as intermediate (Steiner) nodes in a routing to save the routing cost.

For each edge (u, v) in G, we use $w(u, v)$ to denote its weight. If (u, v) is an edge in G such that $w(u, v)$ is larger than the weight of the shortest path between u and v in G, then (u, v) is useless in any routing and hence can be ignored. Therefore, we may assume that for each pair $\{u, v\}$ of nodes in G, $w(u, v)$ equals the weight of the shortest path between u to v in G. It follows that the edge weight function $w(\cdot, \cdot)$ of G satisfies the triangle inequality.

Let $\mathcal{T}^*$ be an optimal k-tree routing in network (G, s, D). Let $R^* = \sum_{T \in \mathcal{T}^*} w(T)$ denote the weight of the k-tree routing $\mathcal{T}^*$. Clearly, if d is a marked destination node in a routing tree $T \in \mathcal{T}^*$, then $w(s, d) \leq w(T)$. Thus, we have

$$\sum_{d \in D} w(s, d) \leq \sum_{T \in \mathcal{T}^*} \sum_{d \in D \cap T} w(s, d) \leq \sum_{T \in \mathcal{T}^*} (k \times w(T)) = k \times \sum_{T \in \mathcal{T}^*} w(T) \leq k \times R^*. \tag{2.1}$$

In the following design of the approximation algorithm for kMTR, we first apply the best known approximation algorithm for the SMT problem (which has a worst-case performance ratio of ρ) to obtain a Steiner tree T^0 on $\{s\} \cup D$ in network (G, s, D). Recall that T^0 is a subgraph of G that is a D-marked Steiner tree with $D \cap T^0 = D$. Since the weight of an optimal Steiner tree is a lower bound on R^*, the weight of tree T^0 is upper bounded by ρR^*, that is, $w(T^0) \leq \rho R^*$. We now root tree T^0 at source s. Note that tree T^0 does not necessarily correspond to a k-tree routing, because some subtrees rooted at child nodes of s in T^0 may contain more than k marked destination nodes.

In the sequel, for a D-marked Steiner tree T in G and a node v in T, we use T_v to denote the subtree of T rooted at v. For a child u of an internal node v in T, the subtree T_v together with edge (v, u) is called the *branch rooted at v and containing u*. Recall that $D \cap T$ denotes the set of marked destination nodes in T and the size of T is defined as $|D \cap T|$. If $|D \cap T| \leq k$, then T can be used in a k-tree routing to route those nodes in $D \cap T$. If source s is not in T, then we can add s and the edge (s, u) to T, where u is a node in T such that $w(s, u) = \min_{v \in V(T)} w(s, v)$. Let $c(T)$ denote $\min_{v \in V(T)} w(s, v)$. Note that

$c(T) = 0$ if $s \in V(T)$. We call $c(T)$ the *connection cost* of T and define the *routing cost* of T to be $w(T) + c(T)$. Moreover, since $c(T) \leq \min_{d\in D\cap T} w(s, d)$, we have

$$c(T) \leq \frac{1}{|D \cap T|} \sum_{d\in D\cap T} w(s, d). \tag{2.2}$$

Although tree T^0, computed by the approximation algorithm for the SMT problem, does not necessarily correspond to a k-tree routing, it serves as a good starting point because $w(T^0) \leq \rho R^*$. Our idea is to transform T^0 into a k-tree routing without increasing its weight significantly. Basically, the transformation is done by case analysis. Each case corresponds to a lemma in Section 3. With these lemmas, we will define several types of operations in Section 4.1 that can be applied to T^0 (to turn it into a k-tree routing). The whole algorithm is presented in Section 4.2.

3. Tree partitioning lemmas

We will prove several lemmas that help us transform T^0 into a k-tree routing. Essentially, the transformation process is to repeatedly cut a subtree T out of T^0 and route all the destination nodes therein. The number of destination nodes in T satisfies some conditions, to be specified, and accordingly we determine a way to mark the destination nodes in T for routing purposes. The following Lemma 3.1 is proven in [3]; Lemmas 3.4 and 3.5 are used in [3] without proofs.

Lemma 3.1 (*[3]*). *If T is a D-marked Steiner tree such that*

- $\frac{2}{3}k \leq |D \cap T| \leq k$,

then the routing cost of T is at most $w(T) + \frac{3}{2} \times \frac{1}{k} \sum_{d\in D\cap T} w(s, d)$.

Lemma 3.2 (*[2,4]*). *Given a D-marked Steiner tree T such that*

- $k < |D \cap T| \leq \frac{3}{2}k$,

we can compute two disjointly-D-marked Steiner trees X_1 and X_2 from T in polynomial time such that both X_1 and X_2 are of size at most k, $D \cap T = (D \cap X_1) \cup (D \cap X_2)$, *and the total routing cost of X_1 and X_2 is at most* $w(T) + 2 \times \frac{1}{k} \sum_{d\in D\cap T} w(s, d)$.

Lemma 3.3. *Suppose that T is a D-marked Steiner tree satisfying the following conditions:*

- $\frac{4}{3}k \leq |D \cap T| \leq \frac{3}{2}k$.
- *The root r of T has exactly three child nodes* v_1, v_2, *and* v_3.
- $|D \cap T_{v_1}| < \frac{2}{3}k$, $|D \cap T_{v_2}| < \frac{2}{3}k$, *and* $|D \cap T_{v_1}| + |D \cap T_{v_2}| > k$.

Given T, we can compute two disjointly-D-marked Steiner trees X_1 and X_2 in polynomial time such that both X_1 and X_2 are of size at most k, $D\cap T = (D\cap X_1) \cup (D\cap X_2)$, *and the total routing cost of X_1 and X_2 is at most* $\frac{5}{4}w(T) + \frac{\sqrt{2089}+77}{80} \times \frac{1}{k} \sum_{d\in D\cap T} w(s, d)$.

Proof. By the conditions in the lemma, $|D \cap T_{v_3}| < \frac{1}{2}k$. Without loss of generality, we assume that $|D \cap T_{v_1}| \leq |D \cap T_{v_2}|$. Then, $|D \cap T_{v_2}| > \frac{1}{2}k$. For each $i \in \{1, 2, 3\}$, let B_i be the branch rooted at r and containing v_i. We distinguish two cases as follows.

Case 1: $|D \cap T_{v_3}| + |D \cap T_{v_2}| \leq k$. In this case, $|D \cap T_{v_3}| + |D \cap T_{v_1}| \leq k$. Among the nodes in $D \cap T$, we find the $\frac{2}{3}k$ closest nodes to s, and form them into a set C. Similarly, among the nodes in $D \cap T$, we find the $\frac{2}{3}k$ farthest nodes from s, and form them into a set F. Since $|D \cap T| \geq \frac{4}{3}k$, $F \cap C = \emptyset$. Moreover, since $|D \cap T_{v_i}| < \frac{2}{3}k$ for each $i \in \{1, 2, 3\}$, there are at least two indices $i \in \{1, 2, 3\}$ such that $(D \cap T_{v_i}) \cap C \neq \emptyset$. If $(D \cap T_{v_3}) \cap C = \emptyset$, then we set $X_1 = B_1$ and construct X_2 by initializing it as the union of B_2 and B_3 and further unmarking r if it is marked. Otherwise, we find an index $i \in \{1, 2\}$ with $(D \cap T_{v_i}) \cap C \neq \emptyset$, set $X_1 = B_i$ and construct X_2 by initializing it as the union of B_j and B_3 and further unmarking r if it is marked, where j is the other index in $\{1, 2\} - \{i\}$. In any case, $|D \cap X_1| \leq k$, $|D \cap X_2| \leq k$, $(D \cap X_1) \cap C \neq \emptyset$, and $(D\cap X_2)\cap C \neq \emptyset$. Obviously, one of $D\cap X_1$ and $D\cap X_2$ contains d' which is the closest destination node to s among the nodes in $D \cap T$. We assume that $D \cap X_1$ contains d'; the other case is symmetric. Then, $c(X_1) \leq w(s, d') \leq \frac{3}{2} \times \frac{1}{k} \sum_{d\in C} w(s, d)$. Moreover, since $(D \cap X_2) \cap C \neq \emptyset$, $c(X_2) \leq w(s, d'')$ where d'' is the farthest destination node from s among the nodes in C. Furthermore, since $C \cap F = \emptyset$, $w(s, d'') \leq w(s, d''')$ where d''' is the closest destination node to s among the nodes in F. Thus, $c(X_2) \leq w(s, d''') \leq \frac{3}{2} \times \frac{1}{k} \sum_{d\in F} w(s, d)$. Therefore, $c(X_1) + c(X_2) \leq \frac{3}{2} \times \frac{1}{k} \sum_{d\in D\cap T} w(s, d)$. Consequently, the total routing cost of X_1 and X_2 is at most $w(T) + \frac{3}{2} \times \frac{1}{k} \sum_{d\in D\cap T} w(s, d)$, and the lemma is proved.

Case 2: $|D \cap T_{v_3}| + |D \cap T_{v_2}| > k$. We assume that $w(B_1) \leq w(B_3)$; this does not lose generality because our argument will not take advantage of the difference between the two conditions that $|D \cap T_{v_1}| < \frac{2}{3}k$ and $|D \cap T_{v_3}| < \frac{1}{2}k$. We further distinguish two subcases as follows.

Subcase 2.1: $w(B_2) \leq w(B_1)$. We give two options for constructing X_1 and X_2. In the first option, we set $X_1 = T_{v_2}$ and set X_2 to be the union of B_1 and B_3. Obviously, $|D\cap X_1| \leq k$. We also have $|D\cap X_2| \leq k$ because $|D \cap T_{v_2}| > \frac{1}{2}k$ and $|D\cap T| \leq \frac{3}{2}k$.

The total routing cost of X_1 and X_2 is $w_1 \le w(T) + \frac{1}{|D\cap T_{v_1}|+|D\cap T_{v_3}|}\sum_{d\in(D\cap T_{v_1})\cup(D\cap T_{v_3})} w(s,d) + \frac{1}{|D\cap T_{v_2}|}\sum_{d\in D\cap T_{v_2}} w(s,d)$. Since $|D\cap T_{v_2}| \le 2k-2|D\cap T_{v_2}| \le |D\cap T_{v_1}|+|D\cap T_{v_3}|$, we have $w_1 \le w(T) + \frac{1}{|D\cap T_{v_2}|}\sum_{d\in(D\cap T_{v_1})\cup(D\cap T_{v_2})\cup(D\cap T_{v_3})} w(s,d)$.

In the second option, we first partition $D\cap T_{v_2}$ into two disjoint sets C_1 and C_3 with $|C_1| = \left\lceil \frac{|D\cap T_{v_2}|}{2}\right\rceil$ and $|C_3| = \left\lfloor \frac{|D\cap T_{v_2}|}{2}\right\rfloor$. Note that $|C_1| \le \frac{1}{3}k$ because $|D\cap T_{v_2}| < \frac{2}{3}k$ and k is a multiple of 12. Hence, $|D\cap T_{v_1}|+|C_1| < k$ and $|D\cap T_{v_3}|+|C_3| < k$. We construct X_1 by initializing it as the union of B_1 and B_2, unmarking the nodes in C_3, and further unmarking r if it is marked in T. We construct X_2 by initializing it as the union of B_2 and B_3, and further unmarking the nodes in C_1. Since both $|D\cap X_1|$ and $|D\cap X_2|$ are larger than $k-\left\lceil \frac{|D\cap T_{v_2}|}{2}\right\rceil$, the total routing cost of X_1 and X_2 is $w_2 \le \frac{4}{3}w(T) + \frac{1}{k-|D\cap T_{v_2}|/2}\sum_{d\in D\cap(X_1\cup X_2)} w(s,d) = \frac{4}{3}w(T) + \frac{1}{k-|D\cap T_{v_2}|/2}\sum_{d\in(D\cap T_{v_1})\cup(D\cap T_{v_2})\cup(D\cap T_{v_3})} w(s,d)$.

Note that $\min\{w_1,w_2\} \le \frac{1}{4}w_1 + \frac{3}{4}w_2 = \frac{5}{4}w(T) + \left(\frac{1}{4|D\cap T_{v_2}|} + \frac{3}{4(k-|D\cap T_{v_2}|/2)}\right)\sum_{d\in D\cap T} w(s,d)$. From $\frac{1}{2}k < |D\cap T_{v_2}| < \frac{2}{3}k$, we conclude that $\min\{w_1,w_2\} < \frac{5}{4}w(T) + \frac{3}{2}\times\frac{1}{k}\sum_{d\in D\cap T} w(s,d)$. Therefore, in this subcase, choosing the better option between the two proves the lemma.

Subcase 2.2: $w(B_2) > w(B_1)$. Then, $w(B_1) \le \frac{1}{3}w(T)$. For ease of presentation, let $|D\cap T_{v_2}| = \frac{2}{3}k - p$, where $0 < p < \frac{1}{6}k$. We choose an arbitrary subset F_1 of $D\cap T_{v_1}$ with $|F_1| = k - |D\cap T_{v_2}|$. Since $|D\cap T_{v_3}| > \frac{1}{3}k$, we choose an arbitrary subset F_3 of $D\cap T_{v_3}$ with $|F_3| = \frac{1}{3}k$. Again, we give two options for constructing X_1 and X_2. In the first option, we set $X_1 = T_{v_2}$ and set X_2 to be the union of B_1 and B_3. Obviously, $|D\cap X_1| \le k$. We also have $|D\cap X_2| \le k$ because $|D\cap T_{v_2}| > \frac{1}{2}k$ and $|D\cap T| \le \frac{3}{2}k$. The total routing cost of X_1 and X_2 is $w_1 \le w(T) + \frac{1}{|D\cap T_{v_2}|}\sum_{d\in D\cap T_{v_2}} w(s,d) + \frac{1}{|F_1|+|F_3|}\sum_{d\in F_1\cup F_3} w(s,d) = w(T) + \frac{1}{2k/3-p}\sum_{d\in D\cap T_{v_2}} w(s,d) + \frac{1}{2k/3+p}\sum_{d\in F_1\cup F_3} w(s,d)$.

We next describe the second option. We first choose an arbitrary subset F_1' of F_1 with $|F_1'| = x+p$, where $0 \le x \le \frac{1}{3}k$ and its value will be determined later. Set F_1' exists because $|F_1| = \frac{1}{3}k + p$. We now construct X_1 by initializing it as the union of B_1 and B_2, then unmarking the nodes of $(D\cap T_{v_1}) - F_1'$, and further unmarking r if it is marked in T. We then construct X_2 by initializing it as the union of B_1 and B_3 and further unmarking the nodes of F_1'. Since $|F_1| + |D\cap T_{v_2}| = k$ and $|F_1'| \le |F_1|$, we have $|D\cap X_1| \le k$. We also have $|D\cap X_2| \le k$ because $|D\cap T_{v_2}| > \frac{1}{2}k$ and $|D\cap T| \le \frac{3}{2}k$. The total routing cost of X_1 and X_2 is $w_2 \le \frac{4}{3}w(T) + \frac{1}{|F_1'|+|D\cap T_{v_2}|}\sum_{d\in F_1'\cup(D\cap T_{v_2})} w(s,d) + \frac{1}{|F_1-F_1'|+|F_3|}\sum_{d\in(F_1-F_1')\cup F_3} w(s,d) = \frac{4}{3}w(T) + \frac{1}{2k/3+x}\sum_{d\in(D\cap T_{v_2})\cup F_1'} w(s,d) + \frac{1}{2k/3-x}\sum_{d\in(F_1-F_1')\cup F_3} w(s,d)$. Since $\frac{2}{3}k+x \ge \frac{2}{3}k-x$, we have $w_2 \le \frac{4}{3}w(T) + \frac{1}{2k/3+x}\sum_{d\in D\cap T_{v_2}} w(s,d) + \frac{1}{2k/3-x}\sum_{d\in F_1\cup F_3} w(s,d)$.

Note that $\min\{w_1,w_2\} \le \frac{1}{4}w_1 + \frac{3}{4}w_2 \le \frac{5}{4}w(T) + M_1(x)\times\sum_{d\in D\cap T_{v_2}} w(s,d) + M_2(x)\times\sum_{d\in F_1\cup F_3} w(s,d) \le \frac{5}{4}w(T) + \max\{M_1(x),M_2(x)\}\times\sum_{d\in D\cap T} w(s,d)$, where $M_1(x) = \frac{1}{4}\times\frac{1}{2k/3-p} + \frac{3}{4}\times\frac{1}{2k/3+x}$ and $M_2(x) = \frac{1}{4}\times\frac{1}{2k/3+p} + \frac{3}{4}\times\frac{1}{2k/3-x}$.

Next, we want to determine the value for x (or $x(p)$, as it is a function of p) such that $\max\{M_1(x),M_2(x)\}$ is minimized. For a fixed $p\in(0,\frac{1}{6}k)$, $M_1(x)$ is monotonously decreasing in x and $M_2(x)$ is monotonously increasing in x, where $x\in[0,\frac{1}{3}k]$. Clearly, we have $M_1(0) > \frac{3}{2}k > M_2(0)$ and $M_1(\frac{1}{3}k) < \frac{5}{4}k < \frac{5}{2}k < M_2(\frac{1}{3}k)$. So the minimum value of $\max\{M_1(x),M_2(x)\}$ is achieved at a unique value of x, at which $M_1(x) = M_2(x)$ holds. The solutions to the equation $M_1(x) = M_2(x)$ are

$$x = \frac{-(12k^2-27p^2) \pm \sqrt{(12k^2-27p^2)^2 + 144p^2k^2}}{18p},$$

and thus $x^* = \frac{-(12k^2-27p^2)+\sqrt{(12k^2-27p^2)^2+144p^2k^2}}{18p}$ is the unique value in $[0,\frac{1}{3}k]$ such that $M_1(x^*) = M_2(x^*)$. Letting $c = \frac{2}{3}k$, from $\frac{\mathrm{d}M_1(x^*)}{\mathrm{d}p} = \frac{\mathrm{d}M_2(x^*)}{\mathrm{d}p}$, we have

$$\frac{\mathrm{d}x^*}{\mathrm{d}p} = \frac{1}{3}\times\frac{\frac{1}{(c-p)^2}+\frac{1}{(c+p)^2}}{\frac{1}{(c-x^*)^2}+\frac{1}{(c+x^*)^2}}.$$

Hence,

$$\frac{\mathrm{d}M_1(x^*)}{\mathrm{d}p} = \frac{1}{4}\times\left(\frac{1}{(c-p)^2} - \frac{3}{(c+x^*)^2}\frac{\mathrm{d}x^*}{\mathrm{d}p}\right) = \frac{1}{4}\times\frac{(c+p)^2(c+x^*)^2-(c-p)^2(c-x^*)^2}{(c+p)^2(c-p)^2\left((c+x^*)^2+(c-x^*)^2\right)} > 0.$$

This says that $M_1(x^*)$ is monotonously increasing in p in the interval $(0,\frac{1}{6}k)$. Therefore, the maximum value of $M_1(x^*)$ is infinitely close to

$$\frac{1}{4}\times\frac{2}{k} + \frac{3}{4}\times\frac{1}{\left(\frac{2}{3}+\frac{-\frac{45}{4}+\sqrt{(\frac{45}{4})^2+4}}{3}\right)k} = \left(\frac{1}{2}+\frac{9}{\sqrt{2089}-37}\right)\times\frac{1}{k} = \frac{\sqrt{2089}+77}{80}\times\frac{1}{k},$$

when p approaches to $\frac{1}{6}k$.

In summary, we have shown that in Subcase 2.2, the better option among the two has a cost less than $\frac{5}{4}w(T)+\frac{\sqrt{2089}+77}{80}\times\frac{1}{k}\sum_{d\in D\cap T}w(s,d)$. Note that $\frac{\sqrt{2089}+77}{80}\approx 1.5338>\frac{3}{2}$. This proves the lemma. □

Lemma 3.4 ([3]). *Suppose that T is a D-marked Steiner tree satisfying the following conditions:*

- $\frac{3}{2}k<|D\cap T|\le 2k$.
- *The root r of T has exactly three children v_1, v_2, and v_3.*
- $|D\cap T_{v_1}|<\frac{2}{3}k$, $|D\cap T_{v_2}|<\frac{2}{3}k$, *and* $|D\cap T_{v_1}|+|D\cap T_{v_2}|>k$.

Given T, we can compute disjointly-D-marked Steiner trees $X_1,\ldots,X_p$ with $2\le p\le 3$ in polynomial time such that each X_i ($1\le i\le p$) is of size at most k, $D\cap T=\bigcup_{i=1}^{p}(D\cap X_i)$, and the total routing cost of $X_1,\ldots,X_p$ is at most $\frac{5}{4}w(T)+\frac{3}{2}\times\frac{1}{k}\sum_{d\in D\cap T}w(s,d)$.

Proof. We have two options to route all the destination nodes in D_r.

Note that $|D_{v_1}|+|D_{v_2}|>k$. Without loss of generality, assume $|D_{v_1}|\le|D_{v_2}|$. Then, $|D_{v_2}|>\frac{1}{2}k$ and consequently $k\le|D_{v_1}|+|D_{v_3}|\le\frac{3}{2}k$. In the first option of routing, the branch rooted at node r and containing child node v_2 is separated as a routing subtree and Lemma 3.2 is applied to partition the remainder into two subtrees of size $\le k$. The total routing cost is thus $w_1\le w(T_r)+\frac{1}{|D_{v_2}|}\sum_{d\in D_{v_2}}w(s,d)+2\times\frac{1}{k}\sum_{d\in D_r-D_{v_2}}w(s,d)<w(T_r)+2\times\frac{1}{k}\sum_{d\in D_r}w(s,d)$.

In the second option of routing, we duplicate the least weight branch among those three. Each of the duplicated branch copies is merged with another branch to form a subtree and the destination nodes in the duplicated branch are distributed such that both resultant subtrees have size exactly $\frac{1}{2}|D_r|$. This can be done since $|D_{v_1}|<\frac{2}{3}k$, $|D_{v_2}|<\frac{2}{3}k$, and $|D_{v_1}|+|D_{v_2}|>k$. Note that $\frac{3}{4}k\le\frac{1}{2}|D_r|\le k$. Consequently, the total routing cost for these two subtrees is $w_2\le\frac{4}{3}w(T_r)+\frac{4}{3}\times\frac{1}{k}\sum_{d\in D_r}w(s,d)$.

Since $\min\{w_1,w_2\}\le\frac{1}{4}w_1+\frac{3}{4}w_2\le\frac{5}{4}w(T_r)+\frac{3}{2}\times\frac{1}{k}\sum_{d\in D_r}w(s,d)$, we choose the smaller cost routing between the two options and the lemma is proved. □

Lemma 3.5 ([3]). *Suppose that T is a D-marked Steiner tree satisfying the following conditions:*

- $2k<|D\cap T|\le\frac{5}{2}k$.
- *The root r of T has exactly two children v_1 and v_2.*
- $k<|D\cap T_{v_1}|<\frac{4}{3}k$ *and* $k<|D\cap T_{v_2}|<\frac{4}{3}k$.
- *For each $i\in\{1,2\}$, there is a node u_i in T_{v_i} (possibly $u_i=v_i$) such that u_i has exactly two children $x_{i,1}$ and $x_{i,2}$,* $|D\cap T_{x_{i,1}}|<\frac{2}{3}k$, $|D\cap T_{x_{i,2}}|<\frac{2}{3}k$, *and* $|D\cap T_{x_{i,1}}|+|D\cap T_{x_{i,2}}|>k$.

Given T, we can compute three disjointly-D-marked Steiner trees X_1, X_2, and X_3 in polynomial time such that each X_i ($1\le i\le 3$) is of size at most k, $D\cap T=\bigcup_{i=1}^{3}(D\cap X_i)$, and the total routing cost of X_1, X_2, and X_3 is at most $\frac{5}{4}w(T)+\frac{3}{2}\times\frac{1}{k}\sum_{d\in D\cap T}w(s,d)$.

Proof. Without loss of generality, we assume that $w(u_1,x_{11})+w(T_{x_{11}})\le\min\{w(u_1,x_{12})+w(T_{x_{12}}),w(u_2,x_{21})+w(T_{x_{21}}),w(u_2,x_{22})+w(T_{x_{22}})\}$, that is, the branch rooted at u_1 and containing node x_{11} has the least weight among those four branches rooted at u_1 and u_2. Afterwards, we also assume without loss of generality that $|D_{x_{21}}|\le|D_{x_{22}}|$. It follows that $|D_{x_{22}}|>\frac{1}{2}k$. From the branch rooted at r and containing node v_1, we remove tree T_{u_1} to obtain another tree denoted as T_3; Similarly, from the branch rooted at r and containing node v_2, we remove tree T_{u_2} to obtain another tree denoted as T_4. Let D_3 and D_4 denote the destination node set of T_3 and T_4, respectively. Clearly, $D_r=D_{x_{11}}\cup D_{x_{12}}\cup D_3\cup D_4\cup D_{x_{21}}\cup D_{x_{22}}$.

The destination node set $D_{x_{11}}$ is (arbitrarily) partitioned into two subsets $D^1_{x_{11}}$ and $D^2_{x_{11}}$, such that $\frac{2}{3}k\le|D^1_{x_{11}}|+|D_{x_{12}}|\le k$ and $\frac{4}{3}k\le|D_r-D^1_{x_{11}}-D_{x_{12}}|\le\frac{3}{2}k$. It follows from $|D_{x_{22}}|>\frac{1}{2}k$ that $|D^2_{x_{11}}|+|D_3|+|D_4|+|D_{x_{21}}|=|D_r-D^1_{x_{11}}-D_{x_{12}}-D_{x_{22}}|<k$. Let D^0 denote the set of the $\frac{2}{3}k$ farthest destination nodes from $D_r-D^1_{x_{11}}-D_{x_{12}}$. There are two possible cases: In the first case, $D_{x_{22}}\not\subset D^0$, that is, $D_{x_{22}}$ contains some destination node that is not in D^0. We route all the destination nodes in D_r by duplicating the branch rooted at u_1 and containing node x_{11}. One of the two copies of the duplicated branch is merged with the branch also rooted at u_1 but containing node x_{12} to form a subtree, which is assigned as a destination node set $D^1_{x_{11}}\cup D_{x_{12}}$. Another subtree is formed by the branch rooted at u_2 and containing node x_{22}, of which the destination node set is $D_{x_{22}}$. The third subtree is formed by the remainder tree, of which the destination node set is $D^2_{x_{11}}\cup D_3\cup D_4\cup D_{x_{21}}$. The connection cost of the first subtree is $\le\frac{3}{2}\times\frac{1}{k}\sum_{d\in D^1_{x_{11}}\cup D_{x_{12}}}w(s,d)$. Since the last two subtrees both contain some destination nodes that are not in D^0, one of them has connection cost of $w(s,d')$ and the other has connection cost of at most $w(s,d'')$, where d' and d'' are the closest destination nodes from $D_r-D^1_{x_{11}}-D_{x_{12}}$ and D^0, respectively. From $|D_r-D^1_{x_{11}}-D_{x_{12}}|\ge\frac{4}{3}k$, we conclude that the sum of the connection costs of these last two subtrees is $\le\frac{3}{2}\times\frac{1}{k}\sum_{d\in D_r-D^1_{x_{11}}-D_{x_{12}}}w(s,d)$. Thus, the total routing cost of these three subtrees is $\le\frac{5}{4}w(T_r)+\frac{3}{2}\times\frac{1}{k}\sum_{d\in D_r}w(s,d)$, and the lemma is proved.

In the second case, $D_{x_{22}}\subset D^0$, that is, every destination node in $D_{x_{22}}$ is among the $\frac{2}{3}k$ farthest destination nodes from $D^2_{x_{11}}\cup D_3\cup D_4\cup D_{x_{21}}\cup D_{x_{22}}$. We distinguish two subcases of whether or not $w(u_2,x_{21})+w(T_{x_{21}})\le w(u_2,x_{22})+w(T_{x_{22}})+w(T_3)+w(T_4)$. In the first subcase, $w(u_2,x_{21})+w(T_{x_{21}})\le w(u_2,x_{22})+w(T_{x_{22}})+w(T_3)+w(T_4)$.

We have two options to route all the destination nodes in D_r. In the first option, we revise the partition of the destination node set $D_{x_{11}}$ into two subsets $D^1_{x_{11}}$ and $D^2_{x_{11}}$, such that $|D^1_{x_{11}}|+|D_{x_{12}}|=k$, by moving some destination nodes from the old $D^2_{x_{11}}$ to $D^1_{x_{11}}$, if necessary. The destination node subset D^0 is updated correspondingly, which could include some more destination nodes not in the old D^0, due to possible moving destination nodes out of the old $D^2_{x_{11}}$. Since $|D_{x_{21}} \cup D_{x_{22}}| > k$, we conclude that the closest destination node in $D_{x_{21}}$ does not belong to D^0. Let $D^1_{x_{21}}$ denote the subset of $D_{x_{21}}$ to contain the $k - |D_{x_{22}}|$ closest destination nodes; Let $D^2_{x_{21}} = D_{x_{21}} - D^1_{x_{21}}$. We route all the destination nodes in D_r by duplicating the branch rooted at u_1 and containing node x_{11}. One of the two copies of the duplicated branch is merged with the branch also rooted at u_1 but containing node x_{12} to form a subtree, which is assigned destination node set $D^1_{x_{11}} \cup D_{x_{12}}$ (of size k). Another subtree is formed by the branch rooted at u_2 and containing node x_{22}, of which the destination node set is $D_{x_{22}}$. The third subtree is formed by the remainder tree, of which the destination node set is $D^2_{x_{11}} \cup D_3 \cup D_4 \cup D_{x_{21}}$. The connection cost of the first subtree is $\le \frac{1}{k}\sum_{d\in D^1_{x_{11}}\cup D_{x_{12}}} w(s,d)$; The connection cost of the second subtree is at most $w(s,d')$, where d' is the $(\frac{1}{2}k+1)$-st farthest destination node from $D_{x_{22}}$; The connection cost of the third subtree is at most $w(s,d'')$, where d'' is the closest destination node from $D^1_{x_{21}}$, and thus the closest from $D^1_{x_{21}} \cup D_{x_{22}}$. From $|D^1_{x_{21}} \cup D_{x_{22}}| = k$, we conclude that the sum of the connection costs of these last two subtrees is $\le 2\times\frac{1}{k}\sum_{d\in D^1_{x_{21}}\cup D_{x_{22}}} w(s,d)$. Thus, the total routing cost of these three subtrees is $w_1 \le w(T_r) + w(u_1,x_{11}) + w(T_{x_{11}}) + \frac{1}{k}\sum_{d\in D^1_{x_{11}}\cup D_{x_{12}}} w(s,d) + 2\times\frac{1}{k}\sum_{d\in D^1_{x_{21}}\cup D_{x_{22}}} w(s,d)$.

In the second option of routing, we route all the destination nodes in D_r by duplicating the branch rooted at u_2 and containing node x_{21}. One of the two copies of the duplicated branch is merged with the branch also rooted at u_2 but containing node x_{22} to form a subtree, which is assigned as a destination node set $D^1_{x_{21}} \cup D_{x_{22}}$ (of size k). Since $|D_{x_{11}} \cup D_{x_{12}} \cup D_3 \cup D_4 \cup D^2_{x_{21}}| \in (k, \frac{3}{2}k]$, we apply Lemma 3.2 to route them without increasing the tree weight but using two subtrees of total connection cost of $\le 2\times\frac{1}{k}\sum_{d\in D_r - D^1_{x_{21}} - D_{x_{22}}} w(s,d)$. Therefore, the total routing cost of these three subtrees is $w_2 \le w(T_r) + w(u_2,x_{21}) + w(T_{x_{21}}) + \frac{1}{k}\sum_{d\in D^1_{x_{21}}\cup D_{x_{22}}} w(s,d) + 2\times\frac{1}{k}\sum_{d\in D_{x_{11}}\cup D_{x_{12}}\cup D_3\cup D_4\cup D^2_{x_{21}}} w(s,d)$.

Since $\min\{w_1,w_2\} \le \frac{1}{2}(w_1+w_2) \le w(T_r) + \frac{1}{2}(w(u_1,x_{11}) + w(T_{x_{11}}) + w(u_2,x_{21}) + w(T_{x_{21}})) + \frac{3}{2}\times\frac{1}{k}\sum_{d\in D^1_{x_{11}}\cup D_{x_{12}}\cup D^1_{x_{21}}\cup D_{x_{22}}} w(s,d) + \frac{1}{k}\sum_{d\in D^2_{x_{11}}\cup D_3\cup D_4\cup D^2_{x_{21}}} w(s,d) \le \frac{5}{4}w(T_r) + \frac{3}{2}\times\frac{1}{k}\sum_{d\in D_r} w(s,d)$, in this subcase, we choose the smaller cost routing between the two options and the lemma is proved.

In the second subcase, $w(u_2,x_{21}) + w(T_{x_{21}}) > w(u_2,x_{22}) + w(T_{x_{22}}) + w(T_3) + w(T_4)$. From $w(u_1,x_{11}) + w(T_{x_{11}}) \le w(u_1,x_{12}) + w(T_{x_{12}})$, it follows that $w(u_1,x_{11}) + w(T_{x_{11}}) + w(u_2,x_{22}) + w(T_{x_{22}}) + w(T_3) + w(T_4) < w(u_1,x_{12}) + w(T_{x_{12}}) + w(u_2,x_{21}) + w(T_{x_{21}})$. Therefore, $w(u_1,x_{11}) + w(T_{x_{11}}) + w(u_2,x_{22}) + w(T_{x_{22}}) + w(T_3) + w(T_4) < \frac{1}{2}w(T_r)$. Assume without loss of generality that $w(u_1,x_{11}) + w(T_{x_{11}}) + w(T_4) \le w(u_2,x_{22}) + w(T_{x_{22}}) + w(T_3)$. Then, $w(u_1,x_{11}) + w(T_{x_{11}}) + w(T_4) < \frac{1}{4}w(T_r)$. We have one option to route all the destination nodes in D_r.

We partition the destination node set $D_{x_{11}}$ into two subsets $D^1_{x_{11}}$ and $D^2_{x_{11}}$, such that $|D^1_{x_{11}}| = |D_{x_{11}}| + |D_3| - \frac{1}{3}k$. Let $D^2_{x_{11}} = D_{x_{11}} - D^1_{x_{11}}$. It follows that $|D^1_{x_{11}} \cup D_{x_{12}}| \in (\frac{2}{3}k, k)$ and $|D^2_{x_{11}} \cup D_3| = \frac{1}{3}k$. We route all the destination nodes in D_r by duplicating the branch rooted at u_1 and containing node x_{11}, and subtree T_4. One of the two copies of the duplicated branch rooted at u_1 is merged with the other branch also rooted at u_1 but containing node x_{12} to form a subtree, which is assigned as a destination node set $D^1_{x_{11}} \cup D_{x_{12}}$. The remainder subtree is re-rooted at node u_2, which has exactly four branches: Two of them are the same as those two branches rooted at u_2 in the original T_r, which have destination node sets $D_{x_{21}}$ and $D_{x_{22}}$, respectively; The third branch is an exact copy of T_4, which has a destination node set D_4; The last branch is the leftover subtree, which contains another exact copy of T_4 but with no destination node from D_4 assigned to it. That is, the last branch has a destination node set $D^2_{x_{11}} \cup D_3$. Since $|D_{x_{21}}| \le \frac{2}{3}k$, $|D_{x_{22}}| \le \frac{2}{3}k$, $|D_4| < \frac{1}{3}k$, and $|D^2_{x_{11}} \cup D_3| = \frac{1}{3}k$, we conclude that at least two of these four branches are assigned some destination nodes which are among the $\frac{2}{3}k$ closest ones from $D_{x_{21}} \cup D_{x_{22}} \cup D_4 \cup (D^2_{x_{11}} \cup D_3) = D_r - D^1_{x_{11}} - D_{x_{12}}$. Therefore, it is always possible to form two subtrees, each containing two branches, such that their sizes are both at most k and they both contain some destination nodes which are among the $\frac{2}{3}k$ closest ones. The connection cost of the first subtree is $\le \frac{3}{2}\times\frac{1}{k}\sum_{d\in D^1_{x_{11}}\cup D_{x_{12}}} w(s,d)$; The connection costs of the last two subtrees is at most $w(s,d') + w(s,d'')$, where d' and d'' are the closest and the $(\frac{2}{3}k+1)$-st farthest destination node from $D_r - D^1_{x_{11}} - D_{x_{12}}$. From $|D_r - D^1_{x_{11}} - D_{x_{12}}| > \frac{4}{3}k$, we have $w(s,d') + w(s,d'') \le \frac{3}{2}\times\frac{1}{k}\sum_{d\in D_r - D^1_{x_{11}} - D_{x_{12}}} w(s,d)$. Therefore, the total routing cost of these three subtrees is $\le \frac{5}{4}w(T_r) + \frac{3}{2}\times\frac{1}{k}\sum_{d\in D_r} w(s,d)$, and the lemma is proved. □

Lemma 3.6. *Suppose that T is a D-marked Steiner tree satisfying the following conditions:*

- $\frac{5}{2}k < |D \cap T| < \frac{8}{3}k$.
- *The root r of T has exactly two children v_1 and v_2.*
- $k < |D \cap T_{v_1}| < \frac{4}{3}k$ *and* $k < |D \cap T_{v_2}| < \frac{4}{3}k$.
- *For $i \in \{1,2\}$, there is a node u_i in T_{v_i} (possibly $u_i = v_i$) such that u_i has exactly two children $x_{i,1}$ and $x_{i,2}$ in T_{v_i}, $|D\cap T_{x_{i,1}}| < \frac{2}{3}k$, $|D \cap T_{x_{i,2}}| < \frac{2}{3}k$, and $|D\cap T_{x_{i,1}}| + |D \cap T_{x_{i,2}}| > k$.*

Given T, we can compute disjointly-D-marked Steiner trees $X_1,\ldots,X_p$ with $3 \le p \le 4$ in polynomial time such that each X_i ($1 \le i \le p$) is of size at most k, $D\cap T = \bigcup_{i=1}^{p}(D\cap X_i)$, and the total routing cost of $X_1,\ldots,X_p$ is at most $\frac{5}{4}w(T) + \frac{3}{2}\times\frac{1}{k}\sum_{d\in D\cap T} w(s,d)$.

Proof. We give two options to route all the destination nodes in $D \cap T$. In the first option, we obtain four disjointly-D-marked Steiner trees each of size at most k by applying Lemma 3.2 separately to T_{v_1} and to the branch rooted at r and containing v_2. The total routing cost of these four resultant trees is $w_1 \le w(T) + 2 \times \frac{1}{k} \sum_{d \in D \cap T} w(s, d)$.

We next describe the second option. For each $i \in \{1, 2\}$ and each $j \in \{1, 2\}$, let $B_{i,j}$ be the branch rooted at u_i and containing $x_{i,j}$. For each $i \in \{1, 2\}$, let B_i be the branch rooted at r and containing v_i. Let T_3 be the D-marked Steiner tree obtained from B_1 by deleting $x_{1,1}, x_{1,2}$, and their descendants. Similarly, let T_4 be the D-marked Steiner tree obtained from B_2 by deleting $x_{2,1}, x_{2,2}$, and their descendants. Clearly, $D \cap T = (D \cap T_{x_{1,1}}) \cup (D \cap T_{x_{1,2}}) \cup (D \cap T_3) \cup (D \cap T_4) \cup (D \cap T_{x_{2,1}}) \cup (D \cap T_{x_{2,2}})$. Moreover, $|D \cap T_3| \le |D \cap T_{v_1}| - |D \cap T_{x_{1,1}}| - |D \cap T_{x_{1,2}}| + 1 \le (\frac{4}{3}k - 1) - (k + 1) + 1 < \frac{1}{3}k$. Similarly, $|D \cap T_4| < \frac{1}{3}k$. We distinguish two cases as follows.

Case 1: $w(B_{1,2}) + w(B_{2,2}) \le \min\{w(T_3) + w(B_{1,1}), w(T_4) + w(B_{2,1})\}$. In this case, $w(B_{1,2}) + w(B_{2,2}) \le \frac{1}{3}w(T)$. We find a subset Q_1 of $D \cap T_{x_{1,2}}$ such that $|D \cap T_{1,1}| + |Q_1| = \lceil \frac{1}{3}|D \cap T| \rceil$. Set Q_1 exists because $|D \cap T_{x_{1,1}}| < \frac{2}{3}k$, $|D \cap T_{x_{1,2}}| < \frac{2}{3}k$, $|D \cap T_{x_{1,1}}| + |D \cap T_{x_{1,2}}| > k$, and $\frac{5}{6}k < \lceil \frac{1}{3}|D \cap T| \rceil \le k$. Similarly, we find a subset Q_2 of $D \cap T_{x_{2,2}}$ such that $|D \cap B_{2,1}| + |Q_2| = \lfloor \frac{1}{3}|D \cap T| \rfloor$. Set Q_2 exists because $|D \cap T_{x_{2,1}}| < \frac{2}{3}k$, $|D \cap T_{x_{2,2}}| < \frac{2}{3}k$, $|D \cap T_{x_{2,1}}| + |D \cap T_{x_{2,2}}| > k$, and $\frac{5}{6}k \le \lfloor \frac{1}{3}|D \cap T| \rfloor \le k$. We are now ready to construct three disjointly-D-marked Steiner trees X_1, X_2, and X_3 as follows. For $i \in \{1, 2\}$, we construct X_i by initializing it as T_{u_i}, then unmarking the nodes of $(D \cap T_{x_{i,2}}) - Q_i$, and further unmarking u_i if it is marked in T. We construct X_3 from T by deleting $x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}$, and all of their descendants, and further unmarking the nodes of $Q_1 \cup Q_2$. Note that $|D \cap X_3|$ is equal to $\lfloor \frac{1}{3}|D \cap T| \rfloor$ or $\lceil \frac{1}{3}|D \cap T| \rceil$. Obviously, the total routing cost of X_1, X_2, and X_3 is $w_2 \le \frac{4}{3}w(T) + \frac{6}{5} \times \frac{1}{k} \sum_{d \in D \cap T} w(s, d)$. Since $\min\{w_1, w_2\} \le \frac{1}{4}w_1 + \frac{3}{4}w_2 \le \frac{5}{4}w(T) + \frac{7}{5} \times \frac{1}{k} \sum_{d \in D \cap T} w(s, d)$. So, choosing the better option among the two proves the lemma.

Case 2: $w(B_{1,2}) + w(B_{2,2}) > \min\{w(T_3) + w(B_{1,1}), w(T_4) + w(B_{2,1})\}$. Without loss of generality, we assume that $w(T_4) + w(B_{1,1}) < w(T_3) + w(B_{2,1})$. Then, $w(T_4) + w(B_{1,1}) < \frac{1}{3}w(T)$. We find a subset Q_1 of $D \cap T_{x_{1,1}}$ such that $|Q_1| + |D \cap T_{x_{1,2}}| = k$. Note that $|(D \cap T_{x_{1,1}}) - Q_1| + |D \cap T_3| \le \frac{1}{3}k$ because $|D \cap T_{v_1}| < \frac{1}{3}k$. We are now ready to construct three disjointly-D-marked Steiner trees X_1, X_2, and X_3 as follows. We construct X_1 by initializing it as T_{u_1}, then unmarking the nodes of $(D \cap T_{x_{1,1}}) - Q_1$, and further unmarking u_1 if it is marked in T. Note that the connection cost of X_1 is $c(X_1) \le \frac{1}{k} \sum_{d \in ((D \cap T_{x_{1,1}}) - Q_1) \cup (D \cap T_{x_{1,2}})} w(s, d)$.

We construct X_2 and X_3 as follows. First, we obtain a D-marked Steiner tree Y from T by removing $x_{1,2}, x_{2,1}, x_{2,2}$, and all of their descendants, and further unmarking the nodes of $Q_1 \cup (D \cap T_4)$. Note that $|D \cap Y| = |(D \cap T_{x_{1,1}}) - Q_1| + |D \cap T_3| \le \frac{1}{3}k$. We then sort the nodes in $(D \cap T) - (D \cap X_1)$ in ascending order of their distances to s in G. Let F contain the last $\frac{3}{4}k$ nodes in the sorted sequence, and let C contain the other nodes in the sequence. Since $|(D \cap T) - (D \cap X_1)| \ge \frac{3}{2}k$, we have $|C| \ge \frac{3}{4}k$. We now look at the four sets: $D \cap T_{x_{2,1}}$, $D \cap T_{x_{2,2}}$, $D \cap T_4$, and $D \cap Y$. Each of the first two sets is of size at most $\frac{2}{3}k - 1$ while each of the last two sets is of size at most $\frac{1}{3}k$. Thus, at least two of the four sets contain at least one node of C. Consequently, we can always divide the four sets into two groups $\mathcal{G}_1$ and $\mathcal{G}_2$ that satisfy the following two conditions:

1. $\mathcal{G}_1$ contains $D \cap T_{x_{2,1}}$ and one of $D \cap T_4$ and $D \cap Y$, while $\mathcal{G}_2$ contains $D \cap T_{x_{2,2}}$ and the other of $D \cap T_4$ and $D \cap Y$. (*Comment:* The total size of sets in $\mathcal{G}_1$ is at most k and the total size of sets in $\mathcal{G}_2$ is at most k.)
2. At least one set in $\mathcal{G}_1$ contains a node of C and at least one set in $\mathcal{G}_2$ contains a node of C.

If $\mathcal{G}_1$ contains $D \cap T_4$, then we let X_2 be the union of $B_{2,1}$ and T_4 and let X_3 be the union of $B_{2,2}$ and Y; otherwise, we let X_2 be the union of $B_{2,1}$ and Y and let X_3 be the union of $B_{2,2}$ and T_4. In any case, we unmark u_2 in one of X_2 and X_3 if it is marked in T. By Condition 1, $|D \cap X_2| \le k$ and $|D \cap X_3| \le k$. By Condition 2, $(D \cap X_2) \cap C \ne \emptyset$ and $(D \cap X_3) \cap C \ne \emptyset$. Obviously, one of $D \cap X_2$ and $D \cap X_3$ contains d' which is the closest destination node to s among the nodes in $(D \cap X_2) \cup (D \cap X_3)$. We assume that $D \cap X_2$ contains d'; the other case is similar. It follows that the connection cost $c(X_2) \le w(s, d') \le \frac{4}{3} \times \frac{1}{k} \sum_{d \in C} w(s, d)$. Moreover, since $(D \cap X_3) \cap C \ne \emptyset$, the connection cost $c(X_3) \le w(s, d'')$ where d'' is the farthest destination node from s among the nodes in C. Furthermore, since $C \cap F = \emptyset$, $w(s, d'') \le w(s, d''')$ where d''' is the closest destination node to s among the nodes in F. Thus, $c(X_3) \le w(s, d''') \le \frac{4}{3} \times \frac{1}{k} \sum_{d \in F} w(s, d)$. Therefore, $c(X_2) + c(X_3) \le \frac{4}{3} \times \frac{1}{k} \sum_{d \in (D \cap T) - (D \cap X_1)} w(s, d)$. Consequently, the total routing cost of X_1, X_2, and X_3 is $w_2 \le \frac{4}{3}w(T) + \frac{4}{3} \times \frac{1}{k} \sum_{d \in D \cap T} w(s, d)$, because $w(X_1) + w(X_2) + w(X_3) = w(T) + w(B_{1,1}) + w(T_4) \le \frac{4}{3}w(T)$. Finally, since $\min\{w_1, w_2\} \le \frac{1}{4}w_1 + \frac{3}{4}w_2 \le \frac{5}{4}w(T) + \frac{3}{2} \times \frac{1}{k} \sum_{d \in D \cap T} w(s, d)$, choosing the better option among the two proves the lemma. □

4. A $(\frac{\sqrt{2089}+77}{80} + \frac{5}{4}\rho)$-approximation algorithm for kMTR

In the following transformation process that makes T^0 into a feasible k-tree routing, it iteratively cuts out a subtree T from T^0 (rooted at the source node s), such that T satisfies certain constraints; According to the size of T, one of the above lemmas is then applied to design the routing trees for all the destination nodes in T. We summarize these lemmas in Table 1, and define the priority for the operation on each tree that falls into the range of every lemma.

First of all, for the original T^0 or after certain iterations the remainder tree (still denoted as) T^0, if $|D \cap T^0| \le k$, no more operations need to be done and T^0 is used as the routing tree for its destination nodes; if $|D \cap T^0| > k$ but every branch rooted at the source node s and containing a child of s is of size at most k, then again no more operations need to be done, but every

Table 1
Summary of the lemmas and the priority of the operations processing these trees.

	Size of T	Priority of the operation dealing T
Lemma 3.1	$[\frac{2}{3}k, k]$	6
Lemma 3.3	$[\frac{4}{3}k, \frac{3}{2}k]$	4, or 2 if re-rooted
Lemma 3.4	$(\frac{3}{2}k, 2k]$	5, or 3 or 2 if re-rooted
Lemma 3.5	$(2k, \frac{5}{2}k]$	2
Lemma 3.6	$(\frac{5}{2}k, \frac{8}{3}k]$	2

such branch as used to routing the destination nodes therein. Such a "no" operation, without any of the tree partitioning lemmas, is defined as the operation having the highest priority, which is 8. In either of the two cases, T^0 is feasible k-tree routing for all its destination nodes, and the total routing cost is $w(T^0)$; T^0 is united to existing routing trees (from previous iterations, if any) to route all destination nodes in D.

The other case left for consideration is $|D \cap T^0| > k$ and some branch rooted at the source node s in T^0 contains more than k destination nodes. We define a *big* node in T^0 to be an internal node v in T^0 with $|D \cap T^0_v| > k$ (so s is big, for now), and define a *huge* node in T^0 to be an internal node v in T^0 with $|D \cap T^0_v| > 2k$. Note that a big node may be huge, or not. A big node in T^0 is *extremely big* if all its children in T^0 are not big. Similarly, a huge node in T^0 is *extremely huge* if all its children in T^0 are not huge. One may see that since s is not a destination node, it is not extremely big.

4.1. Priorities of operations applying to T^0

If T^0 has an internal node v that has at least three children, among which two of them x_1 and x_2 satisfy $|D\cap T^0_{x_1}|+|D\cap T^0_{x_2}| \le k$, then the following operation modifies T^0 by merging the two branches containing x_1 and x_2 into one. Such an operation has priority 7:

(7.1) Make a copy v_c of v (v_c is not a destination node, even if v is);
(7.2) Delete the edges (v, x_1) and (v, x_2);
(7.3) Add three edges (v, v_c), (v_c, x_1), and (v_c, x_2) so that v_c becomes a new child of v while x_1 and x_2 become the children of v_c. These three edges have weight 0, $w(v, x_1)$ and $w(v, x_2)$, respectively.

If T^0 has an internal node v with $\frac{2}{3}k \le |D \cap T^0_v| \le k$, then T^0_v is cut off from T^0 and used as a routing tree for $D \cap T^0_v$. Such an operation has priority 6. Note that if no priority-6 operation is applicable, then every extremely big node in T^0 has at least two children because $k > \frac{2}{3}k$.

If T^0 has an extremely big node u with at least three children, then the following operation, of priority 5, modifies T^0:

(5.1) Pick three arbitrary children v_1, v_2, and v_3 of u in T^0. One can easily verify that since no higher priority operations apply, $|D \cap T^0_{v_i}| < \frac{2}{3}k$ for each index i, and $|D \cap T^0_{v_i}| + |D \cap T^0_{v_j}| > k$ for every pair of distinct indices i and j.
(5.2) Let T be the union of these three branches rooted at u and containing v_1, v_2, and v_3, respectively. Cut T off from T^0 (if necessary) by leaving a copy of node u, which is a non-destination node even if the original u is, and apply Lemma 3.4 to route all its destination nodes.

Similarly, when no operation of priority 5 or higher applies, every extremely big node u in T^0 has exactly two children and hence satisfies that $k < |D \cap T^0_u| < \frac{4}{3}k$. In addition, every huge node in T^0 has a descendant that is a big-but-not-huge node. For otherwise, assume to the contrary that there is a huge node u which does not have any big-but-not-huge descendant; then inside this subtree T_u there is an extreme huge node v which has no big descendants at all; it follows that v is an extremely big node; due to the fact no operations of priority 5–7 apply, $k < |D \cap T^0_v| < \frac{4}{3}k$, contradicting the assumption that v is huge.

If T^0 has an extremely big node v such that the path from s to v contains a node u with $\frac{4}{3}k \le |D \cap T^0_u| \le \frac{3}{2}k$, then the following operation, of priority 4, modifies T^0:

(4.1) Cut T^0_u off from T^0 (if necessary) by leaving a copy of node u, which is a non-destination node even if the original u is, and re-root it at node v, denoted as T^0_v (as there is no confusion);
(4.2) Apply Lemma 3.3 to route all the destination nodes in T^0_v.

If T^0 has an extremely big node v such that the path from s to v contains a node u with $\frac{3}{2}k \le |D \cap T^0_u| \le 2k$, then an operation of priority 3 modifies T^0. Such an operation is very much the same as the above priority-4 operation, except that Lemma 3.4 replaces Lemma 3.3.

```
INPUT: A network $(G, s, D)$ and an integer $k \geq 1$.
OUTPUT: A $k$-tree routing in $(G, s, D)$.

1. Compute a Steiner tree $T^0$ on $\{s\} \cup D$, using the best known approximation algorithm;
2. Root $T^0$ at $s$;
3. While ($T^0$ is not empty) do:
        Apply an operation of the highest possible priority to $T^0$;
4. Output the $k$-tree routing.
```

Fig. 1. A high-level description of the approximation algorithm.

An operation of priority 2 modifies T^0 by locating an extremely huge node u:

(2.1) Find an extremely big node v_1 that is a descendant of u in T^0 (v_1 is not huge);
(2.2) Let u_1 be the child of u in T^0 that is either v_1 itself or an ancestor of v_1 in T^0; Clearly, since u_1 is not huge and no operations of priority higher than 2 apply, $|D \cap T^0_{u_1}| < \frac{4}{3}k$. It follows that u has at least two children in T^0.
(2.3) If every child u_2 of u in T^0, $u_2 \neq u_1$, satisfies that $|D \cap T^0_{u_2}| \leq \frac{2}{3}k$, then
 (2.3.1) Collect a maximal subset of children of u, including u_1, such that the number of destination nodes inside these corresponding branches, plus 1 if $u \in D$, is less than or equal to $2k$; Let T^0_u denote this subtree rooted at u;
 (2.3.2) Cut T^0_u off from T^0 (if necessary) by leaving a copy of node u to maintain the connectivity of the remainder tree, which is a non-destination node even if the original u is, and re-root it at node v_1, denoted as $T^0_{v_1}$ (as there is no confusion);
 (2.3.3) If $|D \cap T^0_{v_1}| > \frac{3}{2}k$, then apply Lemma 3.4 to route all the destination nodes in $T^0_{v_1}$; Otherwise, apply Lemma 3.3.
(2.4) If there exists a child u_2 of u, $u_2 \neq u_1$, satisfying that $|D \cap T^0_{u_2}| > \frac{2}{3}k$, then due to there being no operations of priority 6 we conclude that $|D \cap T^0_{u_2}| > k$ (and thus u_2 is big):
 (2.4.1) Let T^0_u denote the subtree rooted at u to contain only two branches containing u_1 and u_2 respectively; Since no operations of priority higher than 2 apply, $2k < |D \cap T^0_u| < \frac{8}{3}k$.
 (2.4.2) Cut T^0_u off from T^0 (if necessary) by leaving a copy of node u to maintain the connectivity of the remainder tree, which is a non-destination node even if the original u is;
 (2.4.3) If $|D \cap T^0_u| \leq \frac{5}{2}k$, then apply Lemma 3.5 to route all the destination nodes in T^0_u; Otherwise, apply Lemma 3.6.

When none of the above introduced operations, of priority 2–8, applies to T^0, we conclude that $k < |D \cap T^0| < \frac{4}{3}k$. Consequently, there is only one extremely big node u in T^0, and $u \neq s$. Let v_1 and v_2 be the only two children of u in T^0, and let v_3 be the parent of u in T^0 (v_3 could be s). The following operation has the least priority of 1:

(1.1) Re-root T^0 at u, such that u has exactly three children v_1, v_2, and v_3; One can easily check that $|D \cap T^0_{v_3}| < \frac{1}{3}k$.
(1.2) Among the nodes in $(D \cap T^0_{v_1}) \cup (D \cap T^0_{v_2})$, find the closest destination node d' to s; It follows that $w(s, d') < \frac{1}{k} \sum_{d \in (D \cap T^0_{v_1}) \cup (D \cap T^0_{v_2})} w(s, d)$.
(1.3) Assume without loss of generality that $d' \in T^0_{v_1}$. Partition T^0 into two subtrees, $T^0_{v_1}$ and the remainder by cutting $T^0_{v_1}$ off from T^0. Note that both subtrees have size $\leq k$, and their total routing cost is $\leq w(T^0) + \frac{1}{k} \sum_{d \in D \cap T^0} w(s, d)$.

4.2. Summary of the algorithm

A high-level description of the complete algorithm is depicted in Fig. 1.

Theorem 4.1. *kMTR admits a $(\frac{\sqrt{2089}+77}{80} + \frac{5}{4}\rho)$-approximation algorithm, where ρ is the best known performance ratio for approximating the SMT problem.*

Proof. Notice that when an operation of priority 8 or 1 is applied, the resultant remainder tree T^0 becomes empty and the algorithm terminates. In each of the other iterations, the algorithm applies an operation of priority 2–7 to cut a subtree T^0_u off from the base Steiner tree T^0 and route all its destination nodes accordingly. The following invariants are maintained:

- The total routing cost of T^0_u is $\leq \frac{5}{4} w(T^0_u) + \frac{\sqrt{2089}+77}{80} \times \frac{1}{k} \sum_{d \in D \cap T^0_u} w(s, d)$;
- All the subtrees cut off from T^0 and the remainder tree T^0 are edge-disjoint from each other, and every destination node appears in exactly one of these trees.

Therefore, the total routing cost of the output k-tree routing is $R \leq \frac{5}{4}w(T^0) + \frac{\sqrt{2089}+77}{80} \times \frac{1}{k}\sum_{d\in D} w(s,d) \leq \frac{5}{4}w(T^0) + \frac{\sqrt{2089}+77}{80}R^*$, where T^0 is the initial Steiner tree obtained before applying any operations and the last inequality follows from Eq. (2.1). Since $w(T^0) \leq \rho R^*$, we have $R \leq (\frac{5}{4}\rho + \frac{\sqrt{2089}+77}{80})R^*$.

One can see that with the base Steiner tree T^0 being computed and for every destination node its distance to the source node s being computed, the rest of the execution time of the approximation algorithm in Fig. 1 is simply linear in $|D|$, the number of destination nodes. This is true since every technical lemma takes time linear in the number of destination nodes in the subtree under consideration. Therefore, the running time of the complete algorithm is dominated by the running time of the ρ-approximation algorithm for the Steiner minimum tree problem, whose complexity could be prohibitively high [14]. □

Acknowledgements

The first author was supported by AICML, ALIDF and NSERC. The second author was supported in part by the Grant-in-Aid for Scientific Research of the Ministry of Education, Science, Sports and Culture of Japan, under Grant No. 20500021. The third author was supported by ALIDF, CFI and NSERC.

References

[1] K. Bharath-Kumar, J.M. Jaffe, Routing to multiple destinations in computer networks, IEEE Transaction on Communications 31 (1983) 343–351.
[2] Z. Cai, Improved algorithms for multicast routing and binary fingerprint vector clustering. Master's thesis, Department of Computing Science, University of Alberta, June 16, 2004.
[3] Z. Cai, Z.-Z. Chen, G.-H. Lin, L. Wang, An improved approximation algorithm for the capacitated multicast tree routing problem, in: Proceedings of the Second Annual International Conference on Combinatorial Optimization and Applications, COCOA 2008, in: LNCS, vol. 5165, 2008, pp. 286–295.
[4] Z. Cai, G.-H. Lin, G. Xue, Improved approximation algorithms for the capacitated multicast routing problem, in: Proceedings of the Eleventh International Computing and Combinatorics Conference, COCOON 2005, in: LNCS, vol. 3593, 2005, pp. 136–145.
[5] C. Gröpl, S. Hougardy, T. Nierhoff, H. J. Prömel, Approximation algorithms for the Steiner tree problem in graphs, in: D.-Z. Du, X. Cheng (Eds.), Steiner Trees in Industries, Kluwer Academic Publishers, 2001, pp. 235–279.
[6] J. Gu, X.D. Hu, X. Jia, M.-H. Zhang, Routing algorithm for multicast under multi-tree model in optical networks, Theoretical Computer Science 314 (2004) 293–301.
[7] J. Gu, X. D. Hu, M.-H. Zhang, Algorithms for multicast connection under multi-path routing model, Information Processing Letters 84 (2002) 31–39.
[8] C. Huitema, Routing in the Internet, Prentice Hall PTR, 2000.
[9] R. Jothi, B. Raghavachari, Approximation algorithms for the capacitated minimum spanning tree problem and its variants in network design, in: Proceedings of the 31st International Colloquium on Automata, Languages and Programming, ICALP 2004, in: LNCS, vol. 3142, 2004, pp. 805–818.
[10] F. Kuo, W. Effelsberg, J.J. Garcia-Luna-Aceves, Multimedia Communications: Protocols and Applications, Prentice Hall, Inc., 1998.
[11] R. Libeskind-Hadas, Efficient collective communication in WDM networks with a power budget, in: Proceedings of the Ninth IEEE Conference on Computer Communications and Networks, ICCCN 2000, 2000, pp. 612–616.
[12] G.-H. Lin, An improved approximation algorithm for multicast k-tree routing, Journal of Combinatorial Optimization 9 (2005) 349–356.
[13] E. Morsy, H. Nagamochi, An improved approximation algorithm for capacitated multicast routings in networks, Theoretical Computer Science 390 (2008) 81–91.
[14] G. Robins, A.Z. Zelikovsky, Improved Steiner tree approximation in graphs, in: Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2000, 2000, pp. 770–779.
[15] L.H. Sahasrabuddhe, B. Mukherjee, Multicast routing algorithms and protocols: A tutorial, IEEE Networks 14 (2000) 90–102.
[16] Z. Wang, J. Crowcroft, Quality-of-service routing for supporting multimedia applications, IEEE Journal on Selected Areas in Communications 14 (1996) 1228–1234.
[17] X. Zhang, J. Wei, C. Qiao, Constrained multicast routing in WDM networks with sparse light splitting, in: Proceedings of IEEE INFOCOM 2000, 2000, pp. 1781–1790.