An Improved Approximation Algorithm for the Capacitated Multicast Tree Routing Problem

Zhipeng Cai * Zhi-Zhong Chen † Guohui Lin * ‡ Lusheng Wang §

March 15, 2008

Abstract

The Capacitated Multicast Tree Routing Problem is considered, in which only a limited number of destination nodes are allowed to receive data in one routing tree and multiple routing trees are needed to send data from the source node to all destination nodes. The goal is to minimize the total cost of these routing trees. An improved approximation algorithm is presented, which has a worst case performance ratio of $\frac{8}{5} + \frac{5}{4}\rho$. Here ρ denotes the best approximation ratio for the Steiner Minimum Tree problem, and it is about 1.55 at the writing of the paper. This improves upon the previous best having a performance ratio of $2 + \rho$.

1 Introduction

Multicast consists of concurrently sending the same data from a single source node to multiple destination nodes. Such a service plays an important role in computer and communication networks supporting multimedia applications [7, 9, 13]. It is well known that multicast can be easily implemented on local area networks (LANs) since nodes connected to a LAN usually communicate over a broadcast network, yet quite challenging to implement in wide area networks (WANs) as nodes connected to a WAN communicate via a switched/routed network [4, 14].

In order to perform multicast communication in WANs, the source node and all the destination nodes must be interconnected. The problem of multicast routing in WANs is thus equivalent to finding a multicast tree in a network that spans the source and all the destination nodes, with its goal to minimize the *cost* of the multicast tree which is the total weight of edges in the tree.

In this paper, the *Capacitated Multicast Routing Problem* is studied in which only a limited number of destination nodes can be assigned to receive the packets sent from the source node during each transmission. The switches or routers in the underlying network are assumed to have the broadcasting ability. For simplicity, such a routing model is called the *multi-tree model* [6, 5]. Multi-tree model has its origin in WDM optical networks with limited light-splitting capabilities. Under this model, we are interested in finding a set of trees such that each tree spans the source node and a limited number of destination nodes that are assigned to receive data and every destination node must be designated to receive data in one of the trees. Compared with the traditional multicast routing model without the capacity constraint (called the *Steiner Minimum Tree* problem which

*Computing Science, University of Alberta. Edmonton, Alberta T6G 2E8, Canada. zhipeng@cs.ualberta.ca.
†Mathematical Sciences, Tokyo Denki University. Hatoyama, Saitama 350-0394, Japan. chen@r.dendai.ac.jp.
‡To whom correspondence should be addressed. ghlin@cs.ualberta.ca. Fax: (780) 492 1071.
§Computer Science, City University of Hong Kong. Tat Chee Avenue, Kowloon, Hong Kong. cswangl@cityu.edu.hk.

allows any number of receivers in the routing tree), this simpler model makes multicast easier and more efficient to be implemented, at the expense of increasing the cost of the routing tree. Specifically, when the number of destination nodes in a tree is limited to at most k, we call it the *Multicast k-Tree Routing* (kMTR) problem, which is formally defined in the following.

For a graph G, we denote its node set by $V(G)$. We model the underlying communication network 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 additive edge weight function $w(\cdot)$ generalizes to subgraphs of G in a natural way. That is, if T is a subgraph of G, then the *weight* (or *cost*) of T, 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 T is a tree, at least one node in T is marked, and each marked node in T is contained in D. For each D-marked Steiner tree T, we use $D \cap T$ to denote the set of marked nodes in T. Note that some nodes in both D and T may not be marked. The *size* of T is the number of marked nodes in 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 every two trees T_1 and T_2 in $\mathcal{T}$. Let k be a given positive integer. A *k-tree routing* in network (G, s, D) is a set $\{T_1, \ldots, T_\ell\}$ of disjointly-D-marked Steiner trees such that each T_i $(1 \le i \le \ell)$ contains s and is of size at most k and $D = \bigcup_{i=1}^{\ell}(D \cap T_i)$. The *weight* (or *cost*) of a k-tree routing is the total weight of trees in the routing. Given a network (G, s, D), the multicast k-tree routing (kMTR) problem asks for a k-tree routing in (G, s, D) whose weight is minimized over all k-tree routings in (G, s, D).

For the kMTR problem, the cases where $k = 1, 2$ can be solved efficiently [5]. The general case of kMTR, where k is not fixed, is NP-hard [4]. In [1, 10], kMTR is proven to be NP-hard when k is a fixed integer greater than 2. The best known approximation algorithm for kMTR ($k \ge 3$) has a worst case performance ratio of $(2 + \rho)$ [1, 2, 8], where ρ is the approximation ratio for the Steiner Minimum Tree problem, and it is about 1.55 [3, 12] at the writing of this paper. Recently, Morsy and Nagamochi presented an approximation algorithm for kMTR ($k \ge 3$) having a worst case performance ratio of $(\frac{3}{2} + \frac{4}{3}\rho)$ [11], which constitutes an improvement only when $\rho < 1.5$.

In this paper, we take advantage of the weight averaging technique introduced in [1, 2] to facilitate the design and analysis of a better approximation algorithm for kMTR. We extend another technique for partitioning routing trees in [1, 2] to guarantee better quality subtrees. Combining them, we achieve an $(\frac{8}{5} + \frac{5}{4}\rho)$-approximation algorithm. This improves upon the previous best approximation ratio of $(2 + \rho)$ [1, 2, 8]. It is also an improvement over $(\frac{3}{2} + \frac{4}{3}\rho)$ [11] as long as $\rho \ge 1.2$. In the next section, we present the tree partitioning process in details, the complete algorithm, and its performance analysis.

2 An $(\frac{8}{5} + \frac{5}{4}\rho)$-Approximation Algorithm for kMTR

Throughout this section, fix a positive integer k and an instance (G, s, D) of the kMTR problem. For ease of explanation, 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 nodes in $V(G) - (D \cup \{s\})$ can be used as intermediate nodes in a routing to save the routing cost.

For each pair (u, v) of nodes in G, we use $w(u, v)$ to denote the weight of the edge between u and v. 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. So, we can 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. Then, the edge weight function of G satisfies the triangle inequality.

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

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

In the following $(\frac{8}{5}+\frac{5}{4}\rho)$-approximation algorithm, we first apply the currently best approximation algorithm for the Steiner Minimum Tree 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 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, $w(T^0) \le \rho R^*$. We now root tree T^0 at source s. Note that tree T^0 does not necessarily correspond to a k-tree routing, because the subtree rooted at some child of s in T^0 may contain more than k marked nodes.

In the following, 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_u 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 nodes in T and the size of T is $|D\cap T|$. If $|D\cap T| \le k$, 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 $c(T) \le \min_{d\in D\cap T} w(s,d)$, we have

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

Although tree T^0 does not necessarily correspond to a k-tree routing, it serves as a good starting point because $w(T^0) \le \rho R^*$. 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 2.1. With these lemmas, we will define several types of operations in Section 2.2 that can be applied to T^0 (to turn it into a k-tree routing). An outline of the whole algorithm is given in Section 2.3.

2.1 Several Lemmas

This section proves several lemmas that will help us transform T^0 into a k-tree routing. Due to the space constraint, proofs of Lemmas 2.5 and 2.6 are moved to the Appendix.

Lemma 2.1 [1, 2] *Given a D-marked Steiner tree T such that*

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

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 2.2 *If T is a D-marked Steiner tree such that*

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

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

Proof. This is trivial since the size of T is at least $\frac{2}{3}k$, following Equation 2.2. □

Lemma 2.3 *Suppose that T is a D-marked Steiner tree satisfying the following conditions:*

- $\frac{3}{2}k \le |D \cap T| \le 2k$.
- *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 through 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. For each $i \in \{1,2,3\}$, let B_i be the branch rooted at r and containing v_i. We give two options to route all the destination nodes in $D \cap T$. To describe the first option, we assume that $|D \cap T_{v_1}| \ge |D \cap T_{v_2}|$ without loss of generality. We also unmark r in both B_2 and B_3 if it is marked in T. Then, $|D \cap T_{v_1}| \ge \frac{1}{2}k$ and $|D \cap T_{v_2}| + |D \cap T_{v_3}| \le \frac{3}{2}k$. Now, if $|D \cap T_{v_2}| + |D \cap T_{v_3}| \le k$, then as the first option, we set $X_1 = B_{v_1}$ and set X_2 to be the union of B_2 and B_3. Otherwise, as the first option, we set $X_1 = B_{v_1}$ and obtain X_2 and X_3 by applying Lemma 2.1 to the union of B_2 and B_3. In both cases, the total routing cost is clearly $w_1 \le w(T) + 2 \times \frac{1}{k}\sum_{d\in D\cap T} w(s,d)$ by Lemma 2.1 and Equation 2.2.

We next describe the second option of routing. Let B_i be the least weight branch among B_1, B_2, B_3. Without loss of generality, we may assume that $i \ne 1$. Let j be the integer in $\{2,3\} - \{i\}$. We first construct two separate D-marked Steiner trees Y_1 and Y_2, where Y_1 is the union of B_i and B_1 and Y_2 is the union of B_i and B_j. Note that $w(Y_1) + w(Y_2) = w(T) + w(B_i) \le \frac{4}{3}w(T)$ because $w(B_i) \le \min\{w(B_1), w(B_j)\}$. Let $\alpha = |D \cap T|$. Since $\frac{3}{2}k \le \alpha \le 2k$ and k is a multiple of 12, $\frac{3}{4}k \le \lfloor \frac{\alpha}{2} \rfloor \le k$ and $\frac{3}{4}k \le \lceil \frac{\alpha}{2} \rceil \le k$. We can partition $D \cap B_i$ into two disjoint sets Q_1 and Q_2 such that $\lfloor \frac{\alpha}{2} \rfloor \le |D \cap T_{v_1}| + |Q_1| \le \lceil \frac{\alpha}{2} \rceil$ and $\lfloor \frac{\alpha}{2} \rfloor \le |D \cap T_{v_j}| + |Q_2| \le \lceil \frac{\alpha}{2} \rceil$. If $i = 2$, then clearly Q_1 exists because $|D \cap T_{v_1}| + |D \cap T_{v_2}| > k \ge \lceil \frac{\alpha}{2} \rceil$ and $|D \cap T_{v_1}| \le \frac{2}{3}k - 1 \le \lfloor \frac{\alpha}{2} \rfloor$; consequently Q_2 also exists because Q_1 exists and $|D \cap T_{v_1}| + |D \cap T_{v_3}| + |D \cap B_2| = \alpha$. If $i = 3$ and $|D \cap T_{v_1}| \ge |D \cap T_{v_2}|$, then Q_1 exists because $|D \cap T_{v_1}| + |D \cap B_3| \ge \lceil \frac{\alpha}{2} \rceil$ and $|D \cap T_{v_1}| \le \frac{2}{3}k - 1 \le \lfloor \frac{\alpha}{2} \rfloor$; consequently Q_2 also exists. Similarly, if $i = 3$ and $|D \cap T_{v_1}| < |D \cap T_{v_2}|$, then Q_2 exists because $|D \cap T_{v_2}| + |D \cap B_3| \ge \lceil \frac{\alpha}{2} \rceil$ and $|D \cap T_{v_2}| \le \frac{2}{3}k - 1 \le \lfloor \frac{\alpha}{2} \rfloor$; consequently Q_1 also exists. Now, we obtain X_1 from Y_1 by unmarking all the nodes of Q_2 and obtain X_2 from Y_2 by unmarking all the nodes of Q_1. Since $|D \cap X_1| \ge \lfloor \frac{\alpha}{2} \rfloor \ge \frac{3}{4}k$ and $|D \cap X_2| \ge \lfloor \frac{\alpha}{2} \rfloor \ge \frac{3}{4}k$, the total routing cost of X_1 and X_2 is $w_2 \le \frac{4}{3}w(T) + \frac{4}{3} \times \frac{1}{k}\sum_{d\in D\cap T} w(s,d)$ by Equation 2.2.

Now, $\min\{w_1, w_2\} \le \frac{1}{4}w_1 + \frac{3}{4}w_2 \le \frac{5}{4}w(T) + \frac{3}{2} \times \frac{1}{k}\sum_{d\in D\cap T} w(s,d)$. So, choosing the better option between the two proves the lemma. □

Lemma 2.4 *Suppose that T is a D-marked Steiner tree satisfying the following conditions:*

- $\frac{5}{2}k \le |D \cap T| \le 3k$.

- *The root r of T has exactly two children v_1 and v_2.*
- $k < |D \cap T_{v_1}| \leq \frac{3}{2}k$ *and* $k < |D \cap T_{v_2}| \leq \frac{3}{2}k$.
- *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 $3 \leq p \leq 4$ in polynomial time such that each X_i $(1 \leq i \leq p)$ 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 through X_p is at most $\frac{5}{4}w(T) + \frac{8}{5} \times \frac{1}{k}\sum_{d \in D \cap T} w(s,d)$.

Proof. We give two options to route all the destination nodes in $D \cap T$. In the first option, we apply Lemma 2.1 to T_{v_1} and T_{v_2} separately to obtain four disjointly-D-marked Steiner trees of size at most k. The total routing cost of these trees is $w_1 \leq w(T) + 2 \times \frac{1}{k}\sum_{d \in D \cap T} w(s,d)$.

We next describe the second option of routing. 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 $x_{i,j}$. Without loss of generality, we assume that $w(B_{1,1}) + w(B_{2,1}) \leq w(B_{1,2}) + w(B_{2,2})$. We first construct three separate D-marked Steiner trees Y_1, Y_2, and Y_3 as follows. Y_1 is the union of $B_{1,1}$ and $B_{1,2}$, Y_2 is the union of $B_{2,1}$ and $B_{2,2}$, and Y_3 is obtained from T by deleting $x_{1,2}$, $x_{2,2}$, and their descendants. Note that $w(Y_1)+w(Y_2)+w(Y_3) = w(T) + w(B_{1,1}) + w(B_{2,1}) \leq \frac{3}{2}w(T)$. Let $\alpha = |D \cap T|$. Since $\frac{5}{2}k \leq \alpha \leq 3k$ and k is a multiple of 12, $\frac{5}{6}k \leq \lfloor\frac{\alpha}{3}\rfloor \leq k$ and $\frac{5}{6}k \leq \lceil\frac{\alpha}{3}\rceil \leq k$. Since $|D \cap T_{x_{1,2}}| < \frac{2}{3}k \leq \lfloor\frac{\alpha}{3}\rfloor$ and $|D \cap T_{x_{1,1}}| + |D \cap T_{x_{1,2}}| > k \geq \lceil\frac{\alpha}{3}\rceil$, we can compute a subset Q_1 of $D \cap B_{1,1}$ such that $|D \cap T_{x_{1,2}}| + |Q_1| = \lceil\frac{\alpha}{3}\rceil$. For a similar reason, we can compute a subset Q_2 of $D \cap B_{2,1}$ such that $|D \cap T_{x_{2,2}}| + |Q_2| = \lfloor\frac{\alpha}{3}\rfloor$. No matter what the value of α is, we always have that $\lfloor\frac{\alpha}{3}\rfloor \leq |D \cap T| - |D \cap T_{x_{1,2}}| - |Q_1| - |D \cap T_{x_{2,2}}| - |Q_2| \leq \lceil\frac{\alpha}{3}\rceil$. Now, we obtain X_1 from Y_1 by unmarking the nodes of $(D \cap B_{1,1}) - Q_1$, obtain X_2 from Y_2 by unmarking the nodes of $(D \cap B_{2,1}) - Q_2$, and obtain X_3 from Y_3 by unmarking the nodes of $Q_1 \cup Q_2$. Since $|D \cap X_i| \geq \lfloor\frac{\alpha}{3}\rfloor \geq \frac{5}{6}k$, the total routing cost of X_1 through X_3 is $w_2 \leq \frac{3}{2}w(T) + \frac{6}{5} \times \frac{1}{k}\sum_{d \in D \cap T} w(s,d)$.

Now, $\min\{w_1, w_2\} \leq \frac{1}{2}(w_1 + w_2) \leq \frac{5}{4}w(T) + \frac{8}{5} \times \frac{1}{k}\sum_{d \in D \cap T} w(s,d)$. Choosing the better option between the two proves the lemma. □

Lemma 2.5 *Suppose that T is a D-marked Steiner tree satisfying the following conditions:*

- $2k < |D \cap T| \leq \frac{5}{2}k$.
- *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 disjointly-D-marked Steiner trees X_1, X_2, and X_3 in polynomial time such that each X_i $(1 \leq i \leq 3)$ 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. See Appendix. □

Lemma 2.6 *Suppose that T is a D-marked Steiner tree satisfying the following conditions:*

- $\frac{4}{3}k \le |D \cap T| \le \frac{3}{2}k$.
- *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 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 $\frac{5}{4}w(T) + \frac{8}{5} \times \frac{1}{k}\sum_{d \in D \cap T} w(s,d)$.

PROOF. See Appendix. □

2.2 Operations to Be Applied to T^0

We are now ready to describe how to transform the initial Steiner tree T^0 (rooted at the source node s) into a k-tree routing. The transformation will be done by performing eight types of operations (namely, type-i operations with $i \in \{0, \ldots, 7\}$) on T^0 until T^0 becomes empty. When performing theses operations on T^0, we will maintain the following invariants:

(I1) A type-i operation is applied to T^0 only when no type-j operations with $j < i$ can be applied.

(I2) The source node s always remains in T^0.

We define a *big* node in T^0 to be an internal node v in T^0 with $|D \cap T^0_v| > k$, 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 in T^0 may be a huge node or not. A big node in T^0 is *extreme* if all its children in T^0 are not big. Similarly, a huge node in T^0 is *extreme* if all its children in T^0 are not huge.

We next proceed to the definition of the operations on T^0. A type-0 operation can be applied on T^0 if $|D \cap T^0| \le k$ or every branch rooted at s and containing a child of s is of size at most k. In the former case, a *type-0 operation* on T^0 includes T^0 in the output k-tree routing and then deletes the whole tree. In the latter case, a *type-0 operation* on T^0 includes each branch rooted at the root of T^0 (and containing a child of the root) in the output k-tree routing and then deletes the whole tree. In either case, the total routing cost equals $w(T^0)$ (i.e., no connection cost is needed when a type-0 operation is applied) because $s \in V(T^0)$ by Invariant (I2). Note that if no type-0 operations can be applied to T^0, then s is a big node in T^0 but is not an extreme big node in T^0 for $s \notin D$, implying that extreme big nodes always exist in T^0 and they are different from s.

If T^0 has an internal node v that has at least three children and has two children x_1 and x_2 with $|D \cap T^0_{x_1}| + |D \cap T^0_{x_2}| \le k$, then a *type-1 operation* modifies T^0 as follows:

1. Make a copy v_c of v (without marking v_c even if v is marked in T^0).
2. Delete the edges (v, x_1) and (v, x_2).
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. (*Comment*: (v, v_c) is a dummy edge of weight 0.)

If T^0 has an internal node v with $\frac{2}{3}k \le |D \cap T^0_v| \le k$, then a *type-2 operation* modifies T^0 as follows:

1. Include T^0_v in the output k-tree routing (cf. Lemma 2.2).

2. Remove v and all its descendants from T^0.

Note that if no type-2 operations can be applied to T^0, then every extreme big node in T^0 has at least two children because $k > k-1 \geq \frac{2}{3}k$.

If T^0 has an extreme big node u with at least three children, then a *type-3 operation* modifies T^0 as follows:

1. Pick three arbitrary children v_1, v_2, and v_3 of u in T^0. (*Comment*: Since u is an extreme big node in T^0 and no type-2 operations can be applied to T^0, $|D \cap T^0_{v_j}| < \frac{2}{3}k$ for each $j \in \{1,2,3\}$. Moreover, since no type-1 operations can be applied to T^0, $|D \cap T^0_{v_i}| + |D \cap T^0_{v_j}| > k$ for every pair (i,j) with $1 \leq i < j \leq 3$.)

2. Let T be the union of the three branches rooted at u and containing v_1, v_2, or v_3.

3. Use T to obtain a set of D-marked Steiner trees as described in Lemma 2.3, and include them in the output k-tree routing.

4. Remove v_1, v_2, v_3, and their descendants from T^0.

5. If u is marked in T^0, then unmark it in T^0.

Note that if neither type-2 nor type-3 operations can be applied to T^0, then every extreme big node v in T^0 has exactly two children and hence satisfies that $k < |D \cap T^0_v| < \frac{4}{3}k$. Moreover, we can claim that every huge node in T^0 has a descendant that is a big but not huge node, if neither type-2 nor type-3 operations can be applied to T^0. For a contradiction, assume that the claim does not hold. Then, there is an extreme huge node v in T^0 whose children are not big nodes. So, v is an extreme big node in T^0. Thus, $k < |D \cap T^0_v| < \frac{4}{3}k$, contradicting the assumption that v is huge.

If T^0 has an extreme big vertex v such that the path from s to v contains a node u with $\frac{4}{3}k \leq |D \cap T^0_u| \leq \frac{3}{2}k$, then a *type-4 operation* modifies T^0 as follows:

1. Construct a D-marked Steiner tree T by initializing it as T^0_u and re-rooting it at v.

2. Use T to obtain two D-marked Steiner trees as described in Lemma 2.6, and include them in the output k-tree routing.

3. Remove u and its descendants from T^0.

If T^0 has an extreme big node v such that the path from s to v contains a node u with $\frac{3}{2}k \leq |D \cap T^0_u| \leq 2k$, then a *type-5 operation* modifies T^0 in the same way as a type-4 operation does except that Lemma 2.3 is used instead of Lemma 2.6.

If T^0 has a huge node, then a *type-6 operation* modifies T^0 as follows:

1. Select an (arbitrary) extreme huge node u in T^0.

2. Find an extreme big node v_1 that is a descendant of u in T^0 (*Comment:* As claimed before, v_1 is big but not huge, implying that $v_1 \neq u$.)

3. Let u_1 be the child of u in T^0 that is v_1 itself or an ancestor of v_1 in T^0. (*Comment:* $|D \cap T^0_{u_1}| < \frac{4}{3}k$ because u_1 is not huge and neither type-4 nor type-5 operations can be applied to T^0. Consequently, u has at least two children in T^0.)

4. If every child u_2 of u in T^0 with $u_2 \neq u_1$ satisfies that $|D \cap T^0_{u_2}| \leq \frac{2}{3}k$, then modify T^0 as follows:

   (a) Construct a D-marked Steiner tree T by initializing it as T^0_u and then repeatedly deleting a child $u_2 \neq u_1$ and the descendants of u_2 until $|D \cap T| \leq 2k$. (*Comment:* $|D \cap T| \geq \frac{4}{3}k$ because $|D \cap T^0_{u_2}| < \frac{2}{3}k$ for each child u_2 of u in T^0 with $u_2 \neq u_1$.)

   (b) Re-root T at v_1.

   (c) If $|D \cap T| > \frac{3}{2}k$, then use T to obtain two or three D-marked Steiner trees as described in Lemma 2.3 and include them in the output k-tree routing. Otherwise, use T to obtain two D-marked Steiner trees as described in Lemma 2.6 and include them in the output k-tree routing.

   (d) Remove the nodes in $V(T) - \{u\}$ from T^0.

   (e) If u is marked in T^0, then unmark it in T^0.

5. If some child u_2 of u in T^0 with $u_2 \neq u_1$ satisfies that $|D \cap T^0_{u_2}| > \frac{2}{3}k$, then modify T^0 as follows:

   (a) Find an extreme big node v_2 in $T^0_{u_2}$. (*Comment:* Since u is an extreme huge node in T^0, $|D \cap T^0_{u_2}| \leq 2k$. Consequently, u_2 must be a big node in T^0 because $|D \cap T^0_{u_2}| > \frac{2}{3}k$ no type-2 operations can be applied to T^0. Moreover, $|D \cap T^0_{u_2}| < \frac{4}{3}k$ because neither type-4 nor type-5 operations can be applied to T^0. Possibly, $v_2 = u_2$.)

   (b) Construct a D-marked Steiner tree T by setting it to be the union of the two branches rooted at u and containing u_1 or u_2. (*Comment:* Clearly, $2k < |D \cap T| < \frac{8}{3}k$.)

   (c) If $|D \cap T| \leq \frac{5}{2}k$, then use T to obtain three D-marked Steiner trees as described in Lemma 2.5 and include them in the output k-tree routing. Otherwise, use T to obtain three or four D-marked Steiner trees as described in Lemma 2.4 and include them in the output k-tree routing.

   (d) Remove the nodes in $V(T) - \{u\}$ from T^0.

   (e) If u is marked in T^0, then unmark it in T^0.

Suppose that no type-i operations with $0 \leq i \leq 6$ can be applied to T^0. Then, $k < |D \cap T^0| < \frac{4}{3}k$. Consequently, there is only one extreme big node u in T^0. As mentioned before, s is a big but not extreme big node in T^0. So, $u \neq s$. Let v_1 and v_2 be the children of u in T^0, and let v_3 be the parent of u in T^0 (possibly, $v_3 = s$). Now, a *type-7 operation* modifies T^0 as follows:

1. Re-root T^0 at u (so that v_3 becomes a child of u, too). (*Comment:* $|D \cap T^0_{v_3}| < \frac{1}{3}k$ because $k < |D \cap T^0| < \frac{4}{3}k$ and $|D \cap T^0_{v_1}| + |D \cap T^0_{v_1}| > k$.)

2. Among the nodes in $(D \cap T^0_{v_1}) \cup (D \cap T^0_{v_2})$, find the closest node d' to s. (*Comment:* $w(s, d') < \frac{1}{k} \sum_{d \in (D \cap T^0_{v_1}) \cup (D \cap T^0_{v_2})} w(s, d)$.)

3. Let $i \in \{1, 2\}$ be the integer with $d' \in T^0_{v_i}$.

4. Include $T^0_{v_i}$ as a D-marked Steiner tree in the output k-tree routing. (*Comment:* $c(T^0_{v_i}) \leq w(s, d') < \frac{1}{k} \sum_{d \in D \cap T} w(s, d)$.)

5. Obtain a tree T by deleting v_i and its descendants from T^0. (*Comment:* $|D \cap T| < k$ because $|D \cap T^0_{v_3}| < \frac{1}{3}k$ and $|D \cap T^0_{v_j}| < \frac{2}{3}k$, where j is the integer in $\{1,2\} - \{i\}$.)

6. Include T as a D-marked Steiner tree in the output k-tree routing. (*Comment:* Since s remains in T^0 after Step 5, the connection cost of T^0 is 0. Thus, the total routing cost of $T^0_{v_i}$ and T is at most $w(T^0) + \frac{1}{k}\sum_{d \in D \cap T} w(s,d)$.)

7. Remove the whole tree T^0.

2.3 Summary of the Algorithm

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

INPUT: A network (G, s, D).
OUTPUT: a k-tree routing in (G, s, D).

1. Compute a Steiner tree T^0 on $\{s\} \cup D$, using the currently best approximation algorithm;
2. Root T^0 at s;
3. While (T^0 is not empty) do:
3.1. Let i be the smallest j such that a type-j operation, $0 \le j \le 7$, can be applied to T^0;
3.2. Perform a type-i operation on T^0;
4. Output the k-tree routing.

Figure 1: A high-level description of the $(\frac{8}{5} + \frac{5}{4}\rho)$-approximation algorithm for kMTR.

Theorem 2.7 kMTR ($k \ge 3$) *admits an* $(\frac{8}{5}+\frac{5}{4}\rho)$*-approximation algorithm, where* ρ *is the currently best performance ratio for approximating the* Steiner Minimum Tree *problem.*

PROOF. Notice that whenever we cut a subtree T out of the base Steiner tree T^0 by performing a type-i operation with $i \in \{0, \ldots, 7\}$, we maintain the following invariants:

- We construct a set $\mathcal{T}$ of disjointly-D-marked Steiner trees from T and include them in the output k-tree routing, where the total routing cost of the trees in $\mathcal{T}$ is at most $\frac{5}{4}w(T) + \frac{8}{5} \times \frac{1}{k}\sum_{d \in D \cap T} w(s,d)$.

- After cutting T out of T^0, T^0 may share a node with T but does not share an edge with T, and no node of $D \cap T$ is marked in T^0.

By the above invariants, the total routing cost of the trees in the output k-tree routing is $R \le \frac{5}{4}w(T^0) + \frac{8}{5} \times \frac{1}{k}\sum_{d \in D} w(s,d) \le \frac{5}{4}w(T^0) + \frac{8}{5}R^*$, where T^0 is the initial Steiner tree obtained in Step 1 of the algorithm and the last inequality follows from Equation 2.1. Since $w(T^0) \le \rho R^*$, we have $R \le (\frac{5}{4}\rho + \frac{8}{5})R^*$. □

References

[1] 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.

[2] Z. Cai, G.-H. Lin, and G. L. Xue. Improved approximation algorithms for the capacitated multicast routing problem. In *Proceedings of the Eleventh International Computing and Combinatorics Conference (COCOON 2005)*, LNCS 3593, pages 136–145. Springer, 2005.

[3] C. Gröpl, S. Hougardy, T. Nierhoff, and H. J. Prömel. Approximation algorithms for the Steiner tree problem in graphs. In D.-Z. Du and X. Cheng, editors, *Steiner Trees in Industries*, pages 235–279. Kluwer Academic Publishers, 2001.

[4] J. Gu, X. D. Hu, X. Jia, and M.-H. Zhang. Routing algorithm for multicast under multi-tree model in optical networks. *Theoretical Computer Science*, 314:293–301, 2004.

[5] J. Gu, X. D. Hu, and M.-H. Zhang. Algorithms for multicast connection under multi-path routing model. *Information Processing Letters*, 84:31–39, 2002.

[6] R. L. Hadas. Efficient collective communication in WDM networks. In *Proceedings of IEEE ICCCN 2000*, pages 612–616, 2000.

[7] C. Huitema. *Routing in the Internet*. Prentice Hall PTR, 2000.

[8] R. Jothi and B. Raghavachari. Approximation algorithms for the capacitated minimum spanning tree problem and its variants in network design. *ACM Transactions on Algorithms*, 1:265–282, 2005.

[9] F. Kuo, W. Effelsberg, and J. J. Garcia-Luna-Aceves. *Multimedia Communications: Protocols and Applications*. Prentice Hall, Inc., 1998.

[10] G.-H. Lin. An improved approximation algorithm for multicast k-tree routing. *Journal of Combinatorial Optimization*, 9:349–356, 2005.

[11] E. Morsy and H. Nagamochi. An improved approximation algorithm for capacitated multicast routings in networks. *Theoretical Computer Science*, 390:81–91, 2008.

[12] G. Robins and A. Z. Zelikovsky. Improved Steiner tree approximation in graphs. In *Proceedings of the* 11*th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2000)*, pages 770–779, 2000.

[13] Z. Wang and J. Crowcroft. Quality-of-service routing for supporting multimedia applications. *IEEE Journal on Selected Areas in Communications*, 14:1228–1234, 1996.

[14] X. Zhang, J. Wei, and C. Qiao. Constrained multicast routing in WDM networks with sparse light splitting. In *Proceedings of IEEE INFOCOM 2000*, pages 1781–1790, March 26–30, 2000.

A Proof of Lemma 2.5

Proof. 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 $x_{i,j}$. Without loss of generality, we assume that $w(B_{1,1}) \le \min\{w(B_{1,2}), w(B_{2,1}), w(B_{2,2})\}$ and $|D\cap T_{x_{2,1}}| \le |D\cap T_{x_{2,2}}|$. Then, $|D\cap T_{x_{2,2}}| > \frac{1}{2}k$. 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 partition $D\cap T_{x_{1,1}}$ into two disjoint sets Q_1 and Q_2 such that $\frac{2}{3}k \le |Q_1|+|D\cap T_{x_{1,2}}| \le k$ and $\frac{4}{3}k \le |D\cap T|-|Q_1|-|D\cap T_{x_{1,2}}| \le \frac{3}{2}k$. Since $|D\cap T_{x_{2,2}}| > \frac{1}{2}k$, we have $|Q_2|+|(D\cap T_3)\cup(D\cap T_4)|+|D\cap T_{x_{2,1}}| = |D\cap T|-|Q_1|-|D\cap T_{x_{1,2}}|-|D\cap T_{x_{2,2}}| < k$. Among the nodes in $(D\cap T)-Q_1-(D\cap T_{x_{1,2}})$, we find the $\frac{2}{3}k$ farthest nodes from s; let F be the set of them. Depending on whether $D\cap T_{x_{2,2}}$ is a subset of F or not, there are two possible cases to consider:

Case 1: $D\cap T_{x_{2,2}}$ is not a subset of F. In this case, we construct X_1 by initializing it as T_{u_1}, then unmarking the nodes of Q_2, and further unmarking u_1 if it is marked. We simply let $X_2 = T_{x_{2,2}}$. To obtain X_3, we first obtain a D-marked Steiner tree Y_3 from T by deleting $x_{1,2}$, $x_{2,2}$, and their descendants, and then unmarking the nodes of Q_1. Note that $w(X_1)+w(X_2)+w(X_3) = w(T) + w(B_{1,1}) \le \frac{5}{4}w(T)$. Obviously, $c(X_1) \le \frac{3}{2}\times\frac{1}{k}\sum_{d\in Q_1\cup(D\cap T_{x_{1,2}})} w(s,d)$ by Equation 2.2. Since both X_2 and X_3 contain some nodes in $(D\cap T)-F$, $c(X_2) \le \min_{d\in F} w(s,d)$ and $c(X_3) \le \min_{d\in F} w(s,d)$. So, one of X_2 and X_3 has a connection cost of $w(s,d')$ and the other has a connection cost of at most $w(s,d'')$, where d' is the closest node to s among the nodes in $(D\cap T)-Q_1-(D\cap T_{x_{1,2}})$ while d'' is the closest node to s among the nodes in F. Clearly, $w(s,d'') \le \frac{3}{2}\times\frac{1}{k}\sum_{d\in F} w(s,d)$. Moreover, $w(s,d') \le \frac{3}{2}\times\frac{1}{k}\sum_{d\in(D\cap T)-Q_1-(D\cap T_{x_{1,2}})-F} w(s,d)$ because $|D\cap T|-|Q_1|-|D\cap T_{x_{1,2}}|-|F| \ge \frac{2}{3}k$. So, $c(X_2)+c(X_3) \le \frac{3}{2}\times\frac{1}{k}\sum_{d\in(D\cap T)-Q_1-(D\cap T_{x_{1,2}})} w(s,d)$. Thus, the total routing cost of X_1, X_2, and X_3 is at most $\frac{5}{4}w(T)+\frac{3}{2}\times\frac{1}{k}\sum_{d\in D\cap T} w(s,d)$, and we are done.

Case 2: $D\cap T_{x_{2,2}}$ is a subset of F. In this case, we distinguish three subcases as follows.

Subcase 2.1: $w(B_{2,1}) \le w(B_{2,2})+w(T_3)+w(T_4)$. In this subcase, we give two options to route the nodes in $D\cap T$. In the first option, we first partition $D\cap T_{x_{1,1}}$ into two disjoint sets P_1 and P_2 such that $|P_1|+|D\cap T_{x_{1,2}}| = k$ and $Q_1 \subseteq P_1$. Note that $P_2 \subseteq Q_2$. We partition $D\cap T_{x_{2,1}}$ into two disjoint sets C_1 and C_2 such that C_1 consists of the $k-|D\cap T_{x_{2,2}}|$ closest nodes to s. We are now ready to construct 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 P_2, and further unmarking u_1 if it is marked. We simply let $X_2 = T_{x_{2,2}}$. To obtain X_3, we first obtain a D-marked Steiner tree Y_3 from T by deleting $x_{1,2}$, $x_{2,2}$, and their descendants, and then unmarking the nodes of P_1. Note that $w(X_1)+w(X_2)+w(X_3) = w(T)+w(B_{1,1})$. Obviously, $c(X_1) \le \frac{1}{k}\sum_{d\in P_1\cup(D\cap T_{x_{1,2}})} w(s,d)$ by Equation 2.2. Recall that $|D\cap T_{x_{2,2}}| > \frac{1}{2}k$. Among the nodes in $D\cap T_{x_{2,2}}$, we find the $\frac{1}{2}k$ farthest nodes from s; let F' be the set of them. Among the nodes in F', the closest one d' to s satisfies that $w(s,d') \le 2\times\frac{1}{k}\sum_{d\in F'} w(s,d)$. Thus, $c(X_2) \le w(s,d') \le 2\times\frac{1}{k}\sum_{d\in F'} w(s,d)$. Among the nodes in $(D\cap T_{x_{2,1}})\cup(D\cap T_{x_{2,2}})$, the closest one d'' to s does not belong to $D\cap T_{x_{2,2}}$ because $D\cap T_{x_{2,2}} \subseteq F$ and $|D\cap T_{x_{2,1}}|+|D\cap T_{x_{2,2}}| > k > |F|$. Thus, $d'' \in C_1$. Consequently, $c(X_3) \le w(s,d'') \le 2\times\frac{1}{k}\sum_{d\in(C_1\cup(D\cap T_{x_{2,2}}))-F'} w(s,d)$ because $|C_1\cup(D\cap T_{x_{2,2}})| = k$ and $|F'| = \frac{1}{2}k$. So, $c(X_2)+c(X_3) \le 2\times\frac{1}{k}\sum_{d\in C_1\cup(D\cap T_{x_{2,2}})} w(s,d)$. Hence, the total routing cost of X_1, X_2, and X_3

is $w_1 \le w(T) + w(B_{1,1}) + \frac{1}{k}\sum_{d\in P_1\cup(D\cap T_{x_{1,2}})} w(s,d) + 2\times\frac{1}{k}\sum_{d\in C_1\cup(D\cap T_{x_{2,2}})} w(s,d)$.

In the second option of routing, we construct X_1, X_2, and X_3 as follows. We construct X_1 by initializing it as the union of $B_{2,1}$ and $B_{2,2}$, then unmarking the nodes of C_2, and further unmarking u_2 if it is marked. Note that $|D\cap X_1| = k$. To obtain X_2 and X_3, we first construct a D-marked Steiner tree Y as follows. Initially, Y is the D-marked Steiner tree obtained from T by deleting $x_{2,2}$ and its descendants. The nodes of C_1 are then unmarked in Y. This completes the construction of Y. Note that $|X_1| + |Y| = w(T) + w(B_{2,1})$. Moreover, $k < |D\cap Y| = |D\cap T| - k \le \frac{3}{2}k$. So, we obtain X_2 and X_3 by applying Lemma 2.1 to Y. Then, the total routing cost of X_2 and X_3 is at most $w(Y) + 2\times\frac{1}{k}\sum_{d\in(D\cap T)-C_1-(D\cap T_{x_{2,2}})} w(s,d)$. Therefore, the total routing cost of X_1, X_2, and X_3 is $w_2 \le w(T) + w(B_{2,1}) + \frac{1}{k}\sum_{d\in C_1\cup(D\cap T_{x_{2,2}})} w(s,d) + 2\times\frac{1}{k}\sum_{d\in(D\cap T)-(C_1\cup(D\cap T_{x_{2,2}}))} w(s,d)$.

Because $w(B_{1,1}) \le w(B_{1,2})$, $w(B_{2,1}) \le w(B_{2,2}) + w(T_3) + w(T_4)$, and $w(T) = w(B_{1,1}) + w(B_{1,2}) + w(B_{2,1}) + w(B_{2,2}) + w(T_3) + w(T_4)$, we have $\min\{w_1, w_2\} \le \frac{1}{2}(w_1 + w_2) \le w(T) + \frac{1}{2}(w(B_{1,1}) + w(B_{2,1})) + \frac{1}{2}\times\frac{1}{k}\sum_{d\in P_1\cup(D\cap T_{x_{1,2}})\cup C_1\cup(D\cap T_{x_{2,2}})} w(s,d) + \frac{1}{k}\sum_{d\in D\cap T} w(s,d) \le \frac{5}{4}w(T) + \frac{3}{2}\times\frac{1}{k}\sum_{d\in D\cap T} w(s,d)$. So, choosing the better option between the two proves the lemma in this subcase.

Subcase 2.2: $w(B_{2,1}) > w(B_{2,2}) + w(T_3) + w(T_4)$ and $w(B_{1,1}) + w(T_4) \le w(B_{2,2}) + w(T_3)$. Since $w(B_{1,1}) \le w(B_{1,2})$, $w(B_{1,1}) + w(B_{2,2}) + w(T_3) + w(T_4) < w(B_{1,2}) + w(B_{2,1})$. Therefore, $w(B_{1,1}) + w(B_{2,2}) + w(T_3) + w(T_4) < \frac{1}{2}w(T)$. Consequently, $w(B_{1,1}) + w(T_4) < \frac{1}{4}w(T)$. We partition $D\cap T_{x_{1,1}}$ into two disjoint sets P_1 and P_2 such that $|P_1| = |D\cap T_{x_{1,1}}| + |D\cap T_3| - \frac{1}{3}k$. Clearly, $|P_2| + |D\cap T_3| = \frac{1}{3}k$. Moreover, $|P_1| \ge 2$ because $|P_1| \ge |D\cap T_{v_1}| - |D\cap T_{x_{1,2}}| - \frac{1}{3}k$, $|D\cap T_{v_1}| \ge k+1$, and $|D\cap T_{x_{1,2}}| \le \frac{2}{3}k - 1$. Furthermore, $\frac{2}{3}k < |P_1| + |D\cap T_{x_{1,2}}| \le k$ because $|D\cap T_{v_1}| \le |D\cap T_{x_{1,1}}| + |D\cap T_{x_{1,2}}| + |D\cap T_3| \le |D\cap T_{v_1}| + 1$ and $k+1 \le |D\cap T_{v_1}| \le \frac{4}{3}k - 1$. We construct X_1 by initializing it as the union of $B_{1,1}$ and $B_{1,2}$, then unmarking the nodes of P_2, and further unmarking u_1 if it is marked. Clearly, $\frac{2}{3}k < D\cap X_1 = P_1\cup(D\cap T_{x_{1,2}}) \le k$ and hence $c(X_1) \le \frac{3}{2}\times\frac{1}{k}\sum_{d\in D\cap X_1} w(s,d)$.

To construct X_2 and X_3, consider the D-marked Steiner tree Y obtained from T by deleting $x_{1,2}$, $x_{2,1}$, $x_{2,2}$, and their descendants and further unmarking the nodes of $P_1\cup(D\cap T_4)$. Note that $D\cap Y \subseteq P_2\cup(D\cap T_3)$ and hence $|D\cap Y| \le \frac{1}{3}k$. Also recall that $\frac{2}{3}k < D\cap X_1 \le k$. So, $(D\cap T) - (D\cap X_1) \ge k$. Among the nodes in $(D\cap T) - (D\cap X_1)$, we find the $\frac{2}{3}k$ closest nodes to s; let C be the set of them. Similarly, among the nodes in $(D\cap T) - (D\cap X_1)$, we find the $\frac{2}{3}k$ farthest nodes from s; let F be the set of them. A crucial point is that $C\cap F = \emptyset$. This holds because $|(D\cap T) - (D\cap X_1)| = (|P_2| + |D\cap T_3|) + |D\cap T_{v_2}| > \frac{1}{3}k + k = \frac{4}{3}k$. Now, consider 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. 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 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. Then, $c(X_2) \le w(s,d') \le \frac{3}{2} \times \frac{1}{k}\sum_{d \in C} w(s,d)$ because $C \subseteq (D \cap T) - (D \cap X_1) = (D \cap X_2) \cup (D \cap X_3)$. Moreover, since $(D \cap X_3) \cap C \ne \emptyset$, $c(X_3) \le w(s,d'')$ where d'' is the farthest 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 node to s among the nodes in F. Thus, $c(X_3) \le w(s,d''') \le \frac{3}{2} \times \frac{1}{k}\sum_{d \in F} w(s,d)$. Therefore, $c(X_2) + c(X_3) \le \frac{3}{2} \times \frac{1}{k}\sum_{d \in (D\cap T)-(D \cap X_1)} w(s,d)$. Consequently, the total routing cost of X_1, X_2, and X_3 is at most $\frac{5}{4}w(T) + \frac{3}{2} \times \frac{1}{k}\sum_{d \in D \cap T} w(s,d)$, because $w(X_1)+w(X_2)+w(X_3) = w(T)+w(B_{1,1})+w(T_4) \le \frac{5}{4}w(T)$. This establishes the lemma in this subcase.

Subcase 2.3: $w(B_{2,1}) > w(B_{2,2}) + w(T_3) + w(T_4)$ and $w(B_{1,1}) + w(T_4) > w(B_{2,2}) + w(T_3)$. In this subcase, the proof proceeds as in Subcase 2.2 except that we exchange the roles of $B_{1,1}$ and $B_{2,2}$, exchange the roles of $B_{1,2}$ and $B_{2,1}$, exchange the roles of $x_{1,1}$ and $x_{2,2}$, exchange the roles of $x_{1,2}$ and $x_{2,1}$, exchange the roles of u_1 and u_2, exchange the roles of v_1 and v_2, and exchange the roles of T_3 and T_4. This completes the proof of the lemma. □

B Proof of Lemma 2.6

Proof. By the conditions in the lemma, $|D \cap T_{v_3}| < \frac{1}{2}k$. Without loss of generality, we assume that $|D \cap T_{v_1}| \le |D \cap T_{v_2}|$. 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: $|D \cap T_{v_3}| + |D \cap T_{v_2}| \le k$. Then, $|D \cap T_{v_3}| + |D \cap T_{v_1}| \le k$. Among the nodes in $D \cap T$, we find the $\frac{2}{3}k$ closest nodes to s; let C be the set of them. Similarly, among the nodes in $D \cap T$, we find the $\frac{2}{3}k$ farthest nodes from s; let F be the set of them. Since $|D \cap T| \ge \frac{4}{3}k$, $F \cap C = \emptyset$. Moreover, since $|D \cap T_{v_i}| < \frac{2}{3}k$ for each $i \in \{1,2,3\}$, there are at least two integers $i \in \{1,2,3\}$ such that $(D \cap T_{v_i}) \cap C \ne \emptyset$. 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 unmark r if it is marked. Otherwise, we find an integer $i \in \{1,2\}$ with $(D \cap T_{v_i}) \cap C \ne \emptyset$, set $X_1 = B_i$ and construct X_2 by initializing it as the union of B_j and B_3 and further unmark r if it is marked, where j is the integer in $\{1,2\} - \{i\}$. In any case, $|D \cap X_1| \le k$, $|D \cap X_2| \le k$, $(D \cap X_1) \cap C \ne \emptyset$, and $(D \cap X_2) \cap C \ne \emptyset$. Obviously, one of $D \cap X_1$ and $D \cap X_2$ contains d' which is the closest node to s among the nodes in $D \cap T$. We assume that $D \cap X_1$ contains d'; the other case is similar. Then, $c(X_1) \le w(s,d') \le \frac{3}{2} \times \frac{1}{k}\sum_{d \in C} w(s,d)$. Moreover, since $(D \cap X_2) \cap C \ne \emptyset$, $c(X_2) \le w(s,d'')$ where d'' is the farthest 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 node to s among the nodes in F. Thus, $c(X_2) \le w(s,d''') \le \frac{3}{2} \times \frac{1}{k}\sum_{d \in F} w(s,d)$. Therefore, $c(X_1) + c(X_2) \le \frac{3}{2} \times \frac{1}{k}\sum_{d \in D \cap T} w(s,d)$. 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: $|D \cap T_{v_3}| + |D \cap T_{v_2}| > k$. Then, $|D \cap T_{v_1}| + |D \cap T_{v_2}| > k$. We assume that $w(B_1) \le w(B_3)$; 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 $|D \cap T_{v_3}| < \frac{1}{2}k$. Among the nodes in $D \cap T_{v_1}$, we find the $k - |D \cap T_{v_2}|$ farthest nodes from s; let F be the set of them. Moreover, among the nodes in $(D \cap T) - F - (D \cap T_{v_2})$, we find the $\frac{1}{3}k$ closest nodes to s; let C be the set of them. Note that C exists because $|D \cap T| \ge \frac{4}{3}k$ and $|F| + |D \cap T_{v_2}| = k$. We give two options of 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 $D \cap T \le \frac{3}{2}k$. Moreover, since

$C \cup F \subseteq (D \cap T) - (D \cap T_{v_2})$ and $|C| + |F| + |D \cap T_{v_2}| = \frac{4}{3}k$, the total routing cost of X_1 and X_2 is $w_1 \le w(T) + \frac{1}{4k/3 - |D \cap T_{v_2}|} \sum_{d \in C \cup F} w(s,d) + \frac{1}{|D \cap T_{v_2}|} \sum_{d \in D \cap T_{v_2}} w(s,d)$.

In the second option, we first partition set F into two disjoint sets F_1 and F_2 such that $|F_2| + |D \cap T_{v_2}| = \frac{4}{3}k - |D \cap T_{v_2}|$. This can be done because $\frac{1}{2}k < |D \cap T_{v_2}| < \frac{2}{3}k$. We construct X_1 by initializing it as the union of B_1 and B_2, unmarking the nodes in $(D \cap T_{v_1}) - F_2$, and further unmarking r if it is marked. We construct X_2 by initializing it as the union of B_1 and B_3 and further unmarking the nodes in $(D \cap T_{v_2}) \cup F_2$. Clearly, $|D \cap X_1| \le \frac{5}{6}k$ and $|D \cap X_2| \le \frac{5}{6}k$. Moreover, since $|F_1| + |C| = |D \cap T_{v_2}|$ and $F_1 \cup C \subseteq D \cap X_2$, the total routing cost of X_1 and X_2 is $w_2 \le \frac{3}{2}w(T) + \frac{1}{4k/3 - |D \cap T_{v_2}|} \sum_{d \in (D \cap T_{v_2}) \cup F_2} w(s,d) + \frac{1}{|D \cap T_{v_2}|} \sum_{d \in F_1 \cup C} w(s,d)$.

Now, $\min\{w_1, w_2\} \le \frac{1}{2}(w_1 + w_2) \le \frac{5}{4}w(T) + \frac{1}{2} \times (\frac{1}{4k/3 - |D \cap T_{v_2}|} + \frac{1}{|D \cap T_{v_2}|})(\sum_{d \in F \cup (D \cap T_{v_2}) \cup C} w(s,d)$ because $\frac{4}{3}k - |D \cap T_{v_2}| \ge |D \cap T_{v_2}|$. Since $\frac{1}{2}k < |D \cap T_{v_2}| < \frac{2}{3}k$, $\frac{1}{4k/3 - |D \cap T_{v_2}|} + \frac{1}{|D \cap T_{v_2}|} \le \frac{16}{5}k$. Thus, $\min\{w_1, w_2\} \le \frac{5}{4}w(T) + \frac{8}{5} \times \frac{1}{k} \sum_{d \in D \cap T} w(s,d)$. Therefore, choosing the better option between the two proves the lemma. □