Improved Approximation Algorithms for the Quality of Service Multicast Tree Problem*

Marek Karpinski[1], Ion I. Măndoiu[2], Alexander Olshevsky[3], and Alexander Zelikovsky[4]

[1] Department of Computer Science, University of Bonn, Bonn 53117, Germany
marek@theory.cs.uni-bonn.de

[2] Department of Computer Science and Engineering, University of Connecticut
371 Fairfield Rd., Unit 2155, Storrs, CT 06269-2155, USA
ion@cse.uconn.edu

[3] Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Boston, MA 02139-4307
alex_o@mit.edu

[4] Computer Science Department, Georgia State University, Atlanta, GA 30303
alexz@cs.gsu.edu

Abstract. The Quality of Service Multicast Tree Problem is a generalization of the Steiner tree problem which appears in the context of multimedia multicast and network design. In this generalization, each node possesses a rate and the cost of an edge with length l in a Steiner tree T connecting the source to non-zero rate nodes is $l \cdot r_e$, where r_e is the maximum node rate in the component of $T - \{e\}$ that does not contain the source. The best previously known approximation ratios for this problem (based on the best known approximation factor of 1.549 for the Steiner tree problem in networks) are 2.066 for the case of two non-zero rates and 4.212 for the case of unbounded number of rates. In this paper we give improved approximation algorithms with ratios of 1.960 and 3.802, respectively. When the minimum spanning tree heuristic is used for finding approximate Steiner trees, then the previously best known approximation ratios of 2.667 for two non-zero rates and 5.542 for unbounded number of rates are reduced to 2.414 and 4.311, respectively.

Keywords: Approximation algorithm, interconnection network, multicast, quality of service, Steiner tree

* A preliminary version of this article has appeared in the procedings of the *8th International Workshop on Algorithms and Data Structures (WADS 2003)*, volume 2748 of Springer-Verlag's *Lecture Notes in Computer Science*, pp. 401–411. Most of IIM's work was done while he was with the Electrical and Computer Engineering Department, University of California at San Diego. Most of AO's work was done while he was an undergraduate student in Electrical Engineering at Georgia Institute of Technology, Atlanta, GA.

1 Introduction

The Quality of Service Multicast Tree (QoSMT) problem appears in two different contexts: multimedia distribution for users with different bitrate requests [8] and the general design of interconnection networks with different grade of service requests [7]. The problem was formulated as a natural generalization of the Steiner tree problem under the names “Multi-Tier Steiner Tree Problem” [9] and “Grade of Service Steiner Tree Problem” [14]. More recently, the problem has been considered by [5, 8] in the context of multimedia distribution. This problem generalizes the Steiner tree problem in that each node possesses a rate and the cost of a link is not constant but depends both on the cost per unit of transmission bandwidth and the maximum rate routed through the link.

Formally, the QoSMT problem can be stated as follows (see [5]). Let $G = (V, E, l, r)$ be a graph with two functions, $l : E \to R^+$ representing the length of each edge, and $r : V \to R^+$ representing the rate of each node. Let $\{r_0 = 0, r_1, r_2, \dots r_N\}$ be the range of r and S_i be the set of all nodes with rate r_i. The QoSMT problem asks for a minimum cost subtree T of G spanning a given source node s and nodes in $\bigcup_{i\geq 1} S_i$, all of which are referred to as *terminals*. The cost of an edge e in T is $\text{cost}(e) = l(e)r_e$, where r_e, called the *rate of edge* e, is the maximum rate in the component of $T - \{e\}$ that does not contain the source. Note that the nodes in S_0, i.e., zero rate nodes, are not required to be connected to the source s, but may serve as Steiner points for the output tree T.

The QoSMT problem is equivalent to the Grade of Service Steiner Tree Problem (GOSST) [14], which has a slightly different formulation. In GOSST there is no source node and edge rates r_e should be assigned such that the minimum edge rate on the tree path from a terminal with rate r_i to a terminal with rate r_j is at least $min(r_i, r_j)$. It is not difficult to see that these two formulations are equivalent. Indeed, an instance of QoSMT can be transformed into an instance of GOSST by assigning the highest rate to the source. The cost of an edge will remain the same, since each edge e in a tree T will be on the path from the source to the node of the highest rate in the component of $T - \{e\}$ that does not contain the source. Conversely, an instance of GOSST can be transformed into a QoSMT by giving source status to any node with the highest rate.

The QoSMT/GOSST problem was studied before in several contexts. Current et al. [7] gave an integer programming formulation for the problem and proposed a heuristic algorithm for its solution. Colbourn and Xue [6] presented an $O(r^3 n)$ time algorithm for solving the problem on a series-parallel graph, where n is the number of nodes and r is the number of grades of service (distinct rates). Some results for the case of few rates were obtained by Balakrishnan et al. in [1] and [2]. Specifically, [2] (see also [14]) suggested an algorithm for the case of two non-zero rates with approximation ratio of $\frac{4}{3}\alpha < 2.066$, where $\alpha < 1.550$ is the best approximation ratio of an algorithm for the Steiner tree problem. Recently, Charikar et al. [5] gave the first constant-factor approximation algorithm for an unbounded number of rates. They achieved an approximation ratio of $e\alpha < 4.212$.

Steiner Tree Algorithm	LCA [12]	LCA +RNS[11]	BR [15, 3]	MST [13]
Runtime	polynomial	polynomial	$O(n^3)$ [16]	$O(n \log n + m)$ [10]
Best previous approximation ratio [2, 14]	$\frac{4}{3}\frac{1+\ln 3}{2} + \epsilon$ $< 2.066 + \epsilon$	$\frac{20}{9} + \epsilon$ $< 2.223 + \epsilon$	$\frac{22}{9}$ < 2.445	$\frac{8}{3}$ < 2.667
Improved ratio	–	1.960+ϵ	2.237	2.414

Table 1. QoSMT problem with 2 rates. Runtime and approximation ratios of previously known algorithms and of the algorithms given in this paper. In the runtime, n and m denote the number of nodes and edges in the original graph $G = (V, E)$, respectively. Approximation ratios associated with polynomial-time approximation schemes are accompanied by a $+\epsilon$ to indicate that they approach the quoted value from above and do not reach this value in polynomial time.

Steiner Tree Algorithm	LCA [12]	RNS[11]	BR [15, 3]	MST [13]
Best previous approximation ratio [11]	$e\frac{1+\ln 3}{2} + \epsilon$ $< 4.212 + \epsilon$	$e\frac{5}{3} + \epsilon$ $< 4.531 + \epsilon$	$e\frac{11}{6}$ < 4.984	$2e$ < 5.44
Improved ratio	–	3.802 + ϵ	4.059	4.311

Table 2. QoSMT problem with an arbitrary number of rates. Approximation ratios of previously known algorithms and of the algorithms given in this paper.

In this paper we give algorithms with improved approximation factors. Our algorithms have an approximation ratio of 1.960 when there are two non-zero rates and an approximation ratio of 3.802 when there is an unbounded number of rates. The improvement comes from the reuse of higher rate edges in establishing connectivity for lower rate nodes. We give the first analysis of the gain resulting from such reuse, critically relying on approximation algorithms for computing k-restricted Steiner trees. To improve solution quality, we use different Steiner tree algorithms at different stages of the computation. In particular, we use both the Steiner tree algorithm in [12] which has the currently best approximation ratio and the algorithm in [11] which has the best approximation ratio among Steiner tree algorithms producing 3-restricted trees.

Tables 1 and 2 summarize the results of this paper. They present previously known approximation ratios using various Steiner tree approximation algorithms. Note that along with the best approximation ratios resulting from the use of the loss-contracting algorithm in [12], we also give approximation ratios resulting from the use of the algorithm in [11] and the more practical algorithms in [3, 13, 15].

The rest of the paper is organized as follows. In the next section, we tighten the analysis given in [4] for the k-restricted Steiner ratio. In Section 3, we introduce the so called β-convex Steiner tree approximation algorithms and tighten their performance bounds. We give the improved approximation algorithms for QoSMT problem with two, respectively unbounded number of non-zero rates in Sections 4 and 5. Finally, we conclude in Section 6.

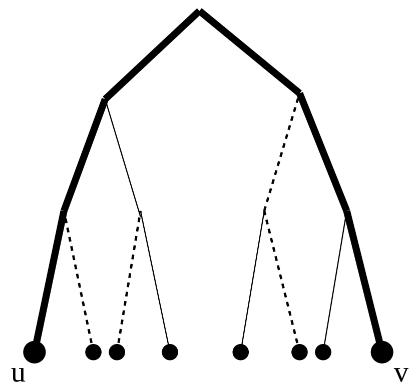

Fig. 1. Optimal Steiner tree T represented as a complete binary tree. Extreme terminals u and v form a diametrical pair of terminals, extreme edges (the path between u and v) are shown thicker. Each L_i represents the total length of a collection of paths (e.g., dashed paths) connecting internal nodes of T to non-extreme terminals via non-extreme edges.

2 A Tighter Analysis of the k-restricted Steiner Ratio

In this section, we tighten the analysis given in [4] for the k-restricted Steiner ratio. The tightened results will be used later to prove the approximation ratio of our algorithms. The exposition begins with a claim from [4] which encapsulates several of the proofs provided in that paper. This claim is then used in a manner slightly different from [4] to arrive at a stronger result.

We begin by introducing some definitions. A Steiner tree is called *full* if every terminal is a leaf. A Steiner tree can be decomposed into components which are full by breaking the tree up at the non-leaf terminals. A Steiner tree is called *k-restricted* if every full component has at most k terminals. Let us denote the length of the optimum k-restricted Steiner tree as opt_k and the length of the optimum unrestricted Steiner tree as opt. By duplicating nodes and introducing zero length edges, it can be assumed that a Steiner tree T is a complete binary tree (see Figure 1). Furthermore, we may assume that the leftmost and rightmost terminals form a diametrical pair of terminals, i.e., the distance between the leftmost and the rightmost terminals is the largest distance between any two vertices in the graph. The leftmost and rightmost terminals will be called *extreme terminals*, and the edges on the path between them will be called *extreme edges*.

Let the k-restricted Steiner ratio ρ_k be $\rho_k = sup\frac{opt_k}{opt}$, where the supremum is taken over all instances of the Steiner tree problem. It has been shown in [4] that $\rho_k = \frac{(r+1)2^r+s}{r2^r+s}$, where r and s are obtained from the decomposition $k = 2^r + s$, $0 \leq s < 2^r$.

Lemma 1. *[4] Given a Steiner tree T, there exist k-restricted Steiner trees T_i, $i = 1, 2, \ldots, r2^r + s$ such that $l(T_i) = l(T) + L_i$, where each L_i represents the total length of a collection of paths connecting internal nodes of T to non-extreme terminals via non-extreme edges in such a way that each non-extreme edge of T is counted at most 2^r times in the sum $L_1 + L_2 + \cdots + L_{r2^r+s}$.*

The claim in [4] is stated for an optimum Steiner tree T, but optimality is not needed in the proof.

We now use Lemma 1 to obtain a tighter bound on the length of the optimal k-restricted Steiner tree.

Theorem 1. *For every full Steiner tree T, $opt_k \leq \rho_k(l-D)+D$, where $l = l(T)$ is the length of T and $D = D(T)$ is the length of the longest path in T.*

Proof. Lemma 1 implies that $L_1 + L_2 + \cdots + L_{r2^r+s} \leq 2^r(l - D)$. From this it follows that there exists L_m such that $L_m \leq \frac{2^r}{r2^r+s}(l-D)$. Since $l(T_m) = l + L_m$, it follows that $l(T_m) \leq l + \frac{2^r}{r2^r+s}(l - D)$. Therefore,

$$\begin{aligned} opt_k &\leq l(T_m) \\ &\leq l + \frac{2^r}{r2^r + s}(l - D) \\ &= \left(1 + \frac{2^r}{r2^r + s}\right)(l - D) + D \\ &= \rho_k(l - D) + D \end{aligned}$$

We now strengthen this theorem to the case of partitioned trees.

Corollary 1. *For every Steiner tree T partitioned into edge-disjoint full components T^i,*

$$opt_k \leq \sum_i \left(\rho_k(l(T^i) - D(T^i)) + D(T^i)\right)$$

Proof. Let opt_k^i be the length of the optimal k-restricted tree for the full component T^i. Then,

$$opt_k \leq \sum_i opt_k^i \leq \sum_i \left(\rho_k(l(T^i) - D(T^i)) + D(T^i)\right)$$

3 β-Convex Steiner Tree Approximation Algorithms

In this section we introduce β-convex α-approximation Steiner tree algorithms and show tighter upper bounds on their output when applied to the QoSMT problem.

Definition 1. *A Steiner tree heuristic A is called a* β-convex α-approximation Steiner tree algorithm *if there exist an integer m and non-negative real numbers*

λ_i, $i = 2, \ldots, m$, with $\beta = \sum_{i=2}^{m} \lambda_i$ and $\alpha = \sum_{i=2}^{m} \lambda_i \rho_i$ such that the length of the tree computed by A, $l(A)$, is upper bounded by

$$l(A) \leq \sum_{i=2}^{m} \lambda_i opt_i,$$

where opt_i is the length of the optimal i-restricted Steiner tree.

The MST-algorithm [13] is 1-convex 2-approximation since its output is the optimal 2-restricted Steiner tree of length opt_2. Every k-restricted approximation algorithm from [3] is 1-convex – the sum of coefficients in the approximation ratio always equals to 1, e.g., for $k = 3$, it is 1-convex 11/6-approximation algorithm since the output tree is bounded by $\frac{1}{2}opt_2 + \frac{1}{2}opt_3$. The output tree for PTAS [11] converges to the optimal 3-restricted Steiner tree and has length $(1+\epsilon)opt_3$, therefore, it is $(1+\epsilon)$-convex $\frac{5}{3}(1+\epsilon)$-approximation algorithm. The currently best approximation ratio of $1 + \frac{\sqrt{3}}{2}$ is achieved by heuristic from [12] which is not known to be β-convex for any value of β.

Given a β-convex α-approximation algorithm A, it follows from Theorem 1 that

$$l(A) \leq \sum_i \lambda_i opt_i \leq \sum_i \lambda_i \rho_i (opt - D) + \beta D = \alpha(opt - D) + \beta D \qquad (1)$$

Let OPT be the optimum cost QoSMT tree T, and let t_i be the length of rate r_i edges in T. Then,

$$cost(OPT) = \sum_{i=1}^{N} r_i t_i$$

Let OPT_k be the subtree of the optimal QoS Multicast tree OPT induced by edges of rate r_i, $i \geq k$. The tree OPT_k spans the source s and all nodes of rate r_k and, therefore, an optimal Steiner tree connecting s and rate-r_k nodes cannot be longer than

$$l(OPT_k) = \sum_{i=k}^{N} t_i$$

The main idea of the proposed algorithms for the QoSMT problem is to reuse connections for the higher rate nodes when connecting lower rate nodes. When connecting nodes of rate r_k, we collapse nodes of rate strictly higher than r_k into the source s thus allowing to reuse higher rate connections for free. Let T_k be an approximate Steiner tree connecting the source s and all nodes of rate r_k after collapsing all nodes of rate strictly higher than r_k into the source s and treating all nodes of rate lower than r_k as Steiner points. If we apply an α-approximation Steiner tree algorithm for finding T_k, then the resulted length can be bounded as follows

$$l(T_k) \leq \alpha l(OPT_k) = \alpha t_k + \alpha t_{k+1} + \ldots + \alpha t_N$$

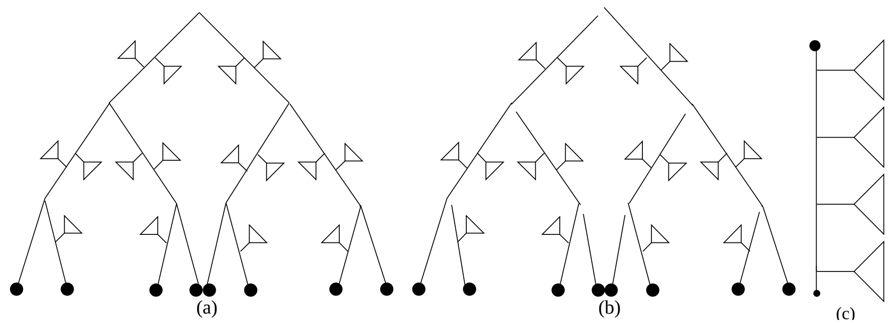

Fig. 2. (a) The subtree OPT_k of the optimal QoS Multicast tree OPT induced by edges of rate r_i, $i \geq k$. Edges of rate greater than r_k (shown as solid lines) form a Steiner tree for $s \cup S_{k+1} \cup \ldots S_N$ (filled circles); attached triangles represent edges of rate r_k. (b) Partition of OPT_k into edge-disjoint connected components OPT_k^i each containing a single terminal of rate r_i, $i > k$. (c) A connected component OPT_k^i which consists of a path D_k^i containing all edges of rate r_i, $i > k$, and attached Steiner trees containing edges of rate r_k.

The following lemma shows that if the tree T_k is obtained using β-convex α-approximation Steiner tree algorithm, then we can obtain a tighter upper bound on the length of T_k.

Lemma 2. *Given an instance of the QoSMT problem, the cost of the tree T_k computed by a β-convex α-approximation Steiner tree algorithm is at most*

$$cost(T_k) \leq \alpha r_k t_k + \beta(r_k t_{k+1} + r_k t_{k+2} + \cdots + r_k t_N)$$

Proof. Let OPT_k be the subtree of the optimal QoS Multicast tree OPT induced by edges of rate r_i, $i \geq k$. By duplicating nodes and introducing zero length edges, it can be assumed that OPT_{k+1} is a complete binary tree with the set of leaves consisting of the source s and all nodes of rate at least r_{k+1}. The edges of rate r_k form subtrees attached to the tree OPT_{k+1} connecting rate r_k nodes to OPT_{k+1} (see Figure 2(a)).

Note that edges of any binary tree T can be partitioned into the edge-disjoint paths connecting internal nodes with leaves as follows. Each internal node v (including the degree-2 root) is split into two nodes v_1 and v_2 such that v_1 becomes a leaf incident to one of the downstream edges and v_2 becomes a degree-2 node (or a leaf if v is the root) incident to an edge connecting v to its parent (if v is not the root) and another downstream edge. Since each node is incident to a downstream edge, each resulted connected component will be a path containing exactly one leaf of T connected to an internal node of T.

We break the binary tree OPT_{k+1} into edge-disjoint paths described above (see Figure 2(b)) and then consider each such path along with all nodes of

Input: Graph $G = (V, E, l)$ with two nonzero rates $r_1 < r_2$, source s, terminal sets S_1 of rate r_1 and S_2 of rate r_2, Steiner tree α_1-approximation algorithm A_1 and a β-convex α_2-approximation algorithm A_2
Output: Low cost QoSMT spanning all terminals

---

1. Compute an approximate Steiner tree ST1 for $s \bigcup S_1 \bigcup S_2$ using algorithm A_1
2. Compute an approximate Steiner tree T_2 for $s \bigcup S_2$ (treating all other points as Steiner points) using algorithm A_1. Next, contract T_2 into the source s and compute the approximate Steiner tree T_1 for s and remaining rate r_1 points using algorithm A_2. Let ST2 be $T_1 \bigcup T_2$
3. Output the minimum cost tree among ST1 and ST2

Fig. 3. QoSMT approximation algorithm for two non-zero rates

rate r_k that are attached to it. This results in a decomposition of OPT_k into edge-disjoint connected components OPT_k^i, where each component consists of a path $D_k^i = OPT_k^i \cap OPT_{k+1}$ and attached Steiner trees with edges of rate r_k (see Figure 2(c)). Furthermore, note that the total length of the paths D_k^i is $l(OPT_{k+1}) = t_{k+1} + t_{k+2} + \cdots + t_N$.

Now we decompose the tree T_k along these full components OPT_k^i and, by Corollary 1, we get:

$$\begin{aligned} l(T_k) &\leq \sum_i \left[\alpha(l(OPT_k^i) - D_k^i) + \beta D_k^i\right] \\ &= \alpha t_k + \beta(t_{k+1} + t_{k+2} + \cdots + t_N) \end{aligned}$$

The lemma follows by multiplying the last inequality by r_k.

4 QoSMT Approximation Algorithm for Two Non-Zero Rates

In this section we give a generic approximation algorithm for the QoSMT problem with two non-zero rates (see Figure 3) and analyze its approximation ratio.

Recall that an edge e has rate r_i if the largest node rate in the component of $T - \{e\}$ that does not contain the source is r_i. Let the optimal Steiner tree in G have cost $opt = r_1 t_1 + r_2 t_2$, with t_1 being the total length of the edges of rate r_1 and t_2 being the total length of the edges of rate r_2. The algorithm in Figure 3 uses as subroutines two Steiner tree algorithms: an algorithm A_1 with an approximation ratio of α_1, and a β-convex algorithm A_2 with an approximation ratio of α_2. It outputs the minimum cost Steiner tree between the tree $ST1$ obtained by running A_1 with a set of terminals containing the source and the nodes with both high and low non-zero rate, and the tree $ST2$ obtained by running A_1 with a set of terminals containing the source and all high rate nodes, contracting the resulting tree into the source, and running A_2 with a set of terminals containing the contracted source and the low rate nodes.

Theorem 2. *The algorithm in Figure 3 has an approximation ratio of*

$$\max\left\{\alpha_2, \quad \max_r \ \alpha_1 \frac{\alpha_1 - \alpha_2 r + \beta r}{\alpha_1 - \alpha_2 r + \beta r^2} opt\right\}$$

Proof. We can bound the cost of ST1 by $cost(ST1) \leq \alpha_1 r_2 (t_1 + t_2)$. To obtain a bound on the cost of ST2 note that $cost(T_2) \leq \alpha_1 r_2 t_2$, and that, by Lemma 2, $cost(T_1) \leq \alpha_2 r_1 t_1 + \beta r_1 t_2$.

Thus, the following two bounds for the costs of $ST1$ and $ST2$ follow:

$$\begin{aligned} cost(ST1) &\leq \alpha_1 r_2 t_1 + \alpha_1 r_2 t_2 \\ cost(ST2) &\leq \alpha_1 r_2 t_2 + \alpha_2 r_1 t_1 + \beta r_1 t_2 \end{aligned}$$

We distinguish the following two cases:
Case 1: Let $\beta r_1 \leq (\alpha_2 - \alpha_1) r_2$. Then

$$\begin{aligned} cost(ST2) &\leq \alpha_1 r_2 t_2 + \alpha_2 r_1 t_1 + \beta r_1 t_2 \\ &\leq \alpha_1 r_2 t_2 + \alpha_2 r_1 t_1 + (\alpha_2 - \alpha_1) r_2 t_2 \\ &\leq \alpha_2 (r_2 t_2 + r_1 t_1) \\ &= \alpha_2 opt \end{aligned}$$

Case 2: Let $\beta r_1 > (\alpha_2 - \alpha_1) r_2$. Then the following two values are positive

$$\begin{aligned} x_1 &= \frac{r_1}{\alpha_1 r_2} (\beta r_1 - (\alpha_2 - \alpha_1) r_2) \\ x_2 &= r_2 - r_1 \end{aligned}$$

We will bound the following linear combination

$$\begin{aligned} x_1 cost(ST1) + x_2 cost(ST2) &= \frac{r_1(\beta r_1 - (\alpha_2 - \alpha_1) r_2)}{\alpha_1 r_2} cost(ST1) + (r_2 - r_1) cost(ST2) \\ &\leq r_1 (\beta r_1 - (\alpha_2 - \alpha_1) r_2)(t_1 + t_2) \\ &\quad + (r_2 - r_1)(\alpha_1 r_2 t_2 + \alpha_2 r_1 t_1 + \beta r_1 t_2) \\ &= ((\beta - \alpha_2) r_1^2 + r_1 r_2 \alpha_1) t_1 + ((\beta - \alpha_2) r_1 r_2 + r_2^2 \alpha_1) t_2 \\ &= ((\beta - \alpha_2) r_1 + r_2 \alpha_1)(r_1 t_1 + r_2 t_2) \\ &\leq (\beta r_1 + \alpha_1 r_2 - \alpha_2 r_1) opt \end{aligned} \tag{2}$$

Let *Approx* be the cost of the tree produced by our approximation algorithm. The inequality (2) implies that

$$\begin{aligned} Approx &= \min\{cost(ST1), cost(ST2)\} \\ &= \frac{x_1 \min\{cost(ST1), cost(ST2)\} + x_2 \min\{cost(ST1), cost(ST2)\}}{x_1 + x_2} \end{aligned}$$

$$\begin{aligned}
&\leq \frac{x_1 cost(ST1) + x_2 cost(ST2)}{x_1 + x_2} \\
&\leq \frac{\beta r_1 + \alpha_1 r_2 - \alpha_2 r_1}{\frac{r_1}{\alpha_1 r_2}(\beta r_1 - (\alpha_2 - \alpha_1) r_2) + r_2 - r_1} opt \\
&\leq \alpha_1 \frac{\beta r_1 r_2 + \alpha_1 r_2^2 - \alpha_2 r_1 r_2}{\beta r_1^2 - (\alpha_2 - \alpha_1) r_2 r_1 + \alpha_1 r_2^2 - \alpha_1 r_1 r_2} opt \\
&\leq \alpha_1 \frac{\alpha_1 - \alpha_2 r + \beta r}{\alpha_1 - \alpha_2 r + \beta r^2} opt
\end{aligned}$$

where $r = \frac{r_1}{r_2}$.

Summarizing the two cases we obtain that *Approx* is at most the maximum of two values – $\alpha_2 opt$ and $\alpha_1 \frac{\alpha_1 - \alpha_2 r + \beta r}{\alpha_1 - \alpha_2 r + \beta r^2} opt$ – which proves the theorem.

We can use Theorem 2 to obtain numerical bounds on the approximation ratios of our solution. Using that $\alpha_1 = 1 + \ln 3/2 + \epsilon$ for the algorithm from [12], $\alpha_2 = 5/3 + \epsilon$ for the algorithm from [11], $\alpha_1 = \alpha_2 = 11/6$ for the algorithm from [3], and $\alpha_1 = \alpha_2 = 2$ for the MST heuristic, and $\beta \to 1$ for all of the above algorithms (except for the algorithm from [12]), we maximize the expression in Theorem 2 to obtain the following theorem.

Theorem 3. *If the algorithm from [12] is used as A_1 and the algorithm from [11] is used as A_2, then the approximation ratio of the QoSMT algorithm in Figure 3 is $1.960 + \epsilon$. If the algorithm from [11] is used in place of both A_1 and A_2, then the approximation ratio is $2.059 + \epsilon$. If the algorithm from [3] is used in place of both A_1 and A_2, then the ratio is 2.237. If the MST heuristic is used in place of both A_1 and A_2, then the ratio is 2.414.*

5 Approximation Algorithm for the QoSMT with Unbounded Number of Rates

In this section, we propose an algorithm for the case of a graph with arbitrarily many non-zero rates $r_1 < r_2 < \cdots < r_N$. Our algorithm (see Figure 4) is a modification of the algorithm in [5]. As in [5], node rates are rounded up to the closest power of some number a starting with a^y, where y is picked uniformly at random between 0 and 1. In other words, we round up node rates to numbers in the set $\{a^y, a^{y+1}, a^{y+2}, \ldots\}$. The only difference is that we *contract* each approximate Steiner tree, T_k, constructed over nodes of rounded rate a^{y+k}, instead of simply taking their union as in [5]. This allows contracted edges to be reused at zero cost by Steiner trees connecting lower rate nodes. The following analysis of this improvement shows that it decreases the approximation ratio from 4.211 to 3.802.

Let T_{opt} be the optimal QoS Multicast tree, and let t_i be the total length of the edges of T_{opt} with rates rounded to a^{y+i}. First, we prove the following technical lemma:

Input: Graph $G = (V, E, l)$, source s, sets S_i of terminals with rate r_i, positive number a, and α-approximation β-convex Steiner tree algorithm
Output: Low cost QoSMT spanning all terminals

1. Pick y uniformly at random between 0 and 1. Round up each rate to the closest power of some number a starting with a^y, i.e., round up to numbers in the set $\{a^y, a^{y+1}, a^{y+2}, \ldots\}$. Form new terminal sets S'_i which are unions of terminal sets with rates rounded to the same number r'_i
2. $T \leftarrow \emptyset$
3. For each non-zero rounded rate r'_i, in decreasing order, do:
   Find an α-approximate Steiner tree T_i spanning $s \bigcup S'_i$
   $T \leftarrow T \cup T_i$
   Contract T_i into source s
4. Output T

Fig. 4. Approximation algorithm for multirate QoSMT

Lemma 3. *Let S be the cost of T_{opt} after rounding node rates as in Figure 4, i.e., $S = \sum_{i=0}^{n} t_i a^{y+i}$. Then,*

$$S \leq \frac{a-1}{\ln(a)} cost(T_{opt})$$

Proof. Our proof follows the proof of Lemma 4 in [5]. First, note that an edge e used at rate r in T_{opt} will be used at the rate a^{y+m}, where m is the smallest integer i such that a^{y+i} is no less than r. Indeed, e is used at rate r in T_{opt} if and only if the maximum rate of a node connecting to the source via e is r, and every such node will be rounded to a^{y+m}. Next, let $r = a^{x+m}$. If $x \leq y$, then the rounded up cost is a^{y-x} times the original cost; otherwise, if $x > y$, is a^{y+1-x} times the original cost. Hence, the expected factor by which the cost of each edge increases is

$$\int_0^x a^{y+x-1} dy + \int_x^1 a^{y-x} dy = \frac{a-1}{\ln a}$$

By linearity of expectation, the expected cost after rounding of T_{opt} is

$$S \leq \frac{a-1}{\ln a} cost(T_{opt})$$

Theorem 4. *The algorithm given in Figure 4 has an approximation ratio of*

$$\min_a \left((\alpha - \beta) \frac{a-1}{\ln a} + \beta \frac{a}{\ln a} \right)$$

Proof. Let *Approx* be the cost of the tree returned by the algorithm in Figure 4, and $Approx_k$ be the cost of the tree T_k constructed by the algorithm when considering rate r_k. Then, by Lemma 2,

$$Approx_k \leq \alpha a^{y+k} t_k + \beta a^{y+k+1} t_{k+1} + \beta a^{y+k+2} t_{k+2} + \cdots + \beta a^{y+n} t_n$$

where n is the total number of rates after rounding. Thus, we obtain the following upper bound on the total cost of our approximate solution.

$$
\begin{array}{llllll}
Approx \leq \alpha t_1 a^y & + \beta t_2 a^y & + \beta t_3 a^y & + \cdots + \beta t_{n-1} a^y & + \beta t_n a^y \\
 & + \alpha t_2 a^{y+1} & + \beta t_3 a^{y+1} & + \cdots + \beta t_{n-1} a^{y+1} & + \beta t_n a^{y+1} \\
 & & \ddots & & \\
 & & & + \alpha t_{n-1} a^{y+n-1} & + \beta t_n a^{y+n-1} \\
 & & & & + \alpha t_n a^{y+n}
\end{array}
$$

$$
= (\alpha - \beta) S + \beta \cdot \left(
\begin{array}{lllll}
t_1 a^y & + t_2 a^y & + t_3 a^y & + \cdots + t_{n-1} a^y & + t_n a^y \\
 & + t_2 a^{y+1} & + t_3 a^{y+1} & + \cdots + t_{n-1} a^{y+1} & + t_n a^{y+1} \\
 & & & \ddots & \\
 & & & + t_{n-1} a^{y+n-1} & + t_n a^{y+n-1} \\
 & & & & + t_n a^{y+n}
\end{array}
\right)
$$

$$
\leq (\alpha - \beta) S + \beta \cdot \left(
\begin{array}{lllll}
\vdots & & & & \\
t_1 a^{y-n+1} & \vdots & & & \\
t_1 a^{y-n+2} & + t_2 a^{y-n+2} & \vdots & & \\
t_1 a^{y-n+3} & + t_2 a^{y-n+3} & + t_3 a^{y-n+3} & & \\
\vdots & \vdots & \vdots & \ddots \vdots & \\
t_1 a^{y-1} & + t_2 a^{y-1} & + t_3 a^{y-1} & + \cdots + t_{n-1} a^{y-1} & \vdots \\
t_1 a^y & + t_2 a^y & + t_3 a^y & + \cdots + t_{n-1} a^y & + t_n a^y \\
 & + t_2 a^{y+1} & + t_3 a^{y+1} & + \cdots + t_{n-1} a^{y+1} & + t_n a^{y+1} \\
 & & \ddots & & \\
 & & & + t_{n-1} a^{y+n-1} & + t_n a^{y+n-1} \\
 & & & & + t_n a^{y+n}
\end{array}
\right)
$$

$$
\leq (\alpha - \beta) S + \beta S \left(1 + \frac{1}{a} + \frac{1}{a^2} + \cdots \right)
$$

$$
\leq (\alpha - \beta) \frac{a-1}{\ln a} cost(T_{opt}) + \beta \frac{a}{\ln a} cost(T_{opt})
$$

where the last inequality follows from Lemma 3.

Numerically, we obtain approximation ratios of 3.802, 4.059, respectively 4.311, when the β-convex α-approximation Steiner tree algorithm used in Figure 4 is the algorithm in [11], [3], respectively the MST heuristic.

Remark. The algorithm in Figure 4 can be easily derandomized using the same techniques as in [5]

6 Conclusions and Open Problems

In this paper we have considered a generalization of the Steiner problem in which each node possesses a rate and the cost of an edge with length l in a Steiner tree T connecting the terminals is $l \cdot r_e$, where r_e is the maximum rate in the component of $T - \{e\}$ that does not contain the source. We have given improved approximation algorithms finding trees with a cost at most 1.960 (respectively 3.802) times the minimum cost for the case of two (respectively unbounded number of) non-zero rates. Our improvement is based on the analysis of the gain resulting from the reuse of higher rate edges in the connectivity of the lower rate edges. An interesting open question is to extend this analysis to the case of three non-zero rates. The best known approximation factor for this case, is $\alpha(5 + 4\sqrt{2})/7 < 2.359$ [2, 14].

References

1. A. Balakrishnan, T.L. Magnanti, P. Mirchandani, *Modeling and Heuristic Worst-Case Performance Analysis of the Two-Level Network Design Problem*, Management Science, **40**: 846-867, (1994)
2. A. Balakrishnan, T.L. Magnanti, P. Mirchandani, *Heuristics, LPs, and Trees on Trees: Network Design Analyses*, Operations Research, **44**: 478-496, (1996)
3. P. Berman and V. Ramaiyer, *Improved Approximations for the Steiner Tree Problem*, Journal of Algorithms 17, 381-408 (1994)
4. A. Borchers and D.Z. Du, *The k-Steiner Ratio in Graphs*, SIAM Journal on Computing, **26**:857-869, (1997)
5. M. Charikar, J. Naor, and B. Schieber, *Resource Optimization in QoS Multicast Routing of Real-Time Multimedia*, IEEE/ACM Transactions on Networking, **12**: 340-348 (2004)
6. C.J. Colbourn and G.L. Xue, *Grade of service Steiner trees in series-parallel networks*, Advances in Steiner Trees (Ding-Zhu Du, J.M. Smith, and J.H. Rubinstein, eds.), Kluwer Academic Publishers, 2000, pp.163–174.
7. J.R. Current, C.S. Revelle, and J.L.Cohon, *The Hierarchical Network Design Problem*, European Journal of Operations Research, **27**: 57-66, (1986)
8. N. Maxemchuk, *Video Distribution on Multicast Networks*, IEEE Journal on Selected Areas in Communications **15**:357-372 (1997)
9. P. Mirchandani, *The Multi-Tier Tree Problem*, INFORMS Journal on Computing, **8**: 202-218, (1996)
10. K. Mehlhorn, *A faster approximation algorithm for the Steiner problem in graphs*, Information Processing Letters **27**: 125-128, (1988)
11. H. Promel and A. Steger, *A New Approximation Algorithm for the Steiner Tree Problem with Performance Ratio* $\frac{5}{3}$, Journal of Algorithms, **36**: 89-101,(2000)
12. G. Robins and A. Zelikovsky, *Improved Steiner Tree Approximation in Graphs*, Proc. of ACM/SIAM Symposium on Discrete Algorithms (SODA 2000), 770-779.
13. H. Takahashi and A. Matsuyama, *An Approximate Solution for the Steiner Problem in Graphs*, Math. Japonica, **6:** 573-577, (1980)
14. G. Xue, G.-H. Lin, D.-Z. Du, *Grade of Service Steiner Minimum Trees in the Euclidean Plane*, Algorithmica, **31**: 479-500, (2001)

15. A. Zelikovsky, *An 11/6-approximation algorithm for the network Steiner problem*, Algorithmica **9**: 463-470, (1993)
16. A. Zelikovsky, *A faster approximation algorithm for the Steiner tree problem in graphs*, Information Processing Letters **46**: 79-83, (1993)