Minimum-cost Broadcast through Varying-size Neighborcast *

Amotz Bar-Noy[1] Prithwish Basu[2] Matthew P. Johnson[3]
Ram Ramanathan[2]

[1] *City University of New York Graduate Center and Brooklyn College*
[2] *Raytheon BBN Technologies*
[3] *Pennsylvania State University*

Abstract. In traditional multihop network broadcast problems, in which a message beginning at one node is efficiently relayed to all others, cost models typically used involve a charge for each *unicast* or each *broadcast*. These settings lead to a minimum spanning tree (MST) problem or the Connected Dominating Set (CDS) problem, respectively. Neglected, however, is the study of intermediate models in which a node can choose to transmit to an arbitrary subset of its neighbors, at a cost based on the number of recipients (due e.g. to acknowledgements or repeat transmissions). We focus in this paper on a transmission cost model of the form $1 + Ak^b$, where k is the number of recipients, $b \geq 0$, and $A \geq 0$, which subsumes MST, CDS, and other problems.

We give a systematic analysis of this problem as parameterized by b (relative to A), including positive and negative results. In particular, we show the problem is approximable with a factor varying from $2 + 2H_\Delta$ down to 2 as b varies from 0 to 1 (via a modified CDS algorithm), and thence with a factor varying from 2 to 1 (i.e., optimal) as b varies from 1 to $\log_2(\frac{1}{A} + 2)$, and optimal thereafter (both via spanning tree).

For arbitrary cost functions of the form $1 + Af(k)$, these algorithms provide a $2 + 2H_\Delta$-approximation whenever $f(k)$ is sublinear and a $(1+A)/A$-approximation whenever $f(k)$ is superlinear, respectively. We also show that the problem is optimally solvable for any b when the network is a clique or a tree.

1 Introduction

A key problem in multihop wireless networks and in networking more generally is that of *network-wide broadcast*, in which a message sourced at one node (the root) must be disseminated to all nodes efficiently. Network-wide broadcast is applicable to dissemination of routing control messages such as link-state updates and route requests, as well as global awareness data (e.g. situation reports). Central to the problem is the notion of cost incurred at each hop in the dissemination

* Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

process. Cost models have typically been based on charges for one of two sorts of transmissions: *unicast*, in which a node sends a message to one of its neighbors, or *broadcast*, in which a node sends a message to all its neighbors, for a fixed cost, regardless of the number of neighbors receiving. The first method means finding a spanning tree; the second means solving a Connected Dominating Set (CDS) problem. Since individual unicasts are prohibitively expensive for all but very sparse networks, the de facto approach for the network-wide broadcast problem today is to use variants of CDS.

Most work to date has focused on the reliable version of CDS (each broadcast transmission is assumed to perfectly reach all its recipients). In real-world networks, however, this is seldom guaranteed; when the network-wide broadcast is of control packets, the effects of unreliability can be particularly onerous. In this paper, we consider the network-wide broadcast problem with reliable multicasting at the link layer, which leads to a non-trivial cost model for each transmission. That is, unlike the CDS problem where the cost of each broadcast is the same, the cost in a reliable multicast model depends on the number of receivers due to the need for acknowledgements and retransmissions.

More generally, the sender at each hop must ensure that all intended recipients, namely the downstream nodes in the constructed broadcast tree, receive the packet. This typically involves sending the packet, waiting for feedback from the intended receivers, and resending to those who did not receive, iterating until obtaining confirmation that all receivers have the packet. This motivates a cost model with a constant term for the transmission itself plus a cost that is a function of the number of recipients k (i.e., $1 + f(k)$).

To find candidates for the function $f(k)$, we consider a range of protocols from the reliable multicast literature, e.g. [13, 9, 6, 14]. These various solutions incur varying costs in the overhead and delay at each hop, ranging from *sublinear* to *linear* to *superlinear* in the number of receivers, as we now outline.

In [13, 9], the feedback takes the form of a *busy tone*, which is a narrow-band signal transmitted in a channel orthogonal to the one used for packets. Since tones are impervious to collisions (if two nodes place a busy tone on a channel, a busy tone is received), the feedback cost is independent of number of receivers, and so depends only on the expected number of retransmissions. As shown in an appendix omitted due to lack of space, the expected number of repetitive transmissions to deliver reliably (without any feedback) to k receivers, via a channel with packet error probability p, is concave in k, and hence the cost function is *sublinear* in k.

Although busy tones reduce the cost of feedback, they require special hardware. A simpler approach is to use acknowledgment (ACK) packets from each receiver. Several techniques may be employed to prevent ACK collisions. In [6], the ACKs are sent sequentially from the receivers and are subject to loss and collisions (in presence of hidden terminals and in very dense scenarios). Under this approach data packets retransmission is caused not just by the loss of data packets but also by the loss of ACKs, which can occur repeatedly, yielding a cost *superlinear* in k. This is also likely when receivers contend for access us-

b value	$b = 0$	$0 < b < 1$	$b = 1$	$1 < b < \log_2 3$	$\log_2 3 \leq b$
solution	CDS	Algorithm 1	pruned CDS; ST	ST	ST
approx	$H_\Delta + 2$	$2^{1-b}+(2H_\Delta)^{1-b}+o(1)$	$\frac{2}{1+1/(H_\Delta+2)}$; 2	c(b)	1
approx LB	$(1-\epsilon)\ln n$	$\frac{(1-\epsilon)\ln n + n^b}{1+n^b}$	$\frac{2}{1+1/((1-\epsilon)\ln n)}$	NP-hard	1

Table 1. Special cases of the problem when $A = 1$, parameterized by cost exponent b.

ing ALOHA or CSMA/CA. On the other hand, if TDMA is employed giving "perfect" access to the ACKs, the cost is clearly linear in number of receivers. Roughly linear cost in k models the protocol in [14] as well. In that, a Request ACK (RAK) and ACK handshake is performed for each receiver that protects ACKs from colliding with hidden terminals and therefore avoids superlinearity while increasing (perhaps significantly) the linear coefficient.

These are but a few reliable MAC protocols. Since a network-wide broadcast application may be serviced by any of these or others, depending on the system, we are interested in a general cost framework that approximately captures a wide range of protocols. In this paper we focus on a stylized family of k-neighborcast cost functions of the form $1 + Ak^b$, where $A > 0$ and $0 \leq b < \infty$, which varies from sublinear to linear to superlinear, depending on the value of b, and subsumes the broadcast and unicast models mentioned above. We find that parameter A in isolation is of relatively little consequence, but the character of the problem depends dramatically on the value of b (relative to A). This leads to approximation guarantees that are parameterized by the value A but, more significantly, to different algorithms for different ranges of b values.

Contributions. Modeling the cost of transmitting to k neighbors as $1 + Ak^b$, we give a systematic analysis of the problem of minimizing the total cost of network-wide broadcast. We give positive and negative results for our problem in a variety of special cases parameterized by b, summarized (for the special case of $A = 1$) in Table 1. In particular, we show the problem is approximable with a factor smoothly varying from $2 + 2H_\Delta$[1] down to 2 as b varies from 0 to 1 (via a modified CDS algorithm; see Fig. 1), and thence with a factor smoothly varying from 2 to 1 (i.e., optimal) as b varies from 1 to $\log_2(\frac{1}{A} + 2)$, and optimal thereafter (both via any spanning tree; see Fig. 2).

For arbitrary cost functions of the form $1 + Af(k)$, these algorithms provide a $2 + 2H_\Delta$-approximation whenever $f(k)$ is sublinear and a $(1 + A)/A$-approximation whenever $f(k)$ is superlinear, respectively. We also show that the problem is optimally solvable for any b when the network is a clique or a tree.

Our algorithms assume for simplicity that there is a specified root v_0; the algorithms and guarantees extend straightforwardly to the setting in which any node can be chosen as root.

Related work. Dominating Set is a classical optimization problem equivalent to Set Cover approximable with factor $H_{\Delta+1}$ which is essentially the best possible

[1] Δ is the maximum degree, and H_n is the nth harmonic number $\sum_{i=1}^{n} 1/i$.

[15]. Connected Dominating Set was studied by [7], which gave an approximation algorithms with factors $2H_\Delta + 2$ and $H_\Delta + 2$ for the unweighted setting, as well as results for the weighted setting which were later improved by [8], using the techniques of [10]. Dominating Set has long been known to be APX-hard for bounded-degree and cubic graphs [12, 1], but hardness results for cubic CDS were given more recently [2, 11]. The network-wide broadcasting problem—an application of CDS—has received much attention in the networking community [17]. In [4, 3], a PTAS for the CDS problem is give when the input is restricted to unit disk graphs. Power considerations are taken into account in [18]. In [5], the minimum latency broadcast problem is studied. Distributed algorithms for connected dominating set are given in [16, 5].

Organization. The rest of the paper is organized as follows. In Section 2 we formally define the problem, present an IP formulation, discuss some special graph topologies that are optimally solvable, and in an omitted appendix we discuss a sublinear setting modeling the cost of acknowledgements. In Section 3, we analyze a number of special cases of the problem parameterized by the value b. Section 4 concludes the paper.

2 Preliminaries

Given is an undirected graph G on n nodes and with maximum degree Δ in which the presence of an edge (u, v) indicates the possibility of directional communication between u and v. A message originating at the root must be relayed to all other nodes. The goal is to minimize the total cost of the transmissions. A node can send the message targeted to a specific subset of neighbors, an action we sometimes call *neighborcasting*. While the message may be heard by other nodes in the vicinity, the chosen receivers are guaranteed to receive the message, at a cost which depends on their number. We do not address the problem of scheduling of broadcasts, ACKs, downstream rebroadcasts etc. in a collision-free manner. We concentrate purely on the total cost incurred during broadcast obeying the idealized cost model defined as follows. The cost of multicasting a message from a node to k neighbors (which we call a *k-cast*) is:

$$m(k) = 1 + Ak^b$$

for some constants $A > 0$ and $b \geq 0$. (We use $a = 1/A$ rather than A when convenient; when $A = 0$, the problem becomes equivalent to Connected Dominating Set.) The problem solution is specified by the neighborcast(s) performed by each node. We emphasize that a node can perform multiple neighborcasts, which can be preferable when $b > 1$. The optimal cost of a send node with $d \leq \Delta$ receivers is the total cost $M(d)$ of the best partition of d into p neighborcast sets (with p possibly 1) of sizes $k_1 + k_2 + \cdots + k_p = d$. In discussing a particular multicast solution, we refer to non-leaf nodes, i.e. nodes performing transmissions (of which there are some number s), as *send nodes* or *senders*; we refer to all nodes other than the root as *receivers* (of which there are $m = n - 1$). We use *multicast*, *neighborcast*, and *transmission* interchangeably.

Assuming $M(d)$ is precomputed for each $1 \leq d \leq \Delta$ (see below), the problem can be defined by integer programming formulation (1-7), which requires that all the transmissions of node v be "paid for" by a single y_{vi}, which is then done with the optimal cost $M_i = M(i)$ for one node transmitting (possibly using multiple transmissions) to i neighbors.

$$\min \sum_{v,i} M_i y_{vi} \tag{1}$$

$$\text{s.t.} \sum_{v} x_{vu} = 1 \qquad \forall u \neq v_0 \tag{2}$$

$$i \cdot y_{vi} \geq \sum_{u \in N(v)} x_{vu} - n \cdot (1 - y_{vi}) \qquad \forall v, i \tag{3}$$

$$\sum_{i} y_{vi} = 1 \qquad \forall v \tag{4}$$

$$z_{uv} \geq x_{uv} \qquad \forall u \neq v \tag{5}$$

$$z_{uv} + z_{vu} \leq 1 \qquad \forall u \neq v \tag{6}$$

$$z_{uw} \geq z_{uv} + z_{vw} - 1 \qquad \forall \text{ distinct } u, v, w \tag{7}$$

$$x_{vu}, y_{vi}, z_{uv} \in \{0, 1\}$$

Constraint set 2 ensures that every non-root node receives a transmission from some other node. Constraint set 3 ensures that if $y_{vi} = 1$ (in which case $1 - y_{vi} = 0$) then v transmits to at most i other nodes (where $N(v)$ is the set of v's neighbors); if $y_{vi} = 0$ then the constraint is satisfied trivially. Constraint set 4 ensures that every node $v$has recorded some number, possibly 0, of children. Finally, constraint sets 5,6,7 define a partial order on nodes corresponding to children receiving from parents, which prevents cycles in message-passing.

A straightforward way to compute the $M(d)$ values in quadratic total time is by dynamic programming as follows: $M(0) = 0$ and for any $1 \leq e \leq d$,

$$M(e) = \min_{1 \leq h \leq e} \{m(h) + M(e - h)\} \tag{8}$$

In fact, though, we can compute it more quickly.

Proposition 1. *Each $M(d)$ can be computed in constant time.*

Proof. When $b \leq 1$ a single transmission to all neighbors will be optimal, so consider $b > 1$. Ideally all nodes that transmit will have the same number k of receivers, yielding total cost

$$\frac{n-1}{k}(1 + Ak^b)$$

Allowing k to be fractional, this value is minimized when

$$\frac{d}{dk}\Big(\frac{n-1}{k}(1 + Ak^b)\Big) = 0$$

which occurs when $A(b-1)k^b = 1$ or

$$k = \big(A(b-1)\big)^{-1/b}$$

This value k in general will be fractional, but numbers of recipients must be integral. Since k^b is convex when $b > 1$, the optimal set of receiver set cardinalities will be $\lfloor k \rfloor$ and $\lceil k \rceil$, so that $x \cdot \lfloor k \rfloor + y \cdot \lceil k \rceil = m$ for some integers x, y. If

$$\frac{m(\lfloor k \rfloor)}{\lfloor k \rfloor} < \frac{m(\lceil k \rceil)}{\lceil k \rceil}$$

then the optimal solution will perform $\lfloor k \rfloor$-casts as aggressively as possible. Then $m = x \cdot \lfloor k \rfloor + y \cdot \lceil k \rceil = (x + y) \cdot \lfloor k \rfloor + y$ implies $x + y = (m \div \lfloor k \rfloor)$, and thus there will be $y = (m \,\%\, \lfloor k \rfloor)$ $\lceil k \rceil$-casts and $x = (m \div \lfloor k \rfloor - y)$ $\lfloor k \rfloor$-casts (where $\div$ indicates integer division and $\%$ remainder). Otherwise, performing $\lceil k \rceil$-casts as aggressively as possible will be optimal. In this case, $m = x \cdot \lfloor k \rfloor + y \cdot \lceil k \rceil = (x+y) \cdot \lceil k \rceil - x$ implies $x+y = \lceil m / \lceil k \rceil \rceil$, and thus there will be $y = (\lceil m / \lceil k \rceil \rceil - x)$ $\lceil k \rceil$-casts and $x = -(m \,\tilde{\%}\, \lceil k \rceil)$ $\lfloor k \rfloor$-casts (where *negative remainder* $r = Z \,\tilde{\%}\, d$ is the unique integer r satisfying $-d < r \leq 0$ and $Z = qd + r$ for some nonnegative integer q). □

From this and the fact that the transmission tree in a clique will be a star graph, we have the following:

Corollary 1. *The optimal multicast strategy can be computed in constant time in a clique and in linear time in a (rooted) tree.*

3 Problem settings parameterized by b

We now turn to general graphs, analyzing the problem for different values of b.

3.1 $b = 0$ and $b = 1$

When $b = 0$, any transmission costs $1 + Ak^0 = 1 + A$, and so the problem is equivalent to Connected Dominating Set and thus admits the following:

Proposition 2. *When $b = 0$ the problem is approximable with factor $H_\Delta + 2$ [7] but is not approximable with factor $(1 - \epsilon) \ln n$ for any $\epsilon > 0$ (unless $NP \subseteq DTIME(n^{O(\log \log n)})$) [15].*

When $b = 1$, the receiving costs sum to exactly $A(n-1)$, and so the objective is simply to minimize the number of senders (and hence maximize the number of leaves.

Proposition 3. *When $b = 1$ the problem is approximable with factor $\frac{A+1}{A+1/(H_\Delta+2))}$ but not approximable with factor $\frac{A+1}{A+1/((1-\epsilon)\ln n)}$ (unless $NP \subseteq DTIME(n^{O(\log \log n)})$).*

Proof. When $b = 1$, each transmission to k recipients has cost exactly $1 + Ak$. In a solution with exactly t transmitters that transmit exactly once to each node other than the root, the sum of the transmission costs will be $t + A(n-1)$. That is, the total cost of receiving is invariant (assuming that we take care never to transmit to a node that has already received), and so the objective again becomes minimizing the number of transmitters.

The cost of transmissions can be approximated within factor $f = H_\Delta + 2$ using the unweighted CDS algorithm of [7], which we then prune, i.e. remove redundant edges from the implied multicasts. For the overall approximation ratio we then have:

$$\begin{aligned}\frac{ALG_{CDS} + A \cdot (n-1)}{OPT_{CDS} + A \cdot (n-1)} &\leq \frac{f \cdot OPT_{CDS} + A \cdot (n-1)}{OPT_{CDS} + A \cdot (n-1)} \\ &\leq \frac{f \cdot (n-1)/f + A \cdot (n-1)}{(n-1)/f + A \cdot (n-1)} \\ &\leq \frac{1+A}{1/f + A}\end{aligned}$$

The second inequality follows because the preceding expression is maximized when OPT_{CDS} is as large as possible.

For hardness of approximation, an approximation of factor $\frac{A+1}{A+1/((1-\epsilon)\ln n)}$ would yield an approximation of factor $(1-\epsilon)\ln n$ for CDS. □

As n goes to infinity, the approximation guarantee of the subroutine also goes to infinity, and so becomes weaker and weaker, converging to 2. In fact, this factor can be obtained more easily.

Corollary 2. *Using* any *algorithm to compute a CDS or ST as a subroutine would yield an approximation with factor $(1 + A)/A$ (or 2 in the case of $A = 1$) when $b = 1$.*

Moreover, for similar reasons, we have:

Corollary 3. *Using* any *algorithm to compute a CDS or ST as a subroutine would yield an approximation with factor $(1 + A)/A$ (or 2 in the case of $A = 1$) when the cost function is of the form $1 + m(k)$ for superlinear $m(k)$.*

3.2 $b \geq \log_2 3$

We find a threshold for b for which unicasting performs within a factor $c \geq 1$ of any multicasting solution:

$$\begin{aligned}cm(k) &\geq km(1) \quad &\text{iff} \\ 1 + Ak^b &\geq k/c + kA/c \quad &\text{iff} \\ b &\geq \frac{\log\left(\frac{k/c-1}{A} + k/c\right)}{\log k} \\ &= \log_k\left((a+1)\,k/c - a\right) \end{aligned} \tag{9}$$

The meaning of the Ineq. 9 is that whenever it holds then for these choices of k, a, b unicasting necessarily performs within a factor c of multicasting. We first consider $c = 1$.

Lemma 1. *$f(k,a) = \log_k((a+1)k - a)$ is a decreasing function in $k, \forall a > 0$.*

Proof. Differentiating $f(k,a)$ with respect to k, we get:

$$f'(k,a) = \frac{(a+1)\ k \log k - ((a+1)k - a)\ \log((a+1)k - a)}{k\ ((a+1)k - a)\ \log^2 k} \tag{10}$$

We can prove the lemma by showing that $f'(k,a) < 0$, for $k > 1, a > 0$. Let us denote the numerator of Eq. 10 by:

$$h(k,a) = (a+1)\ k \log k - ((a+1)k - a)\ \log((a+1)k - a)$$

It is easy to verify that $h(1,a) = 0$. Differentiating $h(k,a)$ with respect to k, we get:

$$\begin{aligned} h'(k,a) = (a+1)k\frac{1}{k} &+ (a+1)\log k - ((a+1)k - a) \cdot \\ &\frac{a+1}{(a+1)k - a} - (a+1)\log((a+1)k - a) \\ &= (a+1)\log\frac{k}{(a+1)k - a} \\ &< 0 \quad (\text{since } (a+1)k - a > k \text{ for } a > 0, k > 1) \end{aligned} \tag{11}$$

From Ineq. 11 and the fact that $h(1,a) = 0$, we conclude that $h(k,a) < 0$ for $k > 1, a > 0$; since the denominator of Eq. 10 is positive, it immediately follows that $f'(k,a) < 0$ for $k > 1, a > 0$, and therefore $f(k,a)$ is a decreasing function in k for $a > 0$. □

Proposition 4. *Let $b \geq \log_2(a+2)$. Then unicasting always (i.e., by any spanning tree (ST)) performs at least as well as any multicasting solution.*

Proof. From Lemma 1, $f(k,a)$ is a decreasing function in k for $a = 1/A > 0$. Therefore its maximum for an integer $k > 1$ is obtained for $k = 2$. Hence, by Ineq. 9 the condition that

$$b \geq \log_2(a+2)$$

suffices for unicasting to performs at least as well as multicasting for any value of k. □

Remark 1. Of course k^b is superlinear for any $b > 1$. What this shows is that for $b > \log_2(a+2)$, k receivers should never be divided into multiple transmissions.

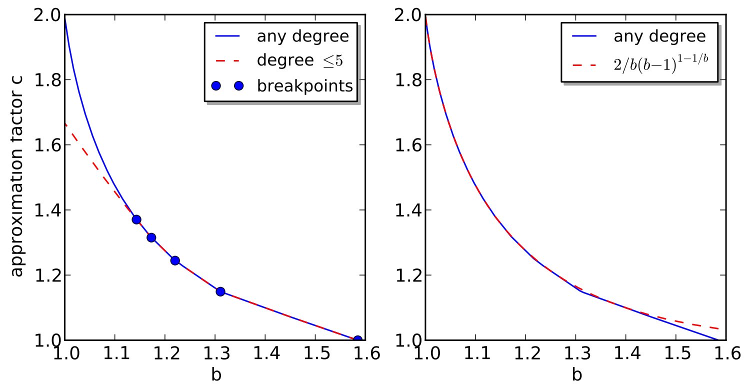

Fig. 1. Approximation factor provided by a spanning tree for each value b (with $A = 1$).

3.3 $1 \leq b \leq \log_2 3$

Now we consider the intermediate setting of $1 \leq b \leq \log_2(a+2)$. We know that at the extremes of 1 and $\log_2(a+2)$, the approximation ratios provided by any spanning tree (ST) are 2 and 1, respectively. Let $b_{a,c}(k) = \log_k((a+1)k/c - a)$, and let $b_a(c) = \max_{k=2}^{\infty} b_{a,c}(k)$, which by Ineq. 9 is the minimum value of b for which any spanning tree provides a c-approximation. We now invert this function to find the approximation factor c as a function of b.

Theorem 1. *Any spanning tree will provide approximation factor* $c_a(b) = \max_{k=2}^{\Delta} 2k/(a+k^b)$.

Proof. The value of $b_a(c)$ for each input c will be equal to $b_{a,c}(k)$ for some input k. Fix k, and consider the function (of c) $b_{a,k}(c) = b_{a,c}(k)$ and its inverted form $c_{a,k}(b) = 2k/(a+k^b)$. For each value k, unicasting to k receivers will have cost at most $c_{a,k}(b)$ times the cost of multicasting to k receivers. The value k can vary from 2 to Δ. Therefore the approximation factor based on b will be $c_a(b) = \max_{k=2}^{\Delta} c_{a,b}(k)$. That is, the function $c_a(b)$ will be composed piecewise of segments of the functions $c_{a,k}(b)$ (see Fig. 1), with *knees* (or breakpoints) at the values b for which $\frac{k}{k+1} = \frac{k^b+a}{(k+1)^b+a}$. □

We now give a simpler though more conservative approximation guarantee.

Proposition 5. *The approximation factor of spanning tree is upper-bounded as a function of b by the function $\hat{c}_a(b) = \frac{2(b-1)}{a+b-1}(\frac{a}{b-1})^{1/b}$ (or $\hat{c}_a(b) = \frac{2}{b}(b-1)^{1-1/b}$ when $a = 1$).*

Proof. Since the goal is to find an envelope of the function $c_a(b)$, we attempt to find the maximum of the function $c_{a,b}(k)$. Differentiating with respect to k and equating the result to 0, we get:

$$c'_{a,b}(k) = \frac{2a - 2(b-1)k^b}{(a+k^b)^2} = 0$$

Solving for the optimal k we get $k = (\frac{a}{b-1})^{1/b}$; and plugging this into $c_{a,b}(k)$ we have:

$$c_{a,b}(k) \leq c_{a,b}\left(\left(\frac{a}{b-1}\right)^{1/b}\right) = \frac{2(b-1)}{a+b-1}\left(\frac{a}{b-1}\right)^{1/b}$$

which simplifies to $\frac{2}{b}(b-1)^{1-1/b}$ when $a = 1$. □

3.4 $0 < b < 1$

We now give an algorithm for the $0 < b < 1$ setting which is an adaptation of the Algorithm I of [7]. The algorithm takes the root node v_0 as a parameter; if any root can be chosen, the algorithm can be run using every possible starting node.

Algorithm 1 $0 < b < 1$ Greedy (given root v_0)

```
color v0 gray and color all other nodes white
while there remain nonblack nodes do
    make a most cost-effective move, using a gray node v1 and possibly a nonblack
    neighbor v2 of v1
    color v1 (and v2 if used) black and color all white neighbors of v1 (and v2) gray
end while
```

The algorithm grows a multicast tree by repeatedly making one of two kinds of moves: 1) a one-node move, choosing a leaf of the current tree to transmit to all its nontree neighbors (i.e., those not yet in the tree); or 2) or a two-node move, choosing a leaf v_1 plus a nontree neighbor v_2 of v_1 and transmitting to all nontree neighbors of v_1 and v_2.

The *cost-effectiveness* of a move is the ratio of its total cost to the number of new nodes added to the tree. Because the cost of each transmission is sublinear in number of receivers, the best move will transmit to all the neighbors of the move's one or two transmitting nodes; the best two-node move will have one of the nodes transmit to all its nontree neighbors and then the other transmit to all its remaining nontree neighbors. Therefore at each step there are only a linear number of moves to consider.

We now prove the approximation guarantee, adapting the arguments of [7].

Theorem 2. *For $0 < b < 1$, Algorithm 1 provides an approximation of factor $2^{1-b} + (2H_{\Delta})^{1-b} + o(1)$ for any $A > 0$.*

Proof. Let OPT be the set of transmissions defining some minimum-cost multicast tree. Let S_i be the set of nodes that are children of some transmitting node v_i in OPT. Since the cost function is sublinear, all the nodes of S_i will receive from a single transmission of v_i. The sets S_i are disjoint. In each move, we add one or two senders to the multicast tree; the cost the move will be charged (divided equally) to the new nodes added to the tree. In the optimal solution, the total cost of transmitting to S_i is exactly $a + |S_i|^b$. We now bound the total cost

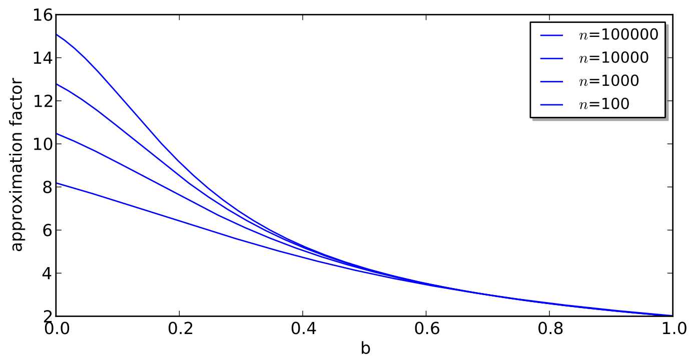

Fig. 2. Approximation factor provided (see the LHS of Ineq. 13) by Algorithm 1 for each value b and several values n (with $A = 1$).

charged to the members of S_i. Let u_j be the number of nodes in S_i remaining uncovered just after move j so that $u = u_0 = |S_i|$ and $u_k = 0$ after some move k. (We restrict our attention to those moves covering nodes in S_i.)

In the first move exactly $u - u_1$ nodes are marked, at a total cost of at most $\max\{a + (u - u_1)^b, 2a + 2((u - u_1)/2)^b\} \leq 2a + 2^{1-b}(u - u_1)^b \leq 2a + 2^{1-b}u^b$.

After the first move (after at least one node in S_i is added to the tree) it becomes possible to choose node v_i in a two-node move. Therefore in any subsequent move $j > 1$, we could cover all u_j remaining nodes of S_i with cost $2a + 2^{1-b}u_j^b$, and so the cost-effectiveness of move j is at worst $(2a + 2^{1-b}u_j^b)/u_j$. These costs sum to:

$$
\begin{aligned}
2a + 2^{1-b}u^b &+ \sum_{j=1}^{k-1} \frac{2a + 2^{1-b}u_j^b}{u_j}(u_j - u_{j+1}) \\
&= (2a + 2^{1-b}u^b) + 2a\sum_{j=1}^{k-1}\frac{u_j - u_{j+1}}{u_j} + 2^{1-b}\sum_{j=1}^{k-1}\frac{u_j^b(u_j - u_{j+1})}{u_j} \\
&\leq (2a + 2^{1-b}u^b) + 2a\sum_{j=1}^{k-1}\frac{u_j - u_{j+1}}{u_j} + 2^{1-b}\sum_{j=1}^{k-1}\frac{u_j - u_{j+1}}{u_j^{1-b}} \qquad (12) \\
&\leq (2a + 2^{1-b}u^b) + 2aH_u + 2^{1-b}H_u^{(1-b)}
\end{aligned}
$$

Here $H_n^{(x)}$ indicates the generalized harmonic number $\sum_{i=1}^{n} 1/i^x$. The last inequality follows from the observation that for monotonically decreasing integers u_j, $u_k = 0$, and $0 \leq b \leq 1$, we have $\sum_{j=1}^{k-1} \frac{u_j - u_{j+1}}{u_j^{1-b}} \leq H_u^{(1-b)}$. That is, the sum is maximized when $u_{j+1} - u_j = 1$ for each j. Otherwise, if there were some u_j such that $u_j - u_{j+1} = d > 1$, then the sum could be increased by replacing d/u_j with $\sum_{\ell=0}^{d-1} 1/(u_j - \ell)$.

When $b = 0$, an approximation guarantee of $2H_\Delta + 2$ obtains, since the algorithm's behavior collapses to that of [7]. Now let $b > 0$. The cost of sending to S_i in OPT is exactly $a + u^b$, which yields the following approximation factor:

$$\frac{(2a + 2^{1-b}u^b) + 2aH_u + 2^{1-b}H_u^{(1-b)}}{a + u^b} \leq 2^{1-b} + o(1) + \frac{2^{1-b}H_u^{(1-b)}}{u^b} \qquad (13)$$
$$\leq 2^{1-b} + o(1) + (2H_\Delta)^{1-b}$$

The last inequality follows by application of the "counting measure" special case of Hölder's inequality, substituting $a_i = 1$, $c_i = 1/i^{1-b}$, $p = 1/b$, $q = 1/(1-b)$ (note that $1/p + 1/q = 1$ as required):

$$\sum_{i=1}^{u} a_i c_i \leq \Big(\sum_{i=1}^{u} a_i^p\Big)^{1/p} \cdot \Big(\sum_{i=1}^{u} c_i^q\Big)^{1/q}$$
$$H_u^{(1-b)} = \sum_{i=1}^{u} 1/i^{1-b} \leq \Big(\sum_{i=1}^{u} 1\Big)^{b} \cdot \Big(\sum_{i=1}^{u} (1/i^{1-b})^{1/(1-b)}\Big)^{1-b} = u^b (H_u)^{1-b}$$

□

More generally, we have:

Corollary 4. *For $m(k) = a + f(k)$ where $f(k)$ is an arbitrary sublinear monotonic increasing cost function, Algorithm 1 provides a $2 + 2H_\Delta$ approximation.*

Proof. We upper-bound the cost of the first move by $2a + 2f(u)$ and the second sum in Ineq. 12 by $2f(u)H_u$ rather than $2^{1-b}H_u^{1-b}$. Substituting this into the LHS of Ineq. 13 yields:

$$\frac{(2a + 2f(u)) + 2aH_u + 2f(u)H_u}{a + f(u)} \leq 2 + 2H_\Delta$$

□

In an omitted appendix we observe that Corollary 4 applies to the repeated broadcast transmission model. In Appendix A we prove a two-part NP-hardness result for each *particular* value of $b \in [0, \log_2(3))$, combining the ranges $b <$ 1.395... and $b \geq$ 1.395..., the value of b at which $m(3) = m(1) + m(2)$ (when $A = 1$), i.e. $3^b = 2 + 2^b$. We also show a hardness of approximation result for $b \in (0, 1)$.

4 Conclusion

In this paper we presented positive and negative results for a multicast problem with cost function $1 + Ak^b$ and for $1 + m(k)$ with $m(k)$ sublinear or superlinear. As stated in the introduction, part of our motivation is that such cost functions can (approximately) model the cost of a transmission to k neighbors in various sorts of realistic systems. In ongoing experimental research, we are pursuing two directions:

- learning the parameters b and A that are (approximately) satisfied by certain network models and systems
- evaluating our algorithms performance based on not just the idealized cost model $1 + Ak^b$ but also on the *systems' actual transmission costs*

References

1. P. Alimonti and V. Kann. Some apx-completeness results for cubic graphs. *Theor. Comput. Sci.*, 237(1-2):123–134, 2000.
2. P. S. Bonsma. Max-leaves spanning tree is apx-hard for cubic graphs. *CoRR*, abs/0912.0226, 2009.
3. X. Cheng, X. Huang, D. Li, W. Wu, and D.-Z. Du. A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks. *NETWORKS*, 42(4):202–208, 2003.
4. B. Das and V. Bharghavan. Routing in ad hoc networks using minimum connected dominating sets. In *Proceedings of the International Conference on Communications (ICC)*, 1997.
5. R. Gandhi, A. Mishra, and S. Parthasarathy. Minimizing broadcast latency and redundancy in ad hoc networks. *IEEE/ACM Transactions on Networking*, 16(4):840–851, August 2008.
6. H. Gossain, N. Nandiraju, K. Anand, and D. P. Agrawal. Supporting mac layer multicast in ieee 802.11 based manets: Issues and solutions. In *LCN*, pages 172–179, 2004.
7. S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374–387, 1998.
8. S. Guha and S. Khuller. Improved methods for approximating node weighted steiner trees and connected dominating sets. *Inf. Comput.*, 150(1):57–74, 1999.
9. S. Gupta, V. Shankar, and S. Lalwani. Reliable multicast mac protocol for wireless lans. In *Communications, 2003. ICC '03. IEEE International Conference on*, volume 1, pages 93 – 97 vol.1, may 2003.
10. P. N. Klein and R. Ravi. A nearly best-possible approximation algorithm for node-weighted steiner trees. *J. Algorithms*, 19(1):104–115, 1995.
11. P. Lemke. The maximum-leaf spanning tree problem in cubic graphs is np-complete. Technical report, University of Minnesota, Mineapolis, 1988.
12. C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *J. Comput. Syst. Sci.*, 43(3):425–440, 1991.
13. W. Si and C. Li. Rmac: a reliable multicast mac protocol for wireless ad hoc networks. In *Parallel Processing, 2004. ICPP 2004. International Conference on*, pages 494 – 501 vol.1, aug. 2004.
14. M.-t. Sun, L. Huang, S. Wang, A. Arora, and T.-H. Lai. Reliable mac layer multicast in ieee 802.11 wireless networks. *Wireless Communications and Mobile Computing*, 3(4):439–453, 2003.
15. V. V. Vazirani. *Approximation algorithms.* Springer Verlag, Germany, 2001.
16. P.-J. Wan, K. M. Alzoubi, and O. Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. *MONET*, 9(2):141–149, 2004.
17. J. Wu and H. Li. A dominating-set-based routing scheme in ad hoc wireless networks. *Telecommunication Systems*, 18(1-3):13–36, 2001.
18. J. Wu, B. Wu, and I. Stojmenovic. Power-aware broadcasting and activity scheduling in ad hoc wireless networks using connected dominating sets. *Wireless Communications and Mobile Computing*, 3(4):425–438, 2003.

A Hardness results

When $b \approx 1.395$, we have $m(3) = m(1) + m(2)$ (when $A = 1$), because this b is the root of $3^b = 2 + 2^b$. This is the point at which a 3-transmission goes cheaper than a 1-transmission plus a 2-transmission to more expensive. We prove hardness for $0 \leq b \leq \log_2 3$ by considering values of b both less than and greater than 1.395.

Lemma 2. *For each particular* $b \in [0, 1.395...)$ *the problem (with* $A = 1$*) is NP-hard, even on cubic graphs.*

Proof. We reduce from Connected Dominating Set restricted to cubic graphs (i.e., every vertex of degree 3), which is known to be APX-hard [2].

Given the CDS instance I_{CDS}, the multicast problem instance I consists of the same graph, combined with the specified b value and $A = 1$. Now, consider an optimal solution OPT to I with s send nodes. Since the graph is cubic, there can be at most one node that performs a transmission to three receivers—the root. Since for $b < 1.395...$ we have these inequalities:

$$m(3) < m(1) + m(2) \tag{14}$$

$$m(2) < m(1) + m(1) \tag{15}$$

Then in any optimal solution the root transmits to three receivers (due to Ineq. 14) and no send node will perform two separate transmissions (due to Ineqs. 14 and 15). Therefore $s_1 + s_2 = s - 1$, where s_i is the number of send nodes transmitting to exactly i receivers.

Now suppose there were a solution to I_{CDS} with $s' < s$ (set $j = s - s'$) send nodes. We will show this implies the existence of a multicast solution of cost less than OPT. Let s'_i be the number of nodes transmitting to exactly i receivers in a minimum-cost assignment of receivers to the s' nodes in the I_{CDS} solution. Note that we may assume that also in *this* solution the root transmits to three receivers (with one transmission); if not, we can reverse the edges between it and any of its neighbors it does not send to, which again by Ineq. 14 will only lower the cost (and perhaps shrink s'). Thus we have $s'_1 + s'_2 = s' - 1 = s_1 - j + s_2$. Observe that we then have $s'_1 = s_1 - 2j$ and $s'_2 = s_2 + j$, and so the net change in the receiver cost is the removal of $2j$ nodes transmitting to 1 receiver each and the addition of j nodes transmitting to two nodes. The net effect of these changes on cost is $j \cdot (1 + 2^b - 2 \cdot (1 + 1^b))$, which is negative for any $b \leq 1$. Thus we obtain a contradiction, and so the s senders constitute an optimal CDS solution. □

Lemma 3. *For each particular* $b \in [1.395..., \log_2(3))$ *the problem (with* $A = 1$*) is NP-hard, even on cubic graphs.*

Proof. For b in the specified range, Ineq. 14 no longer holds, and so an optimal solution will never send to 3 receivers in a single transmission, but only to groups of 1 or 2. Since Ineq. 15 continues to hold, an optimal solution will try to perform as many 2-casts as possible. A node performing only 2-casts will have an even number of children and, unless it is the root, therefore have odd degree in the tree induced by the transmissions. In the case of a graph with n even, therefore, the ideal situation will be for every node but one (the root) to have an even number of children and hence for every node in the induced tree to have odd degree. Determining whether a cubic graph with even n admits a spanning tree with all node degrees odd, however, is known to be NP-hard [11].

Combining Lemmas 2 and 3 we obtain:

Theorem 3. *For each particular* $b \in [0, \log_2(3))$ *(with* $A = 1$*) the problem is NP-hard, even on cubic graphs.*

We now give a hardness of approximation result for $0 < b < 1$ (which, note, does not cover the same range as Lemma 2).

Theorem 4. *For each particular* $b \in (0, 1)$*, the problem not approximable with factor* $\frac{(1-\epsilon)\ln n + n^b}{1+n^b}$ *for any* $\epsilon > 0$ *(unless* $NP \subseteq DTIME(n^{O(\log\log n)})$*).*

Proof. Let OPT be the optimal solution value. Let $OPT' = OPT'_s + OPT'_r$ be the best possible solution value for a solution whose set of senders constitutes an optimal solution to the corresponding unweighted CDS problem. Note that $OPT \leq OPT'$. Let $ALG = ALG_s + ALG_r$ be the cost of a solution returned by some algorithm. (In the sublinear setting we can assume each sender transmits only once.) Let $m = n - 1$ be the number of receivers, and let f upper-bound the ratio ALG_s/OPT_s, which cannot be as good as $(1-\epsilon)\ln n$ for any $\epsilon > 0$ (absent the hardness assumption invoked the theorem statement). Then the most optimistic case for the approximation ratio of ALG compared to OPT' is for ALG to do $ALG_s - 1$ unicasts and for OPT' to do OPT'_s equal-sized transmissions, i.e.:

$$\begin{aligned}
\frac{ALG_s + ALG_r}{OPT'_s + OPT'_r} &= \frac{fOPT'_s + fOPT'_s + (m - fOPT'_s)^b}{OPT'_s + OPT'_s(n/OPT_s)^b} \\
&\geq \frac{fOPT'_s + m^b}{OPT'_s + (OPT'_s)^{1-b}m^b} \\
&\geq \frac{(1-\epsilon)\ln n \cdot OPT'_s + m^b}{OPT'_s + (OPT'_s)^{1-b}m^b} \\
&\geq \frac{(1-\epsilon)\ln n \cdot OPT'_s + n^b}{OPT'_s(1 + n^b)} \\
&\geq \frac{(1-\epsilon)\ln n + n^b}{1 + n^b} \quad \text{(for } n \text{ large enough)}
\end{aligned}$$

Therefore since ALG cannot approximate OPT' within the stated factor, neither can it do so for the only smaller OPT. □