

MINIMUM POWER MULTICASTING WITH DELAY BOUND CONSTRAINTS IN AD HOC WIRELESS NETWORKS

O. E. KUNDAKCIOGLU, K. BULBUL, T. ÜNLÜYURT, K. KILIC AND O.
ERCETIN

Sabanci University,

Orhanlı, Tuzla, 34956, Istanbul, Turkey

Email: {erhunk@su., bulbul@, tonguc@, kkilic@, oercetin@}sabanciuniv.edu

In this paper, we design a new heuristic for an important extension of the minimum power multicasting problem in ad hoc wireless networks^{20,21}. Assuming that each transmission takes a fixed amount of time, we impose constraints on the number of hops allowed to reach the destination nodes in the multicasting application. This setting would be applicable in time critical or real time applications, and the relative importance of the nodes may be indicated by these *delay bounds*. We design a filtered beam search procedure for solving this problem. The performance of our algorithm is demonstrated on numerous test cases by benchmarking it against an optimal algorithm in small problem instances, and against a modified version of the well-known Broadcast Incremental Power (BIP) algorithm²⁰ for relatively large problems.

1. Introduction

In this study, we consider the minimum power multicasting (MPM) problem with delay bounds in ad hoc wireless networks. In this setting, a designated source node sends the same information to a subset of the nodes of the network (destination nodes) by possibly using the rest of the nodes as relay nodes. In addition, for each destination node, we impose a delay bound indicating the maximum number of nodes that the information goes through before reaching that node.

Many new multimedia applications, e.g., video streaming and multimedia conferencing, involve dynamic multiple participants, have stringent end-to-end delay requirements and consume large amount of network resources.

We select a set of nodes and their transmission powers (coverage). Each node transmits at the same transmission rate. A node that receives a packet

waits until the completion of the transmission, checks if the packet can be decoded correctly, and then starts transmitting to its children if selected. Thus, a slotted system architecture can be assumed with a slot length equal to the transmission delay. The propagation delay is neglected. There are no MAC layer collisions, since the same packet is transmitted throughout the network. By using a short and known pilot tone before the transmission of a packet, each receiving node can determine the channel parameters between the transmitter and itself. This information is then used to remove the negative effects of simultaneous transmissions of the same packet.

Unlike in wired networks, wireless multicast sessions may reach several nodes with a single transmission. Assuming omnidirectional antennas, all nodes that are within the maximum range of a node receive the same information when the node transmits. The maximum range depends on the power level of the transmitting node and the channel dependent exponent.

Given a fixed network topology, a designated source node and a set of destination nodes along with their delay bounds, our goal is to construct a multicast tree where the sum of the power levels for all nodes is minimum. This problem is \mathcal{NP} -complete even without the delay bounds, e.g., see ^{4,3}. In this paper, we develop a heuristic algorithm based on filtered beam search and report computational results. We benchmark our proposed algorithm against the optimal solution for small problems and against a modified version of the Broadcast Incremental Power (BIP) algorithm proposed in ²⁰ for the broadcasting problem without delay bounds.

Numerous papers have been published on the broadcasting/multicasting problem. Since all different versions of the problem are \mathcal{NP} -complete ^{4,3,22}, in almost all these papers heuristic algorithms are proposed. These papers can be classified along several dimensions. First, different solution approaches for broadcasting/multicasting are required in link-based wired and node-based wireless networks ²⁰. Second, some papers consider the problem with delay bounds ¹⁴ while others do not incorporate any such constraints ¹⁵. Among those that consider delay bounds, some use the maximum number of relay nodes ⁶, and others use a function of the distance between the destination node and the source node in the tree ¹⁴ as the maximum possible delay. Third, different objective functions are considered, such as minimizing the total power allocation ¹⁵, maximizing the life time of the system ¹⁹ or minimizing the maximum power assigned to the nodes of the network ⁶.

In the literature, different variations of the wired case have been considered with and without delay bound constraints. Typically, for such prob-

lems, shortest path, minimum spanning tree, Steiner tree based algorithms are proposed^{14,15,1}. Some of these algorithms can be implemented in a distributed manner^{2,7,8}. A genetic algorithm is proposed in¹⁸ whereas a branch and cut algorithm is developed in¹⁶. A polynomial time approximation scheme is presented in²² for a restricted version of the problem.

The line of research that considers the wireless case is limited. Alternate integer programming formulations are provided in⁵ for finding the optimal broadcast/multicast tree. In¹³, the problem of minimizing the maximum power is considered, and some special cases are solved optimally in polynomial time. In¹⁰, mixed integer programming models are presented as well as valid inequalities to solve the problem exactly whereas in^{9,17}, various heuristics are proposed. To the best of our knowledge, there is only one published paper for wireless ad hoc networks with delay bounds²³ which provides heuristic algorithms for a special case. A recent general review of optimization issues in broadcasting and multicasting problems is provided in¹¹.

We have several contributions in this paper. First, we extend the multicasting problem in ad hoc wireless networks to include delay bound constraints. Second, in Sect. 2 we provide substantial improvements to an integer programming formulation that was given in⁵. Third, we design a scalable heuristic to solve our problem for practical network sizes. In particular, we apply a general heuristic framework to this multicasting problem that was developed in the context of machine scheduling problems¹².

2. Problem Formulation

2.1. Terminology

We assume a fixed network topology with N nodes where V and D represent the set of all nodes and the set of destination nodes in the multicast application, respectively. Without loss of generality, node 1 is the source node, and it sends a message to all nodes in the set $D \subseteq V$. Each node may receive the multicast message either directly from the source or over a *relay* node that is retransmitting the message. Any node may transmit the message at most once, and such nodes are called *hop* nodes, including the source node. Note that several hop nodes may transmit simultaneously. *Leaf* nodes receive the message, but do not retransmit it. The number of hops from node 1 to node i is defined as the delay of node i . Our objective is to construct a multicast tree with minimum total power consumption such that no delay bound constraint is violated.

The power required to transmit a message from node i to node j in the network is proportional to d_{ij}^α where d_{ij} is the Euclidean distance between nodes i, j , and α is a constant that depends on the channel medium. In general, α is assumed to be between 2 and 4. Note that the locations of the nodes are fixed and known, and we let P be an $N \times N$ matrix in which P_{ij} denotes the minimum transmission power required to reach node j from node i .

A single transmission may reach several nodes in a wireless network, and in order to construct a multicast tree with minimum power consumption, we must keep track of all nodes that are covered by a single transmission $m \rightarrow n$ from node m to node n . Note that the multicast tree is not necessarily a spanning tree because some non-destination nodes may never be reached. Following the terminology in ⁵, we define an $N \times N$ reward matrix R in which the entry R_{mn} is a binary N -vector that specifies all nodes reached by the transmission $m \rightarrow n$. If $R_{mn}(p)$, the p^{th} element of R_{mn} , is equal to 1, then we have $d_{mp} \leq d_{mn}$, and the message sent from m to n does also reach node p . Otherwise, we have $R_{mn}(p) = 0$. Note that the reward matrix may not be symmetric, i.e., the vector R_{mn} is not necessarily equal to the vector R_{nm} .

Let X_{ijk} be a binary variable that assumes the value 1 if the k^{th} transmission in the multicast tree is $i \rightarrow j$, and zero otherwise. These binary variables are necessary to ensure the precedence relationships between the wireless links in the multicast tree, and to incorporate the delay bound constraints. The entry l_i in the delay bound vector $L = [l_1, \dots, l_N]^T$ denotes the maximum number of hops allowed in order to reach node i from node 1. For $i \notin D$, $l_i = \infty$. Since each node may transmit at most once, a node $i \in D$ must receive the multicast message in at most $N - 1$ hops, and we define $k^{\text{max}} = \min(N - 1, \max_i(l_i))$ as the maximum number of hops allowed to reach all nodes in set D .

2.2. IP Formulation

One of our main contributions in this paper is improving an integer programming (IP) formulation that was proposed in ⁵ for solving the minimum power broadcasting/multicasting problem in the absence of delay bound constraints. The IP formulation below is adapted from Formulation A in ⁵ by removing the redundant variables and constraints in that formulation as explained later in this section, and by adding the delay bound constraints (6). This formulation is used for benchmarking our heuristics against the

optimal solution in small problem instances in Sect. 5.

$$\min \sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N \sum_{k=1}^{k^{max}} P_{ij} X_{ijk} \quad (1)$$

$$\sum_{j=2}^N X_{1j1} = 1 \quad (2)$$

$$\sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N X_{ij1} \leq 1 \quad (3)$$

$$\sum_{\substack{j=1 \\ j \neq i}}^N X_{ijk} - \sum_{p=1}^{k-1} \sum_{m=1}^N \sum_{\substack{n=1 \\ n \neq m}}^N R_{mn}(i) X_{mnp} \leq 0 \quad \forall i \neq 1, 2 \leq k \leq k^{max} \quad (4)$$

$$\sum_{p=1}^{k-1} \sum_{m=1}^N \sum_{\substack{n=1 \\ n \neq m}}^N R_{mn}(i) X_{mnp} + \sum_{m=1}^N \sum_{\substack{n=1 \\ n \neq m}}^N X_{mnk} \geq 1 \quad \forall i \in D, 2 \leq k \leq k^{max} \quad (5)$$

$$\sum_{p=1}^{l_i} \sum_{m=1}^N \sum_{\substack{n=1 \\ n \neq m}}^N R_{mn}(i) X_{mnp} \geq 1 \quad \forall i \in D \quad (6)$$

$$X_{ijk} \in \{0, 1\} \quad \forall i, j, i \neq j, \forall k \quad (7)$$

The objective function (1) minimizes the total power consumption. Only the source node transmits in the first hop as ensured by the constraints (2)-(3), and a node $i \geq 2$ cannot transmit in hop $k \geq 2$ unless it has been reached by the first $k - 1$ transmissions by the constraints (4). Constraints (5) ensure that there is *at least* one transmission in hop $k \geq 2$ if there is at least one node $i \in D$ that has not been reached by the first $k - 1$ transmissions. A node $i \in D$ must be reached after l_i transmissions as enforced by the delay bound constraints (6). Note that a node $i \notin D$ may act as a relay node in this model.

In our preliminary computational experiments, we observed that our formulation is able to solve significantly larger models than Formulation A in ⁵ although we do not report detailed results here.

Theoretically, both our IP formulation and Formulation A in ⁵ include $O(N^2)$ constraints^a and $O(N^3)$ variables assuming k^{max} is $O(N)$. Therefore, even our formulation does not scale well when the number of nodes in the network increases, and heuristics are required for networks of practical size. We propose an effective heuristic for the minimum power multicasting problem with delay bounds in the next section.

^aIn ⁵, the authors make an incorrect assertion that their formulation has $O(N^3)$ constraints.

3. Filtered Beam Search

Filtered Beam Search (FBS) first proposed in ¹² for machine scheduling problems is essentially an *approximate* Branch and Bound (B&B) method. It partially inherits the power of B&B to search the feasible space systematically while also limiting the computational time. An example of a search tree for our multicasting problem with $N = 7$, *filter width* $f = 4$ and *beam width* $b = 2$ is presented in Fig. 1. The root node represents the partial multicast tree with only the source node before any transmission, and the nodes at level 1 represent the possible multicast trees after the source transmits its information. In general, we find the multicast trees after k transmissions at level k of the search tree. Some of these may be infeasible, and the rest are either complete or partial. The key to FBS is the two parameters *filter width* f and the *beam width* b . At each level of the tree, we follow a two-stage process to determine which partial multicast trees are likely to provide good solutions when they are completed. First, all nodes corresponding to partial multicast trees are evaluated for their quality by a local evaluation function (LEF) which is called *filtering*. The best f nodes are considered for a thorough investigation by a global evaluation function (GEF). Among these f nodes the best b are retained in the search tree for *beam* construction and the rest of the partial multicast trees are discarded permanently. Typically, GEF requires that we construct a complete solution starting with the partial solution represented by the current node. Thus, it provides a “global view” of the node. Unfortunately, such a thorough analysis for all nodes at each level may be computationally prohibitive for large problems. This is why we need the filtering process based on a “local view”. Later in this section, we discuss the specific local and global evaluation functions employed in this study. In Fig. 1, nodes 2 and 7 are pruned by LEF, and nodes 3 and 4 are pruned by GEF. Nodes 5 and 6 are retained for beam construction. At a node at level k of the

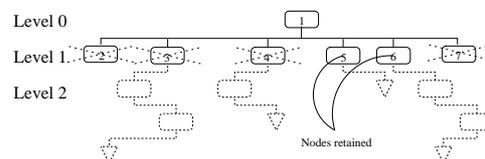


Figure 1. A filtered beam search tree with $N = 7$, $f = 4$ and $b = 2$.

search tree, we keep track of the following information: the current set of

transmissions $T^k = \{i_0 \rightarrow i_1, i_1 \rightarrow i_2, \dots, i_{k-1} \rightarrow i_k\}$, the set of covered nodes C^k that have already received the information, the set of uncovered nodes $NC^k = V \setminus C^k$, the set of nodes NA^k that have already transmitted once and are not allowed for another transmission, and the delay vector $Q^k = [q_1^k, q_2^k, \dots, q_N^k]$ given the current set of transmissions T^k . For a node $i \in C^k$, q_i^k is the actual delay and for a node $i \in NC^k$, q_i^k is a lower bound on the delay of the node as we discuss later. We initialize $T^0 = \emptyset$, $C^0 = \{1\}$, $NC^0 = V \setminus \{1\}$, $NA^0 = \emptyset$ and $Q^0 = [0, -1, \dots, -1]$.

In order to pass from level k to $k+1$ of the search tree, we consider all transmissions $i_k \rightarrow i_{k+1}$ for all partial multicast trees in the beam at level k so that $i_k \in C^k \setminus NA^k$ and $i_{k+1} \in NC^k$. Given a transmission $i_k \rightarrow i_{k+1}$, the delays are updated as:

$$q_p^{k+1} = q_p^k \quad \forall p \in C^k \quad (8)$$

$$q_p^{k+1} = q_{i_k}^k + 1 \quad \forall p \in NC^k \text{ and } R_{i_k i_{k+1}}(p) = 1 \quad (9)$$

$$q_p^{k+1} = \min_{i \in C^{k+1} \setminus NA^{k+1}} q_{i_k}^{k+1} + 1 \quad \forall p \in NC^{k+1}. \quad (10)$$

The delays of the nodes that are covered by the first k transmissions are not affected by the transmission $i_k \rightarrow i_{k+1}$ as given in (8). The delays of the nodes that are not covered by the first k transmissions and are covered by $i_k \rightarrow i_{k+1}$ are updated in (9). For the rest of nodes that are still not covered after the transmission $i_k \rightarrow i_{k+1}$ we can only specify lower bounds on their potential delays. For any such node, the delay is at least one more than the minimum delay of a node that is covered and has not transmitted yet as given in (10). After the updates (8)-(10), if $q_p^{k+1} > l_p$ for some node p , then the transmission $i_k \rightarrow i_{k+1}$ leads to an infeasible multicast tree, and the corresponding node at level k of the search tree is fathomed. Otherwise, given a partial multicast tree at level k and a feasible transmission $i_k \rightarrow i_{k+1}$, $T^{k+1} = T^k \cup \{i_k \rightarrow i_{k+1}\}$, $C^{k+1} = C^k \cup \{p \mid R_{i_k i_{k+1}}(p) = 1\}$, $NC^{k+1} = V \setminus C^{k+1}$ and $NA^{k+1} = NA \cup \{i_k\}$ is obtained at level $k+1$ of the search tree.

If a feasible transmission $i_k \rightarrow i_{k+1}$ added to a partial multicast tree at level k results in a complete multicast tree such that $D \subseteq C^{k+1}$, then this multicast tree is saved as one of the solutions of FBS. The FBS algorithm terminates when there are no more partial multicast trees to consider in the search tree, and the complete multicast tree with the minimum total power requirement is the solution found by FBS.

In this paper, we implement a slight variation of FBS, and we apply the filtering procedure separately to each of the b nodes in the beam at level k of the search tree. In other words, given a partial multicast tree at level k , all possible feasible partial multicast trees at level $k+1$ are

evaluated according to LEF, and the best f are retained to be evaluated by GEF. Therefore, when we move from level k to $k + 1$ of the search tree we initially have at most $b \times f$ partial multicast trees at level $k + 1$ to be evaluated by GEF. After the evaluation by GEF, only b of the partial multicast trees at level $k + 1$ are kept for beam construction, and the rest are discarded permanently. We also introduce a *child width* c for ensuring the diversification of the search and restrict the number of children of any node in the beam to at most c .

Suppose that we consider adding a feasible transmission $i \rightarrow j$ to a partial multicast tree T^k at level k , and we define a priority function Γ_{ij}^k for this transmission as specified in (11). This priority function, for which higher values are preferred, is our local evaluation function. In addition, in order to evaluate $i \rightarrow j$ globally, starting with $i \rightarrow j$ we keep adding feasible transmissions with the highest Γ_{ij}^k values to T^k until we obtain a complete multicast tree. The total power requirement of this complete multicast tree is the global evaluation function for the transmission $i \rightarrow j$. The function Γ_{ij}^k is simple and fast to compute which is important because up to $f \times b$ partial multicast trees must be evaluated globally at each level of the FBS tree.

$$\Gamma_{ij}^k = \pi^{\alpha/2} \left(\frac{|\{p \mid R_{ij}(p) = 1\} \setminus C^k|}{\pi d_{ij}^2} \right)^{\alpha/2} = \frac{|\{p \mid R_{ij}(p) = 1\} \setminus C^k|^{\alpha/2}}{P_{ij}} \quad (11)$$

The term in parentheses is the density of the previously uncovered nodes that are covered by the transmission $i \rightarrow j$. The priority function does also account for previously uncovered nodes that are not in the destination set because these may act as relay nodes. Finally, observe that the power consumption for reception and processing for nodes within the range of $i \rightarrow j$ can easily be incorporated into (11) by adding these quantities into the denominator of Γ_{ij}^k .

4. Modified BIP Algorithm with Delay Bound Constraints

The Broadcast Incremental Power (BIP) algorithm was introduced in ²⁰, and its improved performance against other proposed heuristics was demonstrated computationally. Also, a Multicast Incremental Power (MIP) algorithm was developed in ²⁰ by removing the unnecessary transmissions in a complete broadcast tree obtained by BIP. The BIP algorithm is a greedy construction algorithm. At each iteration, it considers all possible transmissions from the set of covered nodes to the set of uncovered nodes. The transmission with the minimum incremental cost is added to the partial broadcast tree until all nodes are covered. Note that if a node i is already transmitting to a node j , then its incremental power consumption to cover

an additional node k is $P'_i = P_{ik} - P_{ij}$. We modify the BIP algorithm to incorporate delay bound constraints and compare its performance to that of the filtered beam search in Sect. 5.

In the modified BIP (ModBIP) algorithm for our problem, the initializations of the sets T , C , NC and the delay vector Q are identical to those in FBS, i.e., $T = \emptyset$, $C = \{1\}$, $NC = V \setminus \{1\}$ and $Q = [q_1, q_2, \dots, q_N] = [0, -1, \dots, -1]$. At each iteration, we consider all transmissions $i \rightarrow j$ so that $i \in C$, $j \in NC$ and $q_i + 1 \leq l_j$, and the transmission $i' \rightarrow j'$ with the minimum incremental cost is added to the multicast tree. We obtain $T = T \cup \{i' \rightarrow j'\}$, $C = C \cup \{p \mid R_{i'j'}(p) = 1 \text{ and } q_{i'} + 1 \leq l_p\}$, $NC = V \setminus C$. The delays are updated as:

$$q_p = \min(q_p, q_{i'} + 1) \quad \text{if } R_{i'j'}(p) = 1 \text{ and } p \in C \quad (12)$$

$$q_p = q_{i'} + 1 \quad \text{if } R_{i'j'}(p) = 1, p \in NC \text{ and } q_{i'} + 1 \leq l_p. \quad (13)$$

We continue to iterate until all destination nodes are covered. In the final complete multicast tree some transmissions may be redundant. Therefore, a *sweep* operation²⁰ is applied to the final multicast tree to check whether the transmission power of any node can be decreased without violating the feasibility of the multicast tree. In particular, we sort the nodes in non-increasing order of their transmission powers and apply a sweep operation to each node starting at the head of the list. The sweep operation is applied to the final solutions of both ModBIP and FBS.

5. Computational Results

The algorithms for FBS and ModBIP are implemented in C++, and the IP formulation in Sect. 2.2 is solved by ILOG OPL Studio 3.7 and CPLEX 9.0. All computations are performed on an Intel Xeon 2.00 GHz PC with 1 GB of memory.

In all experiments, the nodes are distributed randomly in a 10×10 grid, and the channel loss component α is equal to 4. In order to test to impact of the tightness of the delay bounds, we generate uniformly distributed delays $U(1, \lceil \log_2 N \rceil)$ and $U(1, N - 1)$. Each node is selected as a destination node with probability $p = 0.5$, 0.75 or 1 where $p = 1$ corresponds to broadcasting instances. For each combination of parameters, 50 problem instances are generated randomly for $N = 10, 20, 30, 40, 50$. For small problem instances with $N = 10$ and $N = 20$, we compare the performances of FBS and ModBIP against the optimal solution. For larger problems, the heuristics are benchmarked against each other. After preliminary experiments, the filter, beam and child widths in FBS are set to $f = 0.3N$, $b = 0.2N$ and $c = 0.1N$, respectively.

In each cell of Table 1, the number in the first row is the number of optimal solutions obtained, and the numbers in parentheses are the average and maximum optimality gaps, respectively. We observe that as the number of nodes increases, the performance of FBS improves relative to that of ModBIP in terms of both the number of optimal solutions obtained and average optimality gaps.

N	$p = 0.5$				N	$p = 0.75$			
	$l_i \sim U(1, N-1)$		$l_i \sim U(1, \lceil \log_2 N \rceil)$			$l_i \sim U(1, N-1)$		$l_i \sim U(1, \lceil \log_2 N \rceil)$	
	ModBIP	FBS	ModBIP	FBS	ModBIP	FBS	ModBIP	FBS	
10	29 (0.28, 3.74)	30 (0.14, 1.67)	24 (0.45, 3.11)	36 (0.13, 1.41)	23 (0.33, 3.01)	30 (0.09, 1.52)	20 (0.44, 3.18)	41 (0.05, 0.89)	
20	10 (1.02, 8.61)	18 (0.34, 3.84)	7 (0.67, 4.97)	38 (0.09, 1.19)	11 (0.80, 12.48)	27 (0.08, 0.76)	14 (0.31, 3.05)	34 (0.05, 0.77)	

N	$p = 1$			
	$l_i \sim U(1, N-1)$		$l_i \sim U(1, \lceil \log_2 N \rceil)$	
	ModBIP	FBS	ModBIP	FBS
10	20 (0.17, 1.35)	37 (0.06, 1.19)	21 (0.32, 2.67)	45 (0.04, 0.72)
20	6 (0.64, 5.74)	26 (0.10, 1.38)	15 (0.10, 0.90)	39 (0.02, 0.63)

In each cell of Table 2, the number in the first row is the number of times FBS(ModBIP) performs better than ModBIP(FBS), and the numbers in parentheses are the average and maximum gaps with respect to the best solution found by either heuristic, respectively. The results indicate that FBS obtains significantly better solutions as the delay bounds get tighter and the problem size increases. In addition, in both Tables 1 and 2 the performance of ModBIP improves as p increases.

Tighter problems are solved by an order of magnitude faster by FBS and ModBIP as the feasible space is reduced significantly for such problems. For problem instances with $N = 50$, FBS requires less than 5 seconds on average. However, ModBIP requires an order of magnitude less computation time than FBS in general.

6. Conclusions and Future Work

In this paper, we consider the minimum power multicasting problem in ad hoc wireless networks with delay bound constraints. A combinatorial search algorithm FBS is developed and benchmarked against a modified version of a well-known heuristic developed originally for minimum power broadcasting. Our initial computational results are promising for FBS.

The performance of our search algorithm depends critically on a local priority rule, and we plan to devise more effective rules. In addition, power consumption for reception and processing may be incorporated for a more realistic model as we indicate in Sect. 3. Other important extensions in-

N	p = 0.5				N	p = 0.75			
	$l_i \sim U(1, N-1)$		$l_i \sim U(1, \lceil \log_2 N \rceil)$			$l_i \sim U(1, N-1)$		$l_i \sim U(1, \lceil \log_2 N \rceil)$	
	ModBIP	FBS	ModBIP	FBS	ModBIP	FBS	ModBIP	FBS	
10	$\frac{14}{16}$ (0.25, 3.74)	$\frac{16}{16}$ (0.12, 1.67)	$\frac{7}{7}$ (0.39, 3.11)	$\frac{22}{22}$ (0.08, 1.36)	$\frac{9}{9}$ (0.27, 3.01)	$\frac{22}{22}$ (0.05, 1.52)	$\frac{3}{3}$ (0.40, 3.18)	$\frac{28}{28}$ (0.01, 0.52)	
20	$\frac{15}{29}$ (0.80, 7.27)	$\frac{29}{29}$ (0.19, 3.84)	$\frac{1}{2}$ (0.52, 4.97)	$\frac{42}{42}$ (0.01, 0.58)	$\frac{9}{33}$ (0.70, 11.36)	$\frac{33}{33}$ (0.03, 0.50)	$\frac{2}{2}$ (0.24, 1.93)	$\frac{34}{34}$ (0.01, 0.66)	
30	$\frac{14}{32}$ (0.93, 6.32)	$\frac{32}{32}$ (0.16, 2.65)	$\frac{2}{2}$ (0.38, 4.34)	$\frac{37}{37}$ (0.01, 0.40)	$\frac{12}{37}$ (0.72, 5.42)	$\frac{37}{37}$ (0.06, 0.98)	$\frac{3}{3}$ (0.24, 4.10)	$\frac{31}{31}$ (0.01, 0.29)	
40	$\frac{12}{29}$ (0.98, 10.93)	$\frac{29}{29}$ (0.09, 1.00)	$\frac{3}{3}$ (0.41, 4.85)	$\frac{41}{41}$ (0.01, 0.48)	$\frac{8}{39}$ (0.82, 8.26)	$\frac{39}{39}$ (0.05, 0.64)	$\frac{1}{1}$ (0.12, 0.76)	$\frac{42}{42}$ (0.00, 0.07)	
50	$\frac{7}{42}$ (1.58, 16.88)	$\frac{42}{42}$ (0.04, 1.02)	$\frac{2}{2}$ (0.36, 4.45)	$\frac{38}{38}$ (0.01, 0.31)	$\frac{7}{40}$ (1.13, 7.34)	$\frac{40}{40}$ (0.05, 1.14)	$\frac{1}{1}$ (0.10, 0.75)	$\frac{38}{38}$ (0.00, 0.00)	

N	p = 1			
	$l_i \sim U(1, N-1)$		$l_i \sim U(1, \lceil \log_2 N \rceil)$	
	ModBIP	FBS	ModBIP	FBS
10	$\frac{6}{26}$ (0.15, 1.35)	$\frac{26}{26}$ (0.04, 1.19)	$\frac{1}{4}$ (0.27, 2.67)	$\frac{28}{28}$ (0.00, 0.04)
20	$\frac{2}{42}$ (0.46, 3.36)	$\frac{42}{42}$ (0.01, 0.23)	$\frac{4}{4}$ (0.09, 0.85)	$\frac{32}{32}$ (0.01, 0.56)
30	$\frac{5}{44}$ (0.74, 6.94)	$\frac{44}{44}$ (0.01, 0.14)	$\frac{1}{1}$ (0.24, 2.21)	$\frac{41}{41}$ (0.01, 0.69)
40	$\frac{6}{43}$ (1.05, 6.68)	$\frac{43}{43}$ (0.04, 0.66)	$\frac{4}{4}$ (0.10, 0.88)	$\frac{37}{37}$ (0.02, 0.60)
50	$\frac{4}{44}$ (0.74, 10.53)	$\frac{44}{44}$ (0.03, 1.05)	$\frac{2}{2}$ (0.11, 0.74)	$\frac{37}{37}$ (0.00, 0.07)

clude designing a distributed algorithm and exploring the trade off between transmission rates and delays.

References

1. M. Aissa and A. Ben Mnaouer. A new delay-constrained algorithm for multicast routing tree construction. *Internatinal Journal Of Communication Systems*, 17(10):985–1000, 2004.
2. F. Bauer and A. Varma. Distributed algorithms for multicast path setup in data networks. *IEEE-ACM Transactions On Networking*, 4(2):181–191, 1996.
3. M. Hubaux Galalji and C. C. JP Enz. Energy-efficient broadcasting in all-wireless networks. *Wireless Networks*, 11(1-2):177–188, 2005.
4. X. Z. Cheng, B. Narahari, R. Simha, M. X. Y.Cheng, and D. Liu. Strong minimum energy topology in wireless sensor networks: Np-completeness and heuristics. *IEEE Transactions On Mobile Computing*, 2(3):248–256, 2003.
5. Arindam K. Das, Robert J. Marks, Mohamed El-Sharkawi, Payman Arabshahi, and Andrew Gray. Minimum power broadcast trees for wireless networks: Integer programming formulations. In *Proceedings of IEEE Infocom*, 2003.
6. X. Jia, D. Li, and D. Du. Qos topology control in ad hoc wireless networks. In *Proceedings of IEEE Infocom 2004*, 2004.
7. X. H. Jia, Y. C. Zhang, N. Pissinou, and K. Makki. A distributed multicast routing protocol for real-time multicast applications. *Computer Networks*, 31(1-2):101–110, 1999.
8. C. P. Low and Y. J. Lee. Distributed multicast routing, with end-to-end delay and delay variation constraints. *Computer Communications*, 23(9):848–862, 2000.
9. Robert J. Marks, Arindam K. Das, Mohamed El-Sharkawi, Payman Arabshahi, and Andrew Gray. Minimum power broadcast trees for wireless net-

- works: Optimizing using the viability lemma. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, Scottsdale, Arizona, 2002.
10. R. Montemanni and L. M. Gambardella. Exact algorithms for the minimum power symmetric connectivity problem in wireless networks. *Computers and Operations Research*, 32(11):2891–2904, 2005.
 11. C. A. S. Oliveira and P. M. Pardalos. A survey of combinatorial optimization problems in multicast routing. *Computers and Operations Research*, 32(8):1953–1981, 2005.
 12. P. S. Ow and T. E. Morton. Filtered beam search in scheduling. *International Journal of Production Research*, 26:35–62, 1988.
 13. I. Papadimitriou and L. Georgiadis. Energy-aware broadcast trees in wireless networks. *Mobile Networks and Applications*, 9(6):567–581, 2004.
 14. G. N. Rouskas and I. Baldine. Multicast routing with end-to-end delay and delay variation constraints. *IEEE Journal On Selected Areas In Communications*, 15(3):346–356, 1997.
 15. H. F. Reeves Salama and Y. D. S. Viniotis. Evaluation of multicast routing algorithms for real-time communication on high-speed networks. *IEEE Journal On Selected Areas In Communications*, 15(3):332–345, 1997.
 16. C. S. Sung and J. M. Hong. Branch-and-price algorithm for a multicast routing problem. *Journal Of The Operational Research Society*, 50(11):1168–1175, 1999.
 17. B. Wang and S. K. S. Gupta. S-remit: A distributed algorithm for source-based energy efficient multicasting in wireless ad hoc networks. In *Proceedings of IEEE Globecom*, 2003.
 18. B. Wang and S. K. S. Gupta. Extending the lifetime of multicast trees in wanets. *Journal of Information Science and Engineering*, 20:425–447, 2004.
 19. Z. Y. Wang, B. X. Shi, and E. Zhao. Bandwidth-delay-constrained least-cost multicast routing based on heuristic genetic algorithm. *Computer Communications*, 24(7-8):685–692, 2001.
 20. J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *Proceedings of IEEE Infocom*, pages 585–594, March 2000.
 21. J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. Resource-limited energy-efficient wireless multicast of session traffic. In *Proceedings of 34th International Conference on System Sciences*, January 2000.
 22. G. L. Xue and W. Xiao. A polynomial time approximation scheme for minimum cost delay-constrained multicast tree under a steiner topology. *Algorithmica*, 41(1):53, 72 2005.
 23. W. L. Yang. Constructing energy-efficient multicast trees with delay constraints in wireless ad hoc networks. In *Proceedings of IEEE AINA 2005*, 2005.