Data Broadcast in Wireless Networks Using Approximation Algorithms

Borigarla Koteswara Rao [#1],Amanulla Mohammad[*2] ,Dr.S.Maruthu Perumal [*3]

[#1] IInd M.TECH (CSE) Student, [*2] Assistant. Professor, [*3] Professor & HOD
Department of CSE
Godavari Institute of Engineering and Technology (GIET),
Rajahmundry, AP, India.

Abstract

A wireless sensor network usually consists of a large number of sensor nodes deployed in a field. One of the major communication operations is to broadcast a message from one node to the rest of the others. Broadcasting is a fundamental operation in wireless networks and plays an important role in the communication protocol design. In multihop wireless networks, however, interference at a node due to simultaneous transmissions from its neighbors makes it nontrivial to design a minimum-latency broadcast algorithm, which is known to be NP-complete. We present a simple 12-approximation algorithm for the one-to-all broadcast problem that improves all previously known guarantees for this problem. We then consider the all-to-all broadcast problem where each node sends its own message to all other nodes. For the all-to-all broadcast problem, we present two algorithms with approximation ratios of 20 and 34, improving the best result available in the literature. Our studies indicate that our algorithms perform much better in practice than the worst-case guarantees provided in the theoretical analysis and achieve up to 37 percent performance improvement over existing schemes.

Keywords:

Wireless networks, sensor networks, Ad hoc networking, approximation algorithms, broadcast algorithms, wireless scheduling

1. Introduction

A Wireless Sensor Network (WSN) is a set of nodes deployed to sense some phenomena, collect information and send it to the base station for further processing on multihop paths. The sensor nodes in WSNs communicate via radio transmission. The broadcast nature of the radio transmission is called Wireless Broadcast Advantage (WBA) [1]. This enables a transmitting sensor node to broadcast to all the receiving nodes within its transmission range in a single transmission. However, more than one sensors transmitting simultaneously may result in interference at the receiving sensors. Hence, their transmissions need to be scheduled to avoid interference, such scheduled transmissions are said to be interference-aware transmissions. In WSNs, broadcasting from a source node to all the other nodes in the networks is one of the fundamental operations on which various distributed applications and protocols are based.

Network-wide broadcasting is a fundamental operation in wireless networks, in which a message needs to be transmitted from its source to all the other nodes in the network. There may be multiple messages to be broadcast from multiple sources. Several network protocols rely on broadcasting, for example, information dissemination, service/resource discovery, or routing in multihop wireless networks. Given that key applications of multihop wireless networks include disaster relief and rescue operations, military communication, and prompt object detection using sensors, the design of low latency broadcasting scheme is essential to meeting stringent end-to-end delay requirements for higher-level applications.

Interference is a fundamental limiting factor in wireless networks. When two or more nodes transmit a message to a common neighbor at the same time, the common node will not receive any of these messages. In such a case, we say that collision has occurred at the common node. Interference range may be even larger than the transmission range, in which case a node may not receive a message from its transmitter if it is within the interference range of another node sending a message. Any communication protocol for wireless networks should contend with the issue of interference in the wireless medium.

One of the earliest broadcast mechanisms proposed in the literature is flooding [2], where every node in the network transmits a message to its neighbors after receiving it. Although flooding is extremely simple and easy to implement, Ni *et al.* [3] show that flooding can be very costly and can lead to serious redundancy, bandwidth contention, and collision: a situation known as ***broadcast storm***. Since then, a large amount of research has been directed towards designing broadcast protocols which are collision-free and which reduce redundancy by reducing the number of transmissions. In this paper, we revisit the data broadcast problem and present improved algorithms that guarantee collision-free delivery and achieve low latency.

1.1. Our Contributions

We present algorithms for ONE-TO-ALL and ALL-TO-ALL broadcasting problems. In one-to-all broadcast, there is a source that sends a message to all other nodes in the network. In all-to-all broadcast each node sends its own message to all other nodes. Even the one-to-all broadcasting problem is known to be NP-complete [4]. For both problems, we develop approximation algorithms, which improve the previous results.

- For ONE-TO-ALL BROADCAST problem, we present a simple approximation algorithm (Section IV) that achieves a 12-approximate solution, thereby improving the approximation guarantee of 16 due to Huang *et al.* [5]. Our algorithm is based on the algorithm of Gandhi *et al.* [4] and incorporates the following two ideas that lead to the improvement: (i) processing the nodes greedily – in non-increasing order of the number of receivers, and (ii) allowing nodes to transmit more than once. The latter is particularly counter-intuitive as one would expect that the latency would increase if a node transmits more than once. Note that in [4] the analysis of their algorithm gives an approximation guarantee that is greater than 400.

- We then consider the ALL-TO-ALL BROADCAST problem and present two algorithms (called CDA and ICDA) with approximation guarantees of 20 and 34 respectively (Section V), thereby improving the approximation guarantee of 27 by Huang et al. [6]. Our improved result is due to efficient scheduling techniques to collect data and then perform pipelined broadcasting. In ICDA, all nodes are scheduled to participate in transmissions as early as possible. Even though its theoretical bound is weaker than that of CDA, experimental results show that it provides comparable or better performance than CDA, especially in larger networks.

- We study the performance of our broadcast algorithms through simulations under various conditions. Our results indicate that our algorithms perform much better in practice than the worst case guarantees provided. Our algorithms achieve up to 37% improvement on end-to end latency over existing schemes.

2. Related Work

Several techniques have been proposed for broadcasting in wireless networks. In order to reduce the broadcast redundancy and contentions, they make use of nodes' neighborhood information and determine whether a particular node needs to transmit a message [7–14]. There has been some work on latency-constrained broadcasting in wired networks [15] and some results do exist for radio networks whose models are essentially the same as ours. In particular, Chlamtac and Kutten [16] show that minimum latency broadcast scheduling is NP-Complete for general (non-geometric) graphs. This result does not directly extend to ad hoc networks which are modeled by a restricted class of geometric graphs called disk graphs. Chlamtac and Weinstein gave an algorithm for efficient broadcasting in multihop radio networks. They proved that for arbitrary graphs, the broadcast latency of their schedule is within O (ln (N/r) 2) times the optimal, where N is the number of network nodes and r is the maximum distance from the source to any other node.

Basagni *et al.* present a mobility transparent broadcast scheme for mobile multi-hop radio networks. In their scheme, nodes compute their transmit times once and for all in the beginning. They provide two schemes with bounded latency. These schemes have approximation factors which are linear and polylogarithmic in the number of network nodes. In effect, they assume that the topology of the network is completely unknown. Although their schemes are attractive for highly mobile environments, their approximation factors are far from what is achievable in static and relatively less mobile environments where the broadcast tree and schedule can be computed efficiently.

Chen *et al.* [17] also address the problem of minimizing broadcast latency when the interference range is strictly larger than the transmission range. If α is the ratio of the interference range to the transmission range, then for $\alpha > 1$, they give an $O(\alpha^2)$- approximation algorithm. In particular, when $\alpha = 2$, their algorithm achieves a 26-approximation. However, it is not clear how their algorithm behaves when $\alpha = 1$. For all-to-all broadcast problem, Gandhi et al. [4] present a constant approximation algorithm where the constant factor is quite large (> 1000). Tiwari *et al.* consider the one-to-all broadcast problem in three dimensional space. Mahjourian *et al.* present an approximation algorithm when both interference range and carrier sensing ranges are larger than transmission range. The all-to-all broadcast algorithm by Huang et al. [6] achieves the approximation factor of 27. In this work, we further improve the approximation guarantee for the all-to-all broadcasting.

Hung *et al.* provide centralized and distributed algorithms for broadcasting and experimental study of their algorithms with respect to collision-free delivery, number of transmissions and broadcast latency. While their centralized algorithm is guaranteed to be collision-free, their distributed algorithm is not. They do not provide any guarantees with respect to the number of transmissions and latency of the broadcast schedule. Williams and Camp survey many wireless broadcast protocols discussed above. They provide a neat characterization and experimental evaluation of many of these protocols under a wide range of network conditions.

3. Preliminaries

A. Network Model

When the interference range and the transmission range are identical, a wireless network can be modeled as a unit disk graph (UDG), G = (V, E). The nodes in V are embedded in the plane. Each node u € V has a unit transmission range. Let |u, v| denote the Euclidean distance between u and v. Let D (u) denotes the neighbors of u in G. A node v € D (u) iff $|u, v| \leq 1$.

We assume that time is discrete. Since the medium of transmission is wireless, whenever a node transmits a message, all its neighbors hear the message. We assume that every message transmission occupies a unit time slot: i.e., the latency of a single successful transmission is one unit of time. We say that there is a collision at node w, if w hears a message from two transmitters at the same time. In such a case, we also say that the two transmissions interfere at w. A node w receives a message collision-free iff w hears the message without any collision. We also consider the case when the interference range is strictly larger than the transmission range. Let α denote the ratio of the interference range to the transmission range. Consider nodes u and w such that $1 < |u,w| \leq \alpha$.When w broadcasts a message, even though u will not receive the message correctly (since it is not in D(w)), this can prevent node u from receiving a message broadcast from a node in D(u). Thus, for a node u to receive a message collision-free, a node in D(u) must transmit the message and no other node within a distance of α from u must transmit the message.

B. Problem Statement

We are given a disk graph G = (V, E) and a set of messages M = {1, 2. . . m}. We also have a set of sources for these messages: sources = {sj |sj is the source of message j}. A node can transmit message j only after it receives message j collision-free. A *schedule* specifies, for each message j and each node i, the time at which node i receives message j collision-free and the time at which it transmits message j. If a node does not transmit a message then its transmit time for that message is 0. The latency of the broadcast schedule is the first time at which every node receives all messages. The number of transmissions is the total number of times every node transmits any message. Our goal is to compute a schedule in which the latency is minimized.

We consider one-to-all and all-to-all broadcasting problems. One-to-all broadcasting is the operation where there is one source node s which has a message to send all other nodes. In all-to-all broadcasting, each node v has its own message m (v) to send all other nodes. Even the one-to-all broadcasting problem is known to be NP-complete.

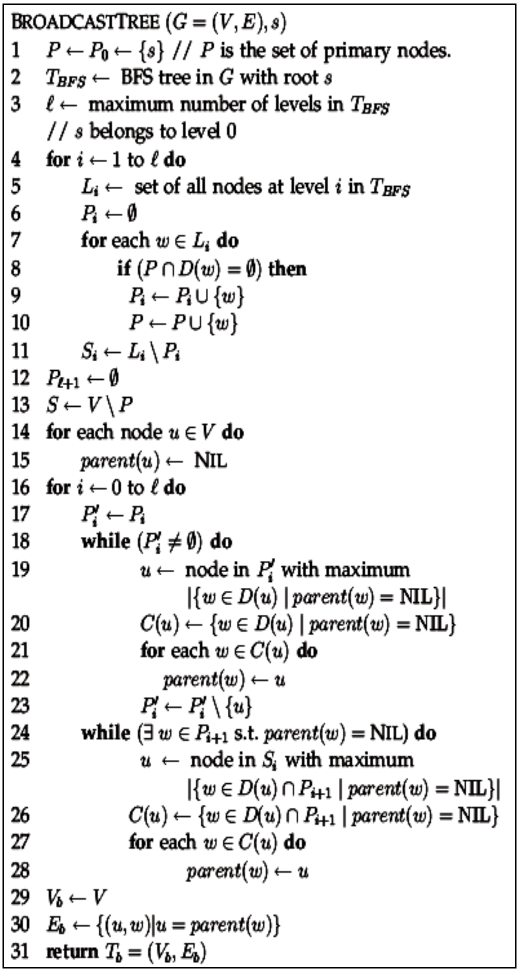
4. ONE-TO-ALL Broadcast Algorithm

The algorithm takes as input a UDG G = (V, E) and a source node s. The algorithm first constructs a *broadcast tree*, Tb, rooted at s in which if a node u is a parent of a node w then u is responsible for transmitting the message to w without any collision at w. It then schedules the transmissions so that every node receives the message collision-free. The two key differences from the algorithm in [4] that lead to a significantly improved approximation guarantee are:

(i) Processing the nodes in a greedy manner while constructing the broadcast tree.

(ii) Allowing a node to transmit more than once.

Both these properties are crucial to the proof, which is central to showing that our algorithm yields a 12- approximate solution. Note that in [4] the analysis of their algorithm gives an approximation ratio of at least 400.



```
BROADCASTTREE ($G = (V, E), s$)
$P \leftarrow P_0 \leftarrow \{s\}$ // $P$ is the set of primary nodes.
$T_{BFS} \leftarrow$ BFS tree in $G$ with root $s$
$\ell \leftarrow$ maximum number of levels in $T_{BFS}$
// $s$ belongs to level 0
for $i \leftarrow 1$ to $\ell$ do
    $L_i \leftarrow$ set of all nodes at level $i$ in $T_{BFS}$
    $P_i \leftarrow \emptyset$
    for each $w \in L_i$ do
        if ($P \cap D(w) = \emptyset$) then
            $P_i \leftarrow P_i \cup \{w\}$
            $P \leftarrow P \cup \{w\}$
    $S_i \leftarrow L_i \setminus P_i$
$P_{\ell+1} \leftarrow \emptyset$
$S \leftarrow V \setminus P$
for each node $u \in V$ do
    $parent(u) \leftarrow$ NIL
for $i \leftarrow 0$ to $\ell$ do
    $P'_i \leftarrow P_i$
    while ($P'_i \neq \emptyset$) do
        $u \leftarrow$ node in $P'_i$ with maximum
            $|\{w \in D(u) \mid parent(w) = \text{NIL}\}|$
        $C(u) \leftarrow \{w \in D(u) \mid parent(w) = \text{NIL}\}$
        for each $w \in C(u)$ do
            $parent(w) \leftarrow u$
        $P'_i \leftarrow P'_i \setminus \{u\}$
    while ($\exists\, w \in P_{i+1}$ s.t. $parent(w) = \text{NIL}$) do
        $u \leftarrow$ node in $S_i$ with maximum
            $|\{w \in D(u) \cap P_{i+1} \mid parent(w) = \text{NIL}\}|$
        $C(u) \leftarrow \{w \in D(u) \cap P_{i+1} \mid parent(w) = \text{NIL}\}$
        for each $w \in C(u)$ do
            $parent(w) \leftarrow u$
$V_b \leftarrow V$
$E_b \leftarrow \{(u, w) | u = parent(w)\}$
return $T_b = (V_b, E_b)$
```

In ONE-TO-ALL BROADCAST, the transmissions are scheduled in two phases. In Phase 1, the algorithm schedules transmissions only to the nodes in set (denoted by X) which contains all primary nodes and non-leaf secondary nodes in Tb. In Phase 2, transmissions are scheduled to send the message to all other nodes. *Note that this leads to some nodes transmitting more than once* which is again a significant departure from the algorithm in [4] in which each node transmits the message at most once. The intuition behind this is that it is not necessary to send a message to terminal nodes early as they are not responsible for relaying the message further. On the other hand, by reducing the number of recipients in the first phase, a node will need to avoid a smaller number of potential conflicts before sending a message to nonterminal nodes, thus reducing the broadcast time.

In Phase 2, transmissions are scheduled so that the nodes in Y = V \ X receive the message. Nodes are considered one level at a time. For each v € Y, parent (v) is responsible for transmitting the message collision-free to v. Since P ∩ Y = ø, the secondary nodes do not transmit in Phase 2. Any transmitting node, u, transmits at the minimum time t that satisfies the above three collision-free constraints.

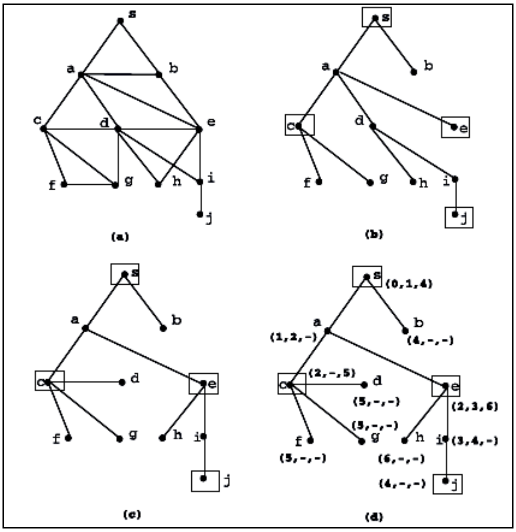
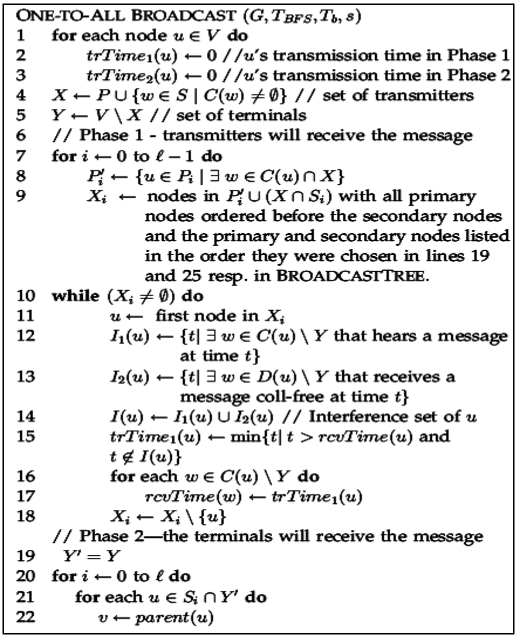
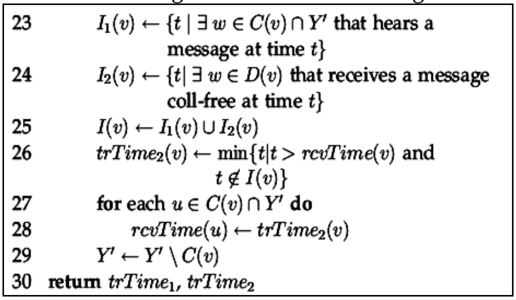


Fig. 1. An illustration of our algorithm. (a) Shows the example network. (b) Shows the BFS tree, TBFS along with the primary nodes (highlighted). (c) Shows the broadcast tree, Tb. (d) Shows the transmission schedule. Besides each node is a 3-tuple, whose members are rcvTime(.), trTime$_1$(.), and trTime $_2$(.), respectively. For instance, source node receives a message at time 0 (as it is the original source of the message) and transmits at time 1 for Phase 1 and at time 4 for Phase 2.



```
ONE-TO-ALL BROADCAST $(G, T_{BFS}, T_b, s)$
for each node $u \in V$ do
    $trTime_1(u) \leftarrow 0$ // $u$'s transmission time in Phase 1
    $trTime_2(u) \leftarrow 0$ // $u$'s transmission time in Phase 2
$X \leftarrow P \cup \{w \in S \mid C(w) \neq \emptyset\}$ // set of transmitters
$Y \leftarrow V \setminus X$ // set of terminals
// Phase 1 - transmitters will receive the message
for $i \leftarrow 0$ to $\ell - 1$ do
    $P'_i \leftarrow \{u \in P_i \mid \exists\, w \in C(u) \cap X\}$
    $X_i \leftarrow$ nodes in $P'_i \cup (X \cap S_i)$ with all primary
          nodes ordered before the secondary nodes
          and the primary and secondary nodes listed
          in the order they were chosen in lines 19
          and 25 resp. in BROADCASTTREE.
while $(X_i \neq \emptyset)$ do
        $u \leftarrow$ first node in $X_i$
        $I_1(u) \leftarrow \{t \mid \exists\, w \in C(u) \setminus Y$ that hears a message
              at time $t\}$
        $I_2(u) \leftarrow \{t \mid \exists\, w \in D(u) \setminus Y$ that receives a
              message coll-free at time $t\}$
        $I(u) \leftarrow I_1(u) \cup I_2(u)$ // Interference set of $u$
        $trTime_1(u) \leftarrow \min\{t \mid t > rcvTime(u)$ and
        $t \notin I(u)\}$
        for each $w \in C(u) \setminus Y$ do
            $rcvTime(w) \leftarrow trTime_1(u)$
        $X_i \leftarrow X_i \setminus \{u\}$
// Phase 2—the terminals will receive the message
 $Y' = Y$
for $i \leftarrow 0$ to $\ell$ do
   for each $u \in S_i \cap Y'$ do
        $v \leftarrow parent(u)$
```

Fig. 1 illustrates our algorithm. Note that for any node, the rcvTime (.) that is shown in the figure is the time at which the node is guaranteed to receive the message collision free in our algorithm.



```
            $I_1(v) \leftarrow \{t \mid \exists\, w \in C(v) \cap Y'$ that hears a
                  message at time $t\}$
            $I_2(v) \leftarrow \{t \mid \exists\, w \in D(v)$ that receives a message
                  coll-free at time $t\}$
            $I(v) \leftarrow I_1(v) \cup I_2(v)$
            $trTime_2(v) \leftarrow \min\{t \mid t > rcvTime(v)$ and
                          $t \notin I(v)\}$
            for each $u \in C(v) \cap Y'$ do
                 $rcvTime(u) \leftarrow trTime_2(v)$
            $Y' \leftarrow Y' \setminus C(v)$
return $trTime_1$, $trTime_2$
```

For example, consider node b. While b receives the message collision free at time 1, in our algorithm it is guaranteed to receive the message collision free at time 4. Similarly, nodes d and h receive message collision free at time 3, but in our algorithm they are guaranteed to receive message collision free at times 5 and 6, respectively. While it is easy to eliminate this slackness from our algorithm, we leave it as is for clarity in exposition.

4.1 General Interference Range Model

Our algorithm and analysis can be easily extended to the case when the interference range of a node is different than its transmission range. The only changes are in lines 13 and 24 of the pseudocode **ONE-TO-ALL BROADCAST**, where D(u) (line 13) and D(v) (line 24) are to be replaced by “interference range of u” and “interference range of v,” respectively. Note that if α, the ratio of interference range to the transmission range is a constant, then so is |Dp (v, α +1)|(note that Dp(v, α +1) remains the same). The rest of the analysis is very similar to the case when α = 1, only the values of some constants will change.

5. ALL-TO-ALL Broadcast Algorithm

We now consider all-to-all broadcasting, in which each node v has a message m(v) to send to all other nodes. We present two algorithms. By adopting efficient scheduling scheme for pipelined broadcasting, the first algorithm achieves an approximation guarantee of 20, which improves the previous best known guarantee of 27 in the literature [6].The second algorithm achieves an approximation factor of 34, which performs well in our experiments.

5.1 Collect-and-Distribute Algorithm (CDA)

The graph radius of G with respect to a node v is the maximum depth of the BFS tree rooted at v. A graph center of G is a node in G with respect to which the graph radius of G is the smallest. Let s be a graph center of G, and R be the graph radius of G with respect to s. Clearly, δc ≥ R. We call transmissions of message m from a node v upward if the message m is originated from the descendant of v. Otherwise, a transmission is called downward. Our schedule consists of two phases. In Phase 1, s collects all the packets by performing upward transmissions. In the Phase 2, s broadcasts all the n packets to all other nodes via downward transmissions.

Phase 1. Node s collects all messages by using the data collection algorithm based on the one by Florens and McEliece [18]. We simplify their algorithm as follows: first construct a BFS tree from s, and sort messages m(v) in non decreasing order of the level of v in the BFS tree. That is, messages that are closer to s appear first in the sorted list. Let us assume that message j be the jth message in the sorted order. We now greedily schedule transmissions by giving priority to message j over any message i > j. The latency of the collection algorithm is at most 3(n- 1) [18].

Phase 2. We construct a broadcast tree Tb using BROADCASTTREE in Section 4. Next, we describe transmission scheduling algorithm. In the algorithm by Gandhi et al. [4], the root node collects all messages and perform one-to-all broadcasting for each message. The root node needs to wait until the previous message reaches L3 before initiating a broadcast for another message to make sure there are no collisions in their algorithm. In our algorithm, we find a schedule by a vertex coloring, which makes sure that all the nodes with the same color can broadcast a message without collision, and show that 17 colors are enough to obtain a collision-free schedule.

5.2 Interleaved Collect-and-Distribute Algorithm

In the 20-approximation algorithm proposed in Section 5.1, all messages are first sent to the root node s in the broadcast tree, and then s sends the messages one by one. Note that in the early stages of the algorithm, until s receives all the messages and starts propagating them, most nodes are idle, thus increasing the broadcast time significantly. We now describe an algorithm in which all nodes participate in broadcasting as soon as possible so as to minimize the broadcast time. The main idea is as follows: suppose that a node v receives a message m(x) forwarded originally from its descendant x in the broadcast tree and relays it to its parent to deliver the message to the root node s. Note that the children of v can also receive the message when v broadcasts it and therefore, they can initiate broadcasting m(x) in their own subtrees in parallel without waiting for the message forwarded from s.

Using the broadcast tree constructed as in CDA, we schedule transmissions for each node as follows: as in CDA, we define a super step but in a slightly different way. That is, in each super step, every node transmits at most one message (either upward or downward) if there is any message that the node received but not sent. Instead of finishing all upward transmissions first, we mix upward or downward transmissions in each superstep with preferences given to upward transmissions. Also for an upward transmission, a node should make sure that its parent and all of its children (except the one which sent the message to v) receive the message. For a downward transmission, v is responsible for sending the message to all of its children.The scheduling algorithm is as follows:

1. Transmissions from terminal nodes. Before starting supersteps, all terminal nodes Y send messages to their parents one by one.

2. Transmissions from internal nodes. In each superstep, each node in X transmits one message if there is any message received but not sent. Each node can receive at most one upward message from its children. Therefore, a node can perform an upward transmission only if its parent has not received an upward message in the superstep. Otherwise, it performs a downward transmission. For each superstep, the algorithm performs the following.

a. Transmissions from primaries. Primaries are scheduled before secondaries. Transmissions from primaries are scheduled based on the vertex-coloring of H1 and the order to process nodes is the same as in CDA. Recall that a node performs an upward transmission if its parent has not received a message from its sibling in the same superstep. Otherwise, it performs a downward transmission. Transmissions from primaries require at most 12 time slots.

b. Upward transmissions from secondaries. Secondary nodes are considered in the same order as the broadcast tree is constructed, and a node can perform an upward transmission if its parent has not received a message from its sibling in the same superstep. Upward transmissions from secondaries require at most 16 time slots as shown below.

c. Downward transmissions from secondaries. Once all upward transmissions are scheduled, nodes which are not scheduled for upward transmissions are considered in the same order as the broadcast tree is constructed, and downward transmissions are scheduled. Downward transmissions require at most five time slots. Below we prove that this algorithm yields a 34-approximation. The following fact will be useful for the analysis of 34-approximation.

6. Conclusion

In this paper, we presented approximation algorithms for broadcasting in multihop wireless networks. Our algorithm for ONE-TO-ALL BROADCASTING gives a 12-approximate solution, and the algorithms for ALL-TO-ALL BROADCASTING give approximation ratios of 20 and 34. Our simulation results show that in practice, these proposed schemes perform much better than the theoretical bound and achieve up to 37 percent latency performance improvement over existing schemes.

7. References

[1] Efficient Broadcasting in Multi-Radio Multi-channel, and Multi-hop wireless network based on self pruning. High Performance Computing and Communications (HPCC 2007), Sept 2007.

[2] J. Jetcheva, Y. Hu, D. Maltz, and D. Johnson, “A simple protocol for multicast and broadcast in mobile ad hoc networks,” July 2001.

[3] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, “The broadcast storm problem in a mobile ad hoc

network," in *Proc. of the 5th Ann. ACM/IEEE Int. Conf. on Mobile Comp. and Net.*, 1999, pp. 151–162.

[4] R. Gandhi, A. Mishra, and S. Parthasarathy, "Minimizing broadcast latency and redundancy in ad hoc networks," *IEEE/ACM Transactions on Networking*, pp. 840–851, Aug 2008.

[5] S. Huang, P. Wan, X. Jia, H. Du, and W. Shang, "Minimum-latency broadcast scheduling in wireless ad hoc networks," in *Proceedings of INFOCOM*, 2007, pp.

[6] S. C.-H. Huang, H. Du, and E.-K. Park, "Minimum-latency gossiping in multi-hop wireless networks," in *Proceedings of MobiHoc*. New York, NY, USA: ACM, 2008, pp. 323–330.

[7] H. Lim and C. Kim, "Multicast tree construction and flooding in wireless ad hoc networks," in *Proc. of the 3rd ACM Int. Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Sys*, 2000, pp. 61–68.

[8] W. Peng and X.-C. Lu, "On the reduction of broadcast redundancy in mobile adhoc networks," in *Proceedings of MobiHoc*, Aug. 11, 2000.

[9] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying: An efficient technique for flooding in mobile wireless networks," INRIA, Tech. Rep. Research Report RR-3898, Feb. 2000.

[10] W. Peng and X. Lu, "Ahbp: An efficient broadcast protocol for mobile adhoc networks," *J. of Science and Tech. - Beijing, China*, 2000.

[11] J. Sucec and I. Marsic, "An efficient distributed network-wide broadcast algorithm for mobile adhoc networks," Rutgers U., Tech. Rep., 2000.

[12] W. Peng and X. Lu, "Efficient broadcast in mobile ad hoc networks using connected dominating sets," *J. of Software - Beijing, China*, 1999.

[13] I. Stojmenovic, M. Seddigh, and J. Zunic, "Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks," *IEEE Trans. on Parallel and Distributed Systems*, pp. 14–25, Jan 2002.

[14] L. Orecchia, A. Panconesi, C. Petrioli, and A. Vitaletti, "Localized techniques for broadcasting in wireless sensor networks," in *Proceedings of DIALM-POMC*, New York, NY, USA, 2004, pp. 41–51.

[15] H. F. Salama, D. S. Reeves, and Y. Viniotis, "The delay-constrained minimum spanning tree problem," in *Second IEEE Symp. on Comp. and Comm.* IEEE Computer Society, 1997, pp. 699–704.

[16] I. Chlamtac and S. Kutten, "On broadcasting in radio networks – problem analysis and protocol design," *IEEE Trans. on Comm.*, vol. 33, no. 12, pp. 1240–1246, 1985.

[17] Z. Chen, C. Qiao, J. Xu, and T. Lee, "A constant approximation algorithm for interference aware broadcast in wireless networks," in *Proceedings of INFOCOM*, 2007, pp. 740–748.

[18] C. Florens and R. McEliece, "Packets distribution algorithms for sensor networks," in *INFOCOM*, 200, pp. 1063 – 1072.

8. About the Authors

Borigarla Koteswara Rao is currently pursuing his M.Tech (CSE) in Computer Science & Engineering Department, GIET, Rajahmundry. His area of interests includes Mobile Computing.

Amanulla Mohammad is currently working as an Assistant Professor in Computer Science & Engineering Department, GIET, Rajahmundry. His research interests include Data Mining.

Dr. S. Maruthu Perumal is currently working as a Head of the Department for Computer Science & Engineering Department, GIET, Rajahmundry. He is awarded with PhD in related field. His research interests include Image Processing, Data Mining & Warehousing, Networks and Security, Software Engineering.