

Dominating Sets and Neighbor Elimination-Based Broadcasting Algorithms in Wireless Networks

Ivan Stojmenovic, Mahtab Seddigh, and Jovisa Zunic

Abstract—In a multihop wireless network, each node has a transmission radius and is able to send a message to all of its neighbors that are located within the radius. In a broadcasting task, a source node sends the same message to all the nodes in the network. In this paper, we propose to significantly reduce or eliminate the communication overhead of a broadcasting task by applying the concept of localized dominating sets. Their maintenance does not require any communication overhead in addition to maintaining positions of neighboring nodes. Retransmissions by only internal nodes in a dominating set is sufficient for reliable broadcasting. Existing dominating sets are improved by using node degrees instead of their *ids* as primary keys. We also propose to eliminate neighbors that already received the message and rebroadcast only if the list of neighbors that might need the message is nonempty. A retransmission after negative acknowledgements scheme is also described. The important features of proposed algorithms are their reliability (reaching all nodes in the absence of message collisions), significant rebroadcast savings, and their localized and parameterless behavior. The reduction in communication overhead for broadcasting task is measured experimentally. Dominating sets based broadcasting, enhanced by neighbor elimination scheme and highest degree key, provides reliable broadcast with ≤ 53 percent of node retransmissions (on random unit graphs with 100 nodes) for all average degrees d . Critical d is around 4, with ≤ 48 percent for ≤ 3 , ≤ 40 percent for $d \geq 10$, and ≤ 20 percent for $d \geq 25$. The proposed methods are better than existing ones in all considered aspects: reliability, rebroadcast savings, and maintenance communication overhead. In particular, the cluster structure is inefficient for broadcasting because of considerable communication overhead for maintaining the structure and is also inferior in terms of rebroadcast savings.

Index Terms—Broadcasting, wireless networks, distributed algorithms, dominating sets, clustering.



1 INTRODUCTION

WIRELESS networks consist of static or mobile hosts (or nodes) which can communicate with each other over the wireless links without any static network interaction. Each mobile host has the capability to communicate directly with another mobile hosts in its vicinity. They can also forward packets destined for other nodes. Examples of such networks are ad hoc, local area, packet radio, and sensor networks, which are used in disaster rescues, wireless conferences in the hall, battlefields, monitoring objects in a possibly remote or dangerous environment, wireless Internet, etc.

In a broadcasting task, a source node sends the same message to all the nodes in the network. In one-to-all model, transmission by each node can reach *all* nodes that are within radius distance from it, while in the one-to-one model (which we studied in a companion paper [29]), each transmission is directed toward only *one* neighbor (using, for instance, directional antennas or separate frequencies for each node).

The broadcasting in literature has been studied mainly for one-to-all model and we will use that model in this paper. Broadcasting is frequently referred to as *flooding*. We will use this term to refer to the broadcasting scheme in which the same message is retransmitted by *all* nodes that receive it. Flooding has been traditionally used for broadcasting. Broadcasting applications include paging a particular host or sending an alarm signal. Flooding/broadcasting is also used for route discovery in a source-initiated on-demand routing (e.g., [2]). Broadcasting can similarly be used in the context of an efficient location-aware routing algorithm as follows: The source S may initiate the destination search process by broadcasting a short message that contains the location of S , *id* of destination D , and some control bits. When the destination search message reaches successfully D , D applies any location-based routing algorithm (e.g., [3] which guarantees delivery if location of destination, in this case S , is accurate) and reports back to S with a short message containing its location. The source S can then apply again the same routing algorithm [3] (or use path created in the previous step by D if that path was recorded in the process) to send the full message toward D . Ho et al. [14] argued that flooding can be a viable candidate for multicast and routing protocols in very dynamic ad hoc networks.

Flooding has been replaced in [15], [23] by a method where each clusterhead and gateway (or border, as renamed in this paper) node in a clustered wireless network forwards the message exactly once. The main-

- I. Stojmenovic is with SITE, University of Ottawa, Ontario K1N 6N5, Canada. E-mail: ivan@site.uottawa.ca.
- M. Seddigh is with Nortel Networks, Ottawa, Ontario, Canada. E-mail: mseddigh@nortelnetworks.com.
- J. Zunic is with the Computer Science Department, Cardiff University, Queen's Buildings, Newport Rd., P.O. Box 916, Cardiff CF24 3XF, Wales, UK. E-mail: J.Zunic@cs.cf.ac.uk.

Manuscript received 9 Nov. 2000; revised 15 June 2001; accepted 29 June 2001.

For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number 113130.

tenance of cluster structure, however, requires excessive communication overhead due to “chain effect” caused by mobility [11], [34]. The concept of internal nodes is proposed in [34]. We propose in this paper to restrict broadcasting to internal nodes only since any other node is directly linked to an internal node. Retransmission savings are confirmed experimentally in addition to reduced communication overhead savings of locally determined internal nodes with respect to cluster structure. For the purpose of the comparison, we have also described a broadcast efficient clustering algorithm, where node degree was used as the primary key and node *id* as the secondary key in clusterhead decisions. We have also applied highest node degree to improve the concept of internal nodes. In addition, a neighbor elimination scheme, which prevents a node from retransmitting if all its neighbors already received the same message, is also proposed.

Ad hoc networks are best modeled by the *unit* graphs constructed in the following way: Two nodes *A* and *B* in the network are neighbors (thus joined by an edge) if the Euclidean distance between them is at most *R*, where *R* is the transmission radius which is equal for all nodes. We have used random unit graphs in our experiments.

The broadcasting algorithms in [18], [19] achieve a high ratio of nodes receiving the message with a reduced amount of rebroadcasting. Each of the methods from [18], [19] has a parameter whose best value may depend on network conditions, which is a global information. Further, their methods are not reliable. We refer to a broadcasting method as *reliable* if the method guarantees message delivery to all nodes connected to the source, assuming that there are no message collisions. That is, if an ideal medium access scheme that will guaranty delivery of each message to all intended neighboring nodes without collisions at any of these nodes is available. One such scheme is time division multiple access, where each node is assigned a time slot which is different from time slots assigned to any of its one or two-hop neighbors [16]. By reducing the broadcast redundancy [18], [22], one can reduce the packet loss due to contention or collision and potentially enhance the reliability of broadcasting. The multipoint relay method, recently proposed in [24], is reliable.

The main contributions of this paper are reliable internal-node-based broadcasting algorithms, enhanced by neighbor elimination scheme, highest degree priority in selecting internal nodes, and retransmission after negative acknowledgements scheme. Internal node maintenance is incorporated into location updates between neighboring nodes if GPS or another location method is available to all the nodes in the network. Global Position System (GPS) provides location information (latitude, longitude, and, possibly, height) to hosts in a wireless network by communication with a satellite network. Alternatively, nodes may measure signal strengths of incoming messages and determine the location of its neighbors by exchanging signal strength information with their neighbors.

The concept of *localized* algorithms was proposed in [7] as distributed algorithms, where simple local node behavior achieves a desired global objective. Our broadcasting

algorithms are localized since each node needs only the location of its neighbors or, alternatively, the list of neighbors of each neighbor. They are also degree independent in two senses: There is no parameter in the algorithms that is set according to the network average degree *d* and the performance of proposed algorithms appears to be relatively stable with respect to *d*.

A relevant literature review is given in Section 2. Section 3 describes new broadcasting algorithms. Performance evaluations are given in section 4, followed by a conclusion.

2 LITERATURE REVIEW

2.1 Broadcasting Algorithms

Pagani and Rossi [23] described a broadcasting protocol for ad hoc networks that is similar to the address searching algorithm by Lauer [15]. It is based on a clustered organization of nodes. Nodes are divided into clusters with one of them serving as clusterhead in each cluster. Each clusterhead has a direct link to any of the nodes in its cluster. Thus, two nodes in the same cluster have hop distance of at most two. In the broadcasting protocol [23], [15], the source node forwards the message to its clusterhead (CH), which then initiates the construction of the virtual spanning tree of all CHs (two neighboring CHs are at a hop distance of two) by forwarding the message to all of them. More precisely, the message is sent to all neighboring CHs which in turn forward it to their neighboring CHs. Nodes do not forward duplicate copies of the same message. All CHs broadcast the message (with one transmission) to the nodes in its own cluster. The broadcasting algorithms [23], [15] use clustering algorithms [9], [12] that are not always able to fully cluster a given set [5].

The distributed clustering algorithm by Lin and Gerla [17] is initiated by all nodes whose *id* is lowest among all their neighbors (local lowest *id* nodes). They broadcast their decision to create clusters (with them as CHs) to all their neighbors. Each node may hear the broadcasts by its neighbors and select the lowest *id* among neighboring CHs, if any. If all neighbors which have lower *id* sent their decisions and none declared itself a CH, the node decides to create its own CH and broadcasts its *id* as cluster *id*. Otherwise, it chooses neighboring CH with lowest *id* and broadcasts such decision. Thus, each node broadcasts its clustering decision after all its neighbors with lower *ids* have already done so. Every node can determine its cluster and only one cluster and transmits exactly one message during the algorithm. A sophisticated maintenance procedure for cluster formation when nodes move is described in [17].

In order to provide a fair comparison with a cluster-based broadcasting algorithm, we used here a combined higher connectivity and lower *id* clustering algorithm [5]. Each node is assigned a pair $did = (d, id)$ (where *d* is node degree), called clusterhead priority. Let $did' = (d', id')$ and $did'' = (d'', id'')$. Then, $did' > did''$ if $d' > d''$ or ($d' = d''$ and $id' < id''$). That is, a node has clusterhead priority over the other if it has higher connectivity or, in case of equal connectivity, has lower *id*. The algorithm then follows the algorithm [17], where lower *id* clusterhead priority is

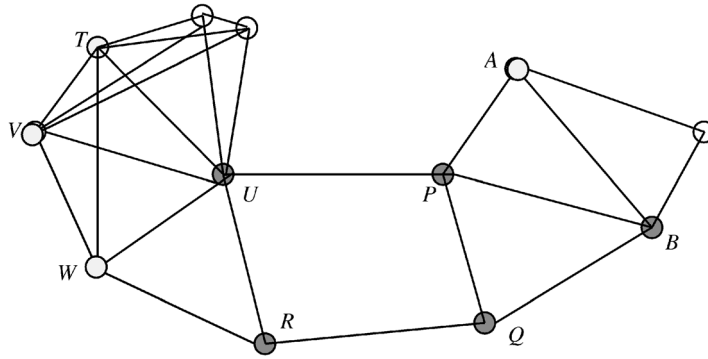


Fig. 1. Intermediate nodes *AVWBUPQRT*, intergateway and gateway nodes *BUPQR*.

replaced with higher *did* clusterhead priority described here. This algorithm will be referred to as the *ConID* algorithm.

This algorithm [5] produces a reduced number of CHs and border nodes, as shown experimentally. Border nodes are nodes that have neighbors that belong to another cluster. In a cluster-based broadcasting algorithm, each CH and border node will transmit the message exactly once. Therefore, their total number needs to be optimized. For example, in Fig. 1, there are two CH nodes (*U* and *B*) and three border nodes (*P*, *Q*, and *R*) when this new algorithm (with *x*-coordinate serving as node *id*) is applied. The lowest-*id* algorithm [LT] produces four CHs (*V*, *R*, *P*, and the rightmost node) and five border nodes (all except the top three). Note that the count applies only if broadcasting is initiated by a node that is supposed to retransmit by the method. Otherwise (that is, if noninternal node or a node that is not a CH or border node is the source for a broadcasting task), the count should be increased by one for transmitting the message from the source to one of its neighbors that is selected for broadcasting by the method. However, to simplify the comparison, we shall ignore that possible count difference in our experiments.

Ni et al. [18], [19] studied the broadcast storm problem. A straightforward broadcasting by flooding is usually very costly and will result in serious redundancy, contention, and collision. They identified this broadcast storm problem by showing how serious it is through analyses and simulations. Several schemes (probabilistic, counter-based, distance-based, location-based, and cluster-based) to reduce redundant rebroadcasts and differentiate timing of rebroadcasts to alleviate this problem are proposed in [18], [19]. These schemes achieve a high percentage of delivery rate with a low number of retransmissions. However, they are not reliable. In the probabilistic scheme [18], each node rebroadcasts the first copy of a received message with a given probability *p*. In the counter-based scheme [18], each node rebroadcasts the message if and only if it received the message from less than *C* neighbors. In the distance-based scheme [18], the message is retransmitted if and only if the distance to each neighbor that already retransmitted the message is $> D$. In the location-based scheme [18], the message is retransmitted if and only if the additional area that can be covered if the node rebroadcasts the message (divided by the area of circle with transmission radius) is greater than the threshold *A*. A simplified version of the method is to rebroadcast the message if the node is not

located inside the convex hull of neighboring nodes that already retransmitted the message. In the cluster-based scheme, lowest ID clustering algorithm [17] is applied and one of the above four methods is then applied on cluster-head and border nodes.

Experiments in [18] have measured reachability *RE* (that is, delivery rate, the ratio of nodes receiving the message), the saved rebroadcasts *SRB* (the ratio of nodes that do not rebroadcast the message), and the average latency until the last host receives the message. One hundred nodes are placed in an area of varying size and fixed transmission radius *R*, resulting in graphs with varying average degrees. The best choice of parameter for each scheme was dependent on average node degree, which is an information of global nature and not locally available to nodes. Location-based method is, according to authors, the best performing method, taking advantage of GPS. However, [18] observed that the reachability at sparser areas is unacceptable, even after clustering is applied first. *SRB* for low degree graphs seems to be low (compared to data obtained in this paper), for *RE* in the range 80-90 percent. We shall obtain, in this paper, reliable broadcasting methods with higher *SRB*.

Recently, Qayyum et al. [24] discussed in detail the multipoint relaying method first presented in [8] for efficient flooding in mobile wireless networks. The same simple technique was also recently exploited in [32] for ad hoc routing. Let us first define relay points for a given point (source or retransmitting node) *S*. A node is called "covered" if it received (directly or via retransmissions by other nodes) a message originating at *S*. Relay points of *S* are 1-hop neighbors of *S* that cover all 2-hop neighbors of *S*. That is, after all relay points of *S* retransmit the message, all 2-hop neighbors of *S* will receive it. The goal is to minimize the number of relay points of *S*. The computation of a multipoint relay set with minimal size is an NP-complete problem, as proven in [24]. Two practical heuristics for selecting the minimum number of forwarding neighbors (i.e., relay points) are given in [4]. The multipoint relay method [8], [24] is a heuristic that works as follows: Let $N1(S)$ and $N2(S)$ be sets of one-hop and two-hop neighbors of *S*, respectively. Further, let the relay points of *S* form the set $MPR(S)$. A small size set $MPR(S)$ is computed by the procedure *relays(S)*, which starts with an empty multipoint relay set $MPR(S)$, then selects first those one-hop neighbors of *S* which are the only neighbors of some node in

$N2(S)$, adds them to $MPR(S)$, and then enters the following loop:

While there still exist some node in $N2(S)$ which is not covered by $MPR(S)$ **do** {
 For each node in $N1(S)$ which is not in $MPR(S)$
 compute the number of nodes that it covers
 among the uncovered nodes in the set $N2(S)$.
 Add to $MPR(S)$ the node for which the number is
 maximal (ties are broken arbitrarily). }

Broadcasting from a node S is then performed as follows:

S broadcasts to all its neighbors, and computes $MPR(S)$.
 Initialize $retransmit(S)=MPR(S)$.

For each node X in $retransmit(S)$ **do** {

- X retransmits the message and computes $MPR(X)$ by calling procedure *relays*(X).
- Add nodes from $MPR(X)$ to the set $retransmit(S)$. }

Note that each node X in $retransmit(S)$ performs the computation (including retransmission) exactly once, by memorizing past traffic. Next, the list of nodes from $MPR(X)$ may be added to the message or, alternatively, each node X informs each of its neighbors Y whether or not Y is in the relay set of X . The later update is performed whenever node mobility causes topological changes in the network. In order to decide its set $MPR(X)$, node X needs to know the list of neighbors for each of its neighbors. In addition, node X shall react to any update from any neighbor by recomputing its relay set and informing all neighbors about their new status. Thus, each location update by node Y is followed by message updates by each neighbor X of Y , which is significant (but nevertheless local) overhead to the algorithm. On the other hand, the algorithm is reliable. It also has competitive performance in terms of reachability and saved message retransmissions (see Section 4).

Gerla et al. [11] proposed a combined clustering and broadcasting algorithm which has no communication overhead for neither maintaining cluster structure nor updating neighborhood information. In their passive clustering algorithm, the cluster structure is updated with existing traffic by adding two bits to each ongoing message. The source S of a broadcasting task will transmit the message to all its neighbors. S will declare itself a clusterhead (for the timeout period that is parameter in the method) if it has no neighboring active clusterhead. Upon receiving the message, each node A will declare itself a clusterhead using the same criterion as the source S . Otherwise, A will check the ratio of neighboring clusterheads and neighboring gateway nodes and declare itself a gateway if that ratio is above certain threshold, which is also a parameter of the method. If A decides to be gateway, it will retransmit the message. Otherwise, A decides to be an ordinary node and does not retransmit the message. The method is not reliable and has global parameters. Moreover, there are pathological cases of poor delivery ratio. The authors report only 35 percent savings in message retransmissions for flooding under certain network conditions.

Peng and Lu [22], independently and almost simultaneously with a conference version of this paper [28] suggested to use the neighbor elimination scheme to reduce the number of rebroadcasts. In their scheme, a node does not need to rebroadcast a message if all its neighbors have been covered by previous transmissions. Peng and Li [22] propose to let nodes with more neighbors rebroadcast earlier, so that more nodes can be covered by one transmission. They propose the following formula to calculate the rebroadcast delay. For node U , the delay time T_{max} is calculated as follows:

$$T0 = (1 + D_{max}(U))/(1 + D(U)),$$

$$T_{max} = random(\Delta * T0),$$

where $D_{max}(U)$ is the maximum degree of a neighbor of U (U is not taken into account), $D(U)$ is the degree of node U , and Δ is a small constant. $Random(x)$ is a function that returns a random integer in interval between 0 and x .

The special case of broadcasting task when all nodes are positioned on a straight line is studied in [1], [25] in the context of intervehicle communications on a highway. The problem of constructing an energy-efficient broadcast tree in a wireless network with nodes having adjustable transmission radii is studied in [13], [36], [35]. Huang and Huang [13] proved in 1989 that the problem of finding minimum total transmission power needed to broadcast a message from one node to all others in one-to-all model is NP-complete. A centralized algorithm (similar to Prim's shortest weighted path algorithm) for static networks which requires the geometric information such as positions or distances is proposed in [36] and analyzed in [35]. A number of centralized (where each node is assumed to know the full graph topology) broadcasting algorithms were proposed in literature. A recent survey of such algorithms, which mainly study worst-case performance and ignore constant multipliers in time complexities is given in [21]. We are interested here only in distributed algorithms. Also, our research and review are concentrated on the expected time complexity with favorable worst cases, that is, the exact latency or number of rounds needed for broadcasting a message.

2.2 Internal Nodes and Dominating Sets

Let G be the graph that corresponds to given wireless network. A set is dominating if all the nodes in G are either in the set or neighbors of nodes in the set. Nodes that belong to a dominating set will be called *internal* nodes for G (of course, a different definition for dominating set leads to different set of internal nodes). Routing based on a connected dominating set is frequently used approach [34], where the searching space for a route is reduced to corresponding internal nodes. It is desirable, in this context, to create dominating set with minimal possible ratio of internal nodes.

The clustering process is an example of creating a dominating set. The dominating set consists of all clusterhead nodes and border nodes, which are nodes which have neighbors from at least two clusters (that is, border nodes connect two CHs). The main drawback of using the cluster structure as the dominating set is its significant

communication overhead for maintaining the structure in a moving environment. Wu and Li [34] gave a literature review of several existing dominating set definitions which have significant overhead in maintaining the structure and do not produce better ratios of internal nodes than the simple definitions given in [34].

Wu and Li [34] proposed a simple and efficient distributed algorithm for calculating connected dominating set in ad hoc wireless networks. They introduced the concept of an *intermediate* node. A node A is an *intermediate* node if there exist two neighbors B and C of A that are not direct neighbors themselves. Wu and Li [34] also introduced two rules that considerably reduce the number of internal nodes in the network. Let $N(u)$ be the (open) set of all neighbors of node u and let $N[u] = N(u) \cup \{u\}$ be the corresponding closed neighbor set; that is, the set of all neighbors and u itself. Suppose that each node has a unique *id* number. Let us define *intergateway* nodes as intermediate nodes that are not eliminated by Rule 1. Next, let the *gateway* nodes be those intergateway nodes that are not eliminated by Rule 2.

Rule 1 [34] is as follows: Consider two intermediate nodes v and u . If $N[v] \subseteq N[u]$ in G and $id(v) < id(u)$, then node v is not an intergateway node. In other words, if any neighbor of v is also a neighbor of u and v is connected to u and has lower *id*, then any path via v can be replaced by a path via u and, thus, node v is not needed as an internal node. We may also say that node v is “covered” by node u . Rule 2 [34] is as follows: Assume that, after applying Rule 1, u and w are two intergateway neighbors of an intergateway node v . If $N(v) \subseteq N(u) \cup N(w)$ in G and

$$id(v) = \min \{id(v), id(u), id(w)\},$$

then node v is declared a nongateway node. In other words, if each neighbor of v is a neighbor of u or w , where u and w are two connected neighbors of v , then v can be eliminated from the list of gateway nodes.

If location information is available, each node can determine whether or not it is an intermediate, intergateway, or gateway node in $O(k^3)$ computation time (where k is the number of its neighbors) and without any message exchanged with its neighbors for that purpose. Otherwise, the maintenance of internal node status requires the knowledge of neighbors for each neighbors.

3 NEW BROADCASTING ALGORITHMS

3.1 Reducing Ratios of Internal Nodes

We propose replacing node *ids* with a record $key = (degree, x, y)$, where *degree* is the number of neighbors of a node and x and y are its two coordinates in the plane. In both rules from [34], nodes shall compare first their degrees and node with higher degree has greater chances of remaining an internal node. In case of ties, x -coordinate is used to resolve (or an *id* if GPS is not available). If x -coordinates are also the same, use y -coordinate for final decision. Such comparison rule will result in fewer remaining nodes in the graph. The example in Fig. 1 has used a defined record *key* instead of *ids*. The information about the degree of neighboring nodes may be gathered

together with information about their location. Let $d(u, v)$ denote the distance between nodes u and v . The procedure for determining the internal nodes status of any given node v can be described as follows:

```

Procedure internal-status( $v$ );
  intermediate( $v$ )=intergateway( $v$ )=gateway( $v$ )=false;
  for each neighbor  $u$  of  $v$  do
    for each neighbor  $w$  of  $v$  do
      if  $d(u, w) > R$  then intermediate( $v$ )=true;
  if intermediate( $v$ ) then { intergateway( $v$ )=true;
    for each neighbor  $u$  of  $v$  do {
      covered=true;
      for each neighbor  $w$  of  $v$  do
        if  $d(w, u) > R$  or  $key(w) > key(u)$  then
          covered=false;
      if covered then intergateway( $v$ )=false }
  if intergateway( $v$ ) then { gateway( $v$ )=true;
    for each neighbor  $u$  of  $v$  do
      for each neighbor  $w$  of  $v$  do {
        covered=true;
        for each neighbor  $z$  of  $v$  do
          if  $d(z, u) > R$  or  $d(z, w) > R$  or
              $key(z) > key(u)$  or
              $key(z) > key(w)$ 
            then covered=false;
        if covered then gateway( $v$ )=false } }
```

3.2 Broadcasting via Internal Nodes

We shall now propose new broadcasting algorithms for wireless networks. The communication overhead of broadcasting algorithms can be significantly reduced by applying the concept of internal nodes. Since any node in the network has a neighbor that is internal node, broadcasting task can be performed by retransmitting the message by each internal node to all its neighbors.

In one-to-all broadcasting, each internal node forwards the message to all its neighbors, which is counted as one message. The comparison with the flooding method (where each node forwards the message exactly once) is then simplified since it suffices to find the ratio of internal nodes in a network. Comparison with algorithm [23] is also simplified since the number of messages in algorithm [23] is equal to the number of clusterheads and border nodes in the clustered graph.

For example, Fig. 1 shows a unit graph with 12 nodes. Nine of these nodes (shaded and named ones) are intermediate, while five of them are intergateway and also gateway nodes. Thus, broadcasting based on the internal nodes concept will require nine, five, and five message transmissions, respectively, compared to 12 in case of flooding.

3.3 Improved Broadcasting by Neighbor Elimination

Fig. 2 illustrates a possible problem with all methods described in [18]. Node A received broadcast message from its only neighbor B . According to probabilistic, counter, distance, or location-based methods, A will rebroadcast the message although there is no other neighbor that needs to

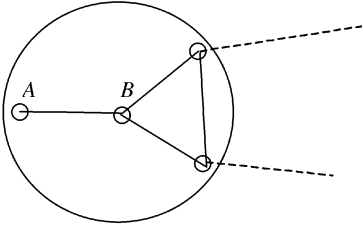


Fig. 2. Node A receiving message from its only neighbor B.

receive the message. Thus, a large additional coverage area may often be empty. Even lowest *id* clustering may treat A as clusterhead (with 50 percent chance), thus imposing rebroadcast at A. Note that A is not an internal node or a relay point for any other point; however this is not always the case.

We shall now propose an improvement for each of the discussed broadcasting algorithms. The improvement is based on the observation given in Fig. 2. A node will rebroadcast the message only if it has a neighbor that might need the message. Thus, some of neighbors are eliminated for rebroadcasting (they will, however, receive the message if it is sent because of other neighbors).

First, each node that is not supposed by the method to rebroadcast will assign itself to one of its possibly retransmitting neighbors. In a clustered structure, this neighbor is the corresponding clusterhead. In the internal node structure, we propose assigning the neighbor as follows: Each noninternal node A will assign itself to neighboring internal node B which has the largest degree. In case of ties, use lowest *id* among candidate neighbors. This rule attaches more neighbors to higher degree nodes thus possibly “emptying” the assigned list of low degree internal nodes.

Second, neighbors that received one of message copies that arrived at a node A are eliminated from the list of neighbors of A that might need the message. Consider, for example, node A in Fig. 3, which received twice the message which is being broadcast from neighbors B and C. Neighbors E and F are eliminated from the broadcast list since they received the same broadcast message from neighbors B and C, respectively. However, node A will, in this example, still rebroadcast the message because of neighbor G which is not “covered” by B or C. Nodes E, F, and G are either internal nodes (that is, belonging to dominating set) or noninternal nodes which are assigned to A. This scheme will further reduce the number of retransmissions in a broadcasting task. Its efficiency may also somewhat depend on the MAC scheme that is being used. The dominating set (using the most powerful gateway node concept) and neighbor elimination based broadcasting scheme can be described as follows: Each node *v* activates the following procedure *broadcast-receive(v,m,u)* upon receiving broadcast type message *m* from a neighboring node *u*.

Procedure *broadcast-receive(v,m,u)*;

If *gateway(v)* **then** {

If *m* received for the first time **then** {

Decide time slot to begin rebroadcasting;

Forward-neighbors(v) = set of all neighbors

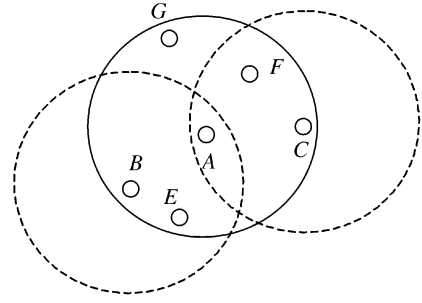


Fig. 3. Node A eliminates neighbors E and F from its broadcast list.

of *v* };

For each neighbor *w* **of** *v* **do**

If $d(u,w) < R$ **then** $\text{forward-neighbors}(v) =$
 $\text{forward-neighbors}(v) - w$;

Wait until time slot for rebroadcasting;

If $\text{forward-neighbors}(v)$ is nonempty and *m* not
already rebroadcasted by *v*

then rebroadcast the message;

Internal node and neighbor elimination schemes require each node to know the exact location of all its neighbors (if GPS or other location technique is applied) or to know the list of neighbors for each of its neighbor. Compared to the multipoint relay method [24], this significantly reduces the communication overhead in the presence of node mobility since the multipoint relay method requires that each neighbor of node that caused topological change to react by informing all of its neighbor about their new relay status. Thus, a reduction of about *d* times is achieved in the communication overhead, where *d* is average node degree in the network. In addition, the fixed choice of retransmitting internal nodes enables their use as virtual backbone for routing with reduced size of routing tables (only internal nodes need to be in routing tables). Multipoint relay method does not provide such a virtual backbone.

4 PERFORMANCE EVALUATION OF BROADCASTING ALGORITHMS

The experiments were carried out in several phases. In the first phase (Section 4.1), an ideal MAC protocol is assumed, which provides for collision-free broadcasting. The number of nodes that rebroadcast the message is counted, and compared to total number of nodes to evaluate the savings. The second phase (Section 4.2) involves a real simulation using a MAC protocol. The third phase includes the proposed neighbor elimination scheme, while the fourth phase adds negative acknowledgement to the scheme for almost guaranteed delivery of message to all nodes.

4.1 Internal Nodes vs. Cluster-Based and Multipoint Relay Broadcasting

The overhead of two clustering algorithms, *LowestID* and *ConID*, was compared in [5]. Table 1 gives the ratios of CH nodes and border (B) nodes and the combined CH+B ratios for random connected graphs with *n* = 100 nodes and degrees from 4 to 12. The random unit graphs were generated as follows: Each of *n* nodes is chosen

TABLE 1
Ratios of CHs and Border Nodes in *LowestID* and *ConID* Algorithms for 100 Nodes

d	<i>LowestID</i>		<i>ConID/LowestID</i> %			<i>ConID</i>	
	CH	B	CH+B	%	CH+B	CH	B
4	0.31	0.36	0.67	75	0.5	0.25	0.25
5	0.27	0.39	0.66	70	0.46	0.21	0.25
6	0.24	0.37	0.61	84	0.51	0.2	0.31
7	0.21	0.42	0.63	87	0.55	0.19	0.36
8	0.2	0.47	0.67	76	0.51	0.16	0.35
9	0.18	0.41	0.59	88	0.52	0.15	0.37
10	0.18	0.46	0.64	84	0.54	0.15	0.39
11	0.16	0.48	0.64	81	0.52	0.13	0.39
12	0.15	0.52	0.67	78	0.52	0.12	0.4

TABLE 2
Percentage of Broadcast Nodes for Each Method for $n = 100$ Nodes and $d = 4 - 10$

method/degree	4	5	6	7	8	9	10
<i>LowestID</i>	67	66	61	63	67	59	64
<i>ConID</i>	50	46	51	55	51	52	54
<i>Multipoint relay</i>	60	59	60	60	61	64	63
<i>intermediate</i>	80	84	88	91	92	94	95
<i>inter-gateway</i>	65	66	66	67	69	69	70
<i>gateway</i>	60	57	54	51	50	48	45

by selecting its x and y coordinates at random in the interval $[0,100)$. In order to control the average node degree d , we sort all $n(n-1)/2$ (potential) edges in the network by their length, in increasing order. The radius R that corresponds to chosen value of d is equal to the length of $nd/2$ th edge in the sorted order. Disconnected graphs were ignored.

The results indicate that *ConID* has consistently between 12 percent and 30 percent lower ratios compared to *LowestID*. The comparison was made also for $n = 50, 200, 500$, and 1,000 nodes and similar results were obtained (especially for CH+B ratios). The percentage of border nodes increases with degree, while the percentage of CH nodes decreases for both algorithms. The CH+B ratio appears to be relatively independent from degree d for each of the algorithms, around 65 percent for *LowestID* and 52 percent for *ConID*. The comparison was made for static networks only. The cluster structure of nodes changes due to node movements and existing maintenance algorithms do not preserve highest degree CH preference. Thus, the presented ratios for *ConID* appear to be optimal.

Table 2 compares clustering, multipoint relay, and internal node methods in terms of percentage of nodes that broadcast the message. Apparently, *intermediate* nodes concept does not achieve satisfactory gains. Gains for *intergateway* nodes are comparable to *LowestID* method. *Gateway* nodes method is comparable to *ConID*; more precisely, it is better for degrees over 6 and somewhat worse for lower degrees. Interestingly, the ratios appear to be relatively stable with respect to degree d , for most methods. More precisely, ratios are quite stable for clustering and *intergateway* methods, somewhat increasing (with increasing d) for *intermediate* method, and notably decreasing for *gateway* method. The multipoint relay method performs well, with savings between intergateway and gateway methods (note that papers [4], [24] did not

evaluate the performance of the multipoint method for solving broadcasting task on random unit graphs). Thus, the gateway method appears to be the overall “winner.” Recall that this comparison involves only retransmissions involved in direct broadcasting tasks and for initial clustering structure. Communication overhead involved in maintaining clustered or internal node structures is not included. As noted before, maintenance of clustered structure (in the presence of moving nodes) is a nontrivial operation and may involve a significant amount of message traffic. On the other hand, internal node structure requires communication with neighbors only when topology changes. Moving nodes pose an additional problem to clustered structure by loosening the highest degree CH property, which effectively moves *ConID* toward *LowestID* algorithm unless a global reclustering (which also means additional overhead) occurs. Therefore, internal nodes concept does seem to perform broadcasting task with significantly lower communication overhead compared to existing method based on clustering, even if clustering process is optimized (as proposed in this paper). Clustering-based broadcasting methods are, therefore, not evaluated further in this paper.

4.2 Internal Nodes vs. Location-Based Broadcasting

We have developed a simulator using C. A simplified version of the MAC specification in IEEE 802.11 standard is referenced to simulate carrier sense multiple access with collision avoidance (CSMA/CA) behavior among hosts. Since the speed of broadcasting a message is significantly larger than the node mobility, we assume that nodes are static while broadcasting is in progress. The experiments were, therefore, carried on random unit graphs, defined as follows: Each n nodes is chosen by selecting its x and y coordinates at random in the interval

TABLE 3
Reachability for Considered Methods

	1 x 1	3 x 3	5 x 5	7 x 7	9 x 9	11 x 11
Intermediate	100	100	100	97	96	98
Inter-gateway	100	100	99	96	96	98
Gateway	100	99	97	94	95	98
Neighbor elimination	100	100	99	97	96	98
Multipoint relay	100	100	99	96	96	99
Location 0.1871	100	100	97	90	72	72
Location 0.0913	100	100	98	96	87	79
Location 0.0469	99	99	99	100	88	81
Location 0.0251	98	99	99	98	90	83
Location 0.0134	97	99	99	99	94	83

TABLE 4
Saved Rebroadcasts for Internal Nodes and Location-Based Broadcasting

	1 x 1	3 x 3	5 x 5	7 x 7	9 x 9	11 x 11
Intermediate	1	1	5	14	22	32
Inter-gateway	97	32	29	34	37	43
Gateway	98	76	54	45	41	45
Neighbor elimination	57	11	14	22	30	45
Multipoint relay	99	45	39	42	46	54
Location 0.1871	90	78	52	37	30	22
Location 0.0913	78	55	36	26	17	12
Location 0.0469	67	42	26	17	12	8
Location 0.0251	61	36	21	12	8	5
Location 0.0134	56	32	28	10	8	6

$[0, m)$. We experimented with $n = 100$ nodes, as in [18]. The average node degree d , the transmission radius R , and the map size m are related to each other. In order to compare our results with those from [18] and to comply with IEEE 802.11 standard, we decided to fix radius R to 500 meters. The map sizes from [18] are equal to $s * R$, for $s = 1, 3, 5, 7, 9, 10$, and 11 . The corresponding average node degrees d are 96.5, 25.4, 10.4, 5.6, 3.5, 2.9, and 2.4, respectively.

It is assumed that one broadcasting task at a time is in the network and no other message traffic while broadcasting is in progress (this is a fair assumption for comparing various broadcasting methods). We used the same parameters as in [18]: The bit rate is 1M per second, the slot time is 20us (microseconds), and the packet size is 280 bits which, with required overhead, took 2,536us in their simulation. Global synchronization can be achieved by adding some dummy bits so that the transmission takes integer $p = 2540/20 = 127$ number of slots. Acknowledgements are not sent. In this protocol, when a node A receives a packet to be transmitted, it first waits for an interframe spacing $DIFS$ period ($DIFS = 2$ in our experiments). Node A then chooses a random integer BC (backoff counter) in interval $[0..31]$. The backoff counter determines the number of transmission-free slots as sensed by A . During periods in which the channel is clear, A decrements BC . When BC reaches 0, A transmits the packet. Once a node starts a transmission, it transmits continuously for p slots until packet is fully transmitted. Thus, a neighboring node receives the packet if it receives collision-free transmissions for the duration of p consecutive slots.

The performance metrics for comparison are [18]:

- Reachability (RE): the number of nodes receiving the broadcast message divided by total number of nodes that are reachable from source (the graph may not be connected);
- Saved ReBroadcast (SRB): $(r-t)/r$, where r is the number of nodes receiving the broadcast message, and t is the number of nodes that actually transmitted the message.

The location-based method is the best method among those presented in [18], according to their experimental data. We will therefore compare the proposed internal-node-based broadcasting and neighbor elimination methods only with the location-based and multipoint relay methods (see Table 3 and Table 4). The average number of reachable nodes from the source was 100, 100, 99.8, 85.7, 36.1, 17.2, and 11.3, respectively. We have listed numerical data from [18] in our tables.

Among internal-node-based methods, the gateway nodes seem to provide best combined values for RE and SRB . RE values for the location-based method are comparable to gateway nodes method for $s = 1, 3, 5$, and 7 , but are lower for $s = 9$ and 11 . SRB data are comparable to the gateway nodes method only for $A = 0.1871$ and only for $s = 3$ and 5 and are lower otherwise. Therefore, since a method is to be selected for arbitrary network density, the gateway-nodes-based broadcasting method is apparently better than any of methods proposed in [18] and has no parameter associated with it. This is especially valid for networks with average degrees of nodes under 10, which is the range that includes practically all throughput efficient networks. The neighbor elimination method has comparable RE to internal-node-based methods. However, it has

TABLE 5
SRB Data for (Inter)Gateway Method Enhanced by Neighbor Elimination Scheme

	1x1	3x3	5x5	7x7	9x9	10x10	11x11
Intergateway+neig.elim.	99	39	36	39	43	49	54
Gateway+neig.elim.	99	81	60	49	47	52	56

TABLE 6
SRB Data for RANA Algorithm

	1x1	3x3	5x5	7x7	9x9	10x10	11x11
Inter-gat.+neigh.elim.	99	28	26	34	41	47	53
Gateway+neigh.elim.	99	75	52	44	44	50	54
Neighbor elimination	57	1	4	16	27	36	43

lower SRB data than (inter)gateway node concept. The multipoint relay method, in this experiments, performed quite close to the gateway method.

The latencies of various methods were also compared. The gateway-based method, for instance, has lower latencies than any of location-based methods (for various parameter values). Differences were small for $A = 0.1871$ but were increasing with the increased value of threshold A .

4.3 Adding Neighbor Elimination Scheme

When both internal node status and neighbor elimination schemes are added, the algorithm works as follows: When an internal node receives a message, it retransmits the message if it has a noneliminated neighboring node which is either noninternal node assigned to it or an internal node.

In our next experiments, we added the neighbor elimination scheme to the proposed internal-node-based broadcasting schemes. The differences in reachability data were negligible (gateway node concept, for instance, had improvements up to 1 percent). The intermediate scheme benefited significantly from the elimination scheme in terms of SRB and latencies (more precisely, the data became similar to those obtained for neighbor elimination scheme alone), while other methods had improvements in 1-11 percent range.

Table 5 shows SRB data for (inter)gateway node concept enhanced by neighbor elimination scheme. The gateway method, with neighbor elimination scheme added, was shown to be the best overall method (in our experiments) before considering negative acknowledgements. The latencies were very close to those reported for the method, where nodes retransmit upon receiving negative acknowledgements.

4.4 Retransmission after Negative Acknowledgements

In order to approach 100 percent RE rate, we have designed RANA (Retransmission After Negative Acknowledgements) broadcasting algorithm. When message is retransmitted by a node A , if a collision at receiving node B occurs within the first two slots of receiving the message, no retransmission request is issued. We assume that two slots are sufficient to identify the sender A and message id . Thus, if collision occurs between the third and $p = 127$ th slot, the receiving node B can send a retransmit request to sending node A , which we assume is four slots long (it should suffice to

include requested message id and the identity of node A to whom request is made, to avoid retransmissions from several neighbors). Node A will retransmit the message only if it received, without collision, all four slots. The MAC protocol used is the same for both retransmit requests and message retransmissions. The retransmission request is canceled if, during the waiting period for sending retransmission request, node B receives the same message without collision from another neighbor.

The calculation of $SRB = (r - t)/r$ is modified to include all transmissions; more precisely, t is the total number of message retransmissions. The retransmission requests are not counted since they are much shorter in length than the message itself. The reachability RE for (inter)gateway-node-based broadcasting enhanced with neighbor elimination scheme has improved and was measured to be over 98.3 percent for all map sizes. Neighbor elimination scheme alone also performed well, with over 97.3 percent RE for all map sizes. SRB data has been reduced by up to 10 percent as a trade off for enhanced (inter)gateway methods, as indicated in Table 6, while neighbor elimination scheme did not give significant retransmission savings for half of map sizes.

Table 7 presents the measured latencies for (inter)gateway neighbor elimination RANA algorithms and for the location-based method with two extreme parameter values (latencies for other parameters are between indicated values). The latencies are expressed in terms of the number of p slot sequences; that is, the number of slots needed for the last node to receive the message divided by p . This measure is chosen in order to compare latency with message length, which is more illustrative than the time, which is proportional to indicated values. Table 7 shows that latencies for RANA algorithm (normalized to message transmission time) are still lower than latencies for location-based methods [18] and comparable to latencies for multipoint relay method [24].

4.5 Impact of Variable Maximal Backoff, Message Size, Traffic, and Mobility

Recall that the backoff counter BC is a random number from the interval $[1..Tmax]$. The main difference between neighbor elimination schemes independently proposed by us and in [22] is that $Tmax$ in [22] depends on the node density relative to its neighbors, while we proposed to use fixed value $Tmax = 31$. We compared these two $Tmax$ formulas using

TABLE 7
Latencies in Terms of Number of Message Lengths

	1x1	3x3	5x5	7x7	9x9	11x11
Int.gat.+neigh.elim. RANA	2.12	19.85	14.73	16.64	12.04	5.03
Gatew.+neigh.elim. RANA	2.21	9.18	12.20	15.85	11.66	4.97
Location A=0.1871	6.63	11.75	11.8	15.5	11.66	6.9
Location A=0.0134	25.8	20.2	16.1	18.4	16.9	8.0
Multipoint relay	1.7	18.6	14.5	16.7	11.7	5.0

both RANA algorithm and the previous version without negative acknowledgements. The difference in performance was negligible. The difference in *RE* values for $T_{max} = 31$ and T_{max} from [22] was in the range -0.4 to $+0.3$. *SRB* data did not change, while change in latency was up to -0.001 seconds.

We then searched for a variable T_{max} formula that will reflect the dynamic nature of node degree after eliminating some neighbors. If a node A decides to retransmit the message, it calculates a random number x in range $[0, 1)$. Then, the backoff counter is calculated as $BC = x * T_{max}$, where T_{max} depends on the number of noneliminated neighbors. We have tested the following two formulas for T_{max} : $T_{max} = 127/NNEN$ and $T_{max} = 20 + 31/NNEN$, where $NNEN$ is the number of internal and noninternal neighbors of A who have not been eliminated yet. Note that x is fixed while $NNEN$ (and consequently T_{max} and BC) are recalculated each time node A receives a new copy of the same message. As a result, BC may increase, resulting in a longer delay due to reduced number of noneliminated neighbors. The change in *RE* for both $T_{max} = 20 + 31/NNEN$ and $T_{max} = 127/NNEN$ was in the range 3.8 and $+0.5$, with no differences in *SRB* and latencies. Therefore, we were not able to find any improvements with variable T_{max} formulas.

The impact of message size p is then experimentally verified by repeating experiments for other sizes. In particular, measurements were done for $p = 64$ and $p = 254$ (that is, half the size and double the size). The impact of varying message size on *RE* and *SRB* data was negligible (except for values of p under T_{max} which reduces the probability of a collision). Latency was naturally affected by the message size.

All the measurements described so far assume that there is only one message to be broadcast at a given time. This assumption is reasonable for many applications, such as alarm signal. Moreover, if the transmission speed is high, in networks with a small and medium number of nodes, one can expect that current broadcasting will finish before the next one is initiated. In order to verify the impact of several concurrent broadcasting tasks in the network, we assumed that each node may initiate a new broadcasting task at the beginning of any time slot with a given small probability. The experimental data confirm the same differences for *RE* and *SRB* data. More precisely, the mutual differences are increasing in favor of better methods. The reason is obvious: Excess transmissions generated by weaker methods create even more collisions when more broadcasting tasks are in the network. That is, more unnecessary retransmissions cause more collisions.

The impact of node mobility is then verified. Each node moves, with a given probability, to a randomly chosen destination position on a straight line with a speed that is also chosen at random between two threshold values. We have implemented only location updates between neighboring nodes, using a method described in [30]. Each node stores the location of its neighbors, which suffices to apply almost all described methods. In addition, it stores the location of two-hop neighbors, using information obtained from neighbors (this information is also needed to apply multipoint relay method). In addition to location, each node also stores the speed and direction of movement of its neighbors (it is either part of a location update message or is calculated on the bases of two last reported neighbor locations). Each node decides to send a message containing its new location to all its neighbors whenever it detects that an existing edge will be broken or that a two-hop neighbor is becoming a direct neighbor. The distance from node A to a node B is measured by using estimated position B' for node B , calculated by using the last known position of B and its reported speed and direction of movement. This kind of location updates reduced significantly the number of such messages. For instance, two nodes moving with same speed and in the same direction do not need to repeatedly report to each other their new positions.

Because of locality of broadcasting procedures and efficiency of described location update technique, the measured differences in *RE* and *SRB* data from the case of static nodes were, in all cases, under 1 percent. The difference is mostly due to adding location update messages in the system and partially by occasional wrong information about neighbor positions. Thus, the proposed broadcasting algorithm perform equally well in case of moving nodes. Note that a similar conclusion is reported also by Peng and Lu [22] for neighbor elimination scheme. In their experiments, there was also about a 1 percent difference between the static scenario and the case where all nodes were in continuous movement.

5 CONCLUSIONS

Our experiments have demonstrated the efficiency of proposed broadcasting algorithms. The broadcasting reliability is achieved with significant reduction in the number of rebroadcasting messages, resulting in reduced contention and collision problems in the network. In addition, internal nodes concept has reduced the maintenance communication cost compared to clustered structure.

The efficiency of broadcasting appears to be directly related to the construction of a connected dominating set of

minimal size. Unfortunately, the problem is NP-complete [13], [10], [34], even for a centralized algorithm, consequently in distributed setting, and more so if a localized algorithm is designed, as in this paper. There are no known bounds on the ratio of dominating set size used in this paper [WS] to the size of minimal connected dominating set. Several centralized heuristics are proposed in [10] but even their performance is far from what is desirable in our case. Thus, constructing smaller dominating sets in localized manner is an open problem.

We believe that further savings and perhaps guaranteed delivery may be achieved by improving proposed algorithms in various ways. The main obstacle to achieving 100 percent success in reaching all nodes seems to be the case of nodes C whose only relevant neighbors are two nodes A and B that simultaneously receive the same broadcast message from a node S . If nodes A and B are not neighbors, their retransmissions will always collide with p/T_{max} ratios over 1. An ultimate method should resolve this problem and seems to be the main open problem left for further study.

One of problems associated with fixed selection of internal nodes for broadcasting is that only selected nodes will retransmit messages, which will reduce their battery power. We are currently designing new algorithms for selecting dominating sets [33], which will take node battery power into consideration. Nodes having more remaining power will be more likely to be in the dominating sets and the status may change at some threshold values. The change can therefore be triggered by change of power status in addition to node mobility.

Another issue in wireless networks is the presence of unidirectional links. Different transmission ranges of the mobile nodes or hidden terminal problem can cause unidirectional links. The performance of proposed scheme in the presence of unidirectional links is left for future study.

ACKNOWLEDGMENTS

The authors are grateful to Xiang-Yang Li, Wei Peng, Y.C. Tseng, Peng-Jun Wan, and Jie Wu for fruitful discussions on broadcasting issues and clarifying their papers. They would also like to thank Wei Peng, for pointing out the existing multipoint relaying method, and the referees, for their careful reading that has greatly improved the clarity of the presentation.

REFERENCES

- [1] L. Briesemeister and G. Hommel, "Role-Based Multicast in Highly Mobile but Sparsely Connected Ad Hoc Networks," *Proc. ACM Symp. Mobile Ad Hoc Networking and Computing (MobiHOC)*, pp. 45-50, Aug. 2000.
- [2] J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," *Proc. Conf. Mobile Computing, MOBICOM*, pp. 85-97, 1998.
- [3] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," *Proc. Third Int'l Workshop Discrete Algorithms and Methods for Mobile Computing and Comm. (DIAL M)*, pp. 48-55, Aug. 1999.
- [4] G. Calinescu, I. Mandoiu, P.J. Wan, and A. Zelikovsky, "Selecting Forwarding Neighbors in Wireless Ad Hoc Networks," *Proc. Fifth Int'l Workshop Discrete Algorithms and Methods for Mobile Computing and Comm. (DIAL M)*, 2001.
- [5] G. Chen and I. Stojmenovic, "Clustering and Routing in Wireless Ad Hoc Networks," Technical Report TR-99-05, Dept. of Computer Science, SITE, Univ. of Ottawa, June 1999.
- [6] S. Datta, I. Stojmenovic, and J. Wu, "Internal Nodes and Shortcut Based Routing with Guaranteed Delivery in Wireless Networks," *Cluster Computing*, to appear.
- [7] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," *Proc. Conf. Mobile Computing, MOBICOM*, pp. 263-270, 1999.
- [8] ITSI STC-RES10 Committee, "Radio Equipment and Systems: High Performance Radio Local Area Network Type 1, Functional Specifications," *journal?book?website?* June 1996.
- [9] A. Ephremides, J.E. Wieselthier, and D.J. Baker, "A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling," *Proc. IEEE*, vol. 75, pp. 56-73, 1987.
- [10] S. Guha and S. Khuller, "Approximation Algorithms for Connected Dominating Sets," *Algorithmica*, vol. 20, pp. 347-387, 1998.
- [11] M. Gerla, T.J. Kwon, and G. Pei, "On Demand Routing in Large Ad Hoc Wireless Networks with Passive Clustering," *Proc. IEEE Wireless Comm. and Network Conf.*, Sept. 2000.
- [12] M. Gerla and J.T.C. Tsai, "Multicasting, Mobile, Multimedia Radio Network," *Wireless Networks*, vol. 1, pp. 255-265, 1995.
- [13] N.F. Huang and T.H. Huang, "On the Complexity of Some Arborinesses Finding Problems on a Multihop Radio Network," *BIT*, vol. 29, pp. 212-216, 1989.
- [14] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath, "Flooding for Reliable Multicast in Multihop Ad Hoc Networks," *Proc. Third Int'l Workshop Discrete Algorithms and Methods for Mobile Computing and Comm. (DIAL M)*, Aug. 1999.
- [15] G. Lauer, "Address Servers in Hierarchical Networks," *Proc. Int'l Conf. Comm.*, pp. 443-451, 1988.
- [16] E.L. Lloyd, "Broadcast Scheduling for TDMA in Wireless Multi-Hop Networks," *Handbook of Wireless Networks and Mobile Computing*, to appear.
- [17] C.R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," *IEEE J. Selected Areas in Comm.*, vol. 15, no. 7, pp. 1265-1275, 1997.
- [18] S.Y. Ni, Y.C. Tseng, Y.S. Chen, and J.P. Sheu, "The Broadcast Storm Problem in a Mobile Ad Hoc Network," *Proc. Conf. Mobile Computing, MOBICOM*, pp. 151-162, Aug. 1999.
- [19] S.Y. Ni, Y.C. Tseng, and J.P. Sheu, "Efficient Broadcasting in a Mobile Ad Hoc Network," *Proc. IEEE Int'l Conf. Distributed Computing and Systems*, pp. 16-19, Apr. 2001.
- [20] A.K. Parekh, "Selecting Routers in Ad Hoc Wireless Networks," *Proc. SBT/IEEE Int'l Telecomm. Symp.*, 1994.
- [21] A. Pelc, "Broadcasting in Wireless Networks," *Handbook of Wireless Networks and Mobile Computing*, to appear.
- [22] W. Peng and X.-C. Lu, "On the Reduction of Broadcast Redundancy in Mobile Ad Hoc Networks," *Proc. First Ann. Workshop Mobile and Ad Hoc Networking and Computing*, pp. 129-130, Aug. 2000.
- [23] E. Pagani and G.P. Rossi, "Providing Reliable and Fault Tolerant Broadcast Delivery in Mobile Ad-Hoc Networks," *Mobile Networks and Applications*, vol. 4, pp. 175-192, 1999.
- [24] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint Relaying: An Efficient Technique for Flooding in Mobile Wireless Networks," Research Report RR-3898, INRIA, Mar. 2000. www.inria.fr/RRRT/RR-3898.html.
- [25] M.T. Sun, W.C. Feng, T.H. Lai, K. Yamada, and H. Okada, "GPS Based Message Broadcast for Adaptive Intervehicle Communications," *Proc. Int'l Conf. Parallel Processing*, Aug. 2000.
- [26] I. Stojmenovic and X. Lin, "Loop-Free Hybrid Single-Path/Flooding Routing Algorithms with Guaranteed Delivery for Wireless Networks," *IEEE Trans. Parallel and Distributed Systems*, to appear. Oct. 2001.
- [27] I. Stojmenovic and X. Lin, "Power-Aware Localized Routing in Wireless Networks," *IEEE Trans. Parallel and Distributed Systems*, to appear.
- [28] I. Stojmenovic and M. Seddigh, "Broadcasting Algorithms in Wireless Networks," *Proc. Int'l Conf. Advances in Infrastructure for Electronic Business, Science, and Education on the Internet SSGRR*, July/Aug. 2000.
- [29] M. Seddigh, J. Solano Gonzalez, and I. Stojmenovic, "RNG and Internal Node Based Broadcasting Algorithms for Wireless One-to-One Networks," *ACM Mobile Computing and Comm. Rev.*, vol. 5, no. 2, to appear.

- [30] I. Stojmenovic, "Location Updates for Efficient Routing in Wireless Networks," *Handbook of Wireless Networks and Mobile Computing*, to appear.
- [31] I. Stojmenovic, M. Russell, and B. Vukobratovic, "Depth First Search and Location Based Localized Routing and QoS Routing in Wireless Networks," *IEEE Int'l Conf. Parallel Processing*, pp. 173-180, Aug. 2000.
- [32] P. Sinha, R. Sivakumar, and B. Vaduvur, "Enhancing Ad Hoc Routing with Dynamic Virtual Infrastructures," *IEEE Joint Conf. Computer and Comm. Societies (INFOCOM)*, 2001.
- [33] J. Wu, M. Gao, and I. Stojmenovic, "On Calculating Power-Aware Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Networks," *IEEE Int'l Conf. Parallel Processing*, Sept. 2001.
- [34] J. Wu and H. Li, "A Dominating Set Based Routing Scheme in Ad Hoc Wireless Networks," *Proc. Third Int'l Workshop Discrete Algorithms and Methods for Mobile Computing and Comm. (DIAL M)*, pp. 7-14, Aug. 1999.
- [35] P.J. Wan, G. Calinescu, X.Y. Li, and O. Frieder, "Minimum-Energy Broadcast Routing in Static Ad Hoc Wireless Networks," *Proc. 20th Ann. Joint Conf. IEEE Computer and Comm. Soc., INFOCOM*, 2001.
- [36] J.E. Wieselthier, G.D. Nguyen, and A. Ephremides, "On the Construction of Energy-Efficient Broadcast and Multicast Trees in Wireless Networks," *Proc. 19th Ann. Joint Conf. IEEE Computer and Comm. Soc., INFOCOM*, Mar. 2000.



Ivan Stojmenovic received the BS, MS, and PhD mathematics degrees in 1979, 1983, and 1985, respectively, in Yugoslavia. He earned a third degree prize at International Mathematics Olympiad for high school students in 1976. In 1980, he joined the Institute of Mathematics, University of Novi Sad (Yugoslavia). In Fall 1988, he joined the University of Ottawa (Canada), where he is currently a full professor in SITE. Since June 2000, he is frequently in Mexico City, Mexico, as a researcher in DISCA, IIMAS, Universidad Nacional Autonoma de Mexico. He has published more than 150 papers in journals and conferences and edited *Wireless Networks and Mobile Computing Handbook* (Wiley, 2001). His research interests include wireless networks, parallel computing, multiple-valued logic, evolutionary computing, neural networks, combinatorial algorithms, computational geometry, graphs, and computer science education. He is currently a managing editor of *Multiple-Valued Logic*, an international journal, and an editor of *Parallel Processing Letters*, *IASTED International Journal of Parallel and Distributed Systems*, *Discrete Mathematics and Theoretical Computer Science*, *Parallel Algorithms and Applications*, and *Tangenta*.



Mahtab Seddigh received the BSc degree in business computing from the University of Winnipeg, Canada and the Master's degree in computer science from the University of Ottawa, Canada. She is working as a software designer in Network Management with Nortel Networks since 1997. Before that, she was a systems analyst with Statistics Canada for two years.



Jovisa Zunic received the MSc and PhD degrees in mathematics and computer science from the University of Novi Sad, Yugoslavia, in 1989 and 1991, respectively. He worked as a professor and researcher at the University of Novi Sad, Yugoslavia, for more than a decade and is currently a senior researcher in the Department of Computer Science at Cardiff University in the United Kingdom. He is also with the Mathematical Institute of the Serbian

Academy of Sciences and Arts. His research interests are in digital image analysis, digital geometry, shape representation and coding of digital objects, discrete mathematics, neural networks, wireless networks, and combinatorial optimization.

► **For more information on this or any computing topic, please visit our Digital Library at <http://computer.org/publications/dlib>.**