

# A Generic Distributed Broadcast Scheme in Ad Hoc Wireless Networks \*

Jie Wu and Fei Dai  
Department of Computer Science and Engineering  
Florida Atlantic University  
Boca Raton, FL 33431  
Email: {jie,fdai}@cse.fau.edu

## Abstract

We propose a generic framework for distributed broadcasting in ad hoc wireless networks. The approach is based on selecting a small subset of hosts (also called nodes) to form a forward node set to carry out a broadcast process. The status of each node, forwarding or non-forwarding, is determined either by the node itself or by other nodes. Node status can be determined at different snapshots of network state along time (called views) without causing problems in broadcast coverage. Therefore, the forward node set can be constructed and maintained through either a proactive process (i.e., “up-to-date”) before the broadcast process or a reactive process (i.e., “on-the-fly”) during the broadcast process. A sufficient condition, called *coverage condition*, is given for a node to take the non-forward status. Such a condition can be easily checked locally around the node. Several existing broadcast algorithms can be viewed as special cases of the generic framework with  $k$ -hop neighborhood information. A comprehensive comparison among existing algorithms is conducted. Simulation results show that new algorithms, which are more efficient than existing ones, can be derived from the generic framework.

**Keywords:** *Ad hoc wireless networks, broadcasting, distributed algorithms, pruning.*

---

\*This work was supported in part by NSF grant CCR 9900646 and grant ANI 0073736.

# 1 Introduction

Ad hoc wireless networks (or simply ad hoc networks) are dynamic in nature. Due to this dynamic nature, global information/infrastructure such as link state and routing table, which are obtained through global information exchanges, are no longer suitable to support routing in ad hoc networks. Broadcasting is a special routing process of transmitting a packet so that each node in a network receives a copy of this packet. *Flooding* is a simple approach to broadcasting with no use of global information/infrastructure; in flooding, a broadcast packet is forwarded by every node in the network exactly once. Simple flooding ensures the coverage; the broadcast packet is guaranteed to be received by every node in the network, providing there is no packet loss caused by collision in the MAC layer and there is no high speed movement of nodes during the broadcast process. Figure 1 (a) shows a network with three nodes. When node  $v$  broadcasts a packet as shown in Figure 1 (b), both nodes  $u$  and  $w$  receive the packet due to the broadcast nature of wireless communication media.  $u$  and  $w$  will then forward the packet to each other. Apparently, the last two transmissions are unnecessary. Redundant transmissions may cause the *broadcast storm problem* [21], in which redundant packets cause contention and collision.

Both deterministic and probabilistic approaches can be used to find a forward node set. The probabilistic approach [7, 21] normally offers a simple scheme in which each node, upon receiving a broadcast packet, forwards the packet with probability  $p$ . The value  $p$  is determined by relevant information gathered at each node. However, the probabilistic approach cannot guarantee full coverage. In the deterministic approach, the forward node set can be selected statically (independent of any broadcast process) [3, 5, 16, 25] or dynamically (during a particular broadcast process) [10, 12, 15, 19, 20]. The forward node set also forms a *connected dominating set* (CDS). A dominating set is a subset of nodes in the network where every node is either in the subset or a neighbor of a node in the subset.

Many distributed broadcast algorithms [3, 5, 10, 12, 15, 16, 19, 20, 25] with no use of global information/infrastructure have been proposed for use in ad hoc networks. Different assumptions and models have been used. So far, no generic framework can capture a large body of distributed broadcast algorithms; this makes the comparison among them difficult. It has been proved that the task of finding the smallest set of forward nodes with global network information/infrastructure is NP-complete. The problem is even more challenging in the absence of global network information/infrastructure. Heuristic methods are normally used to balance cost (in collecting network information and in decision making) and effectiveness (in deriving a small forward node set).

In this paper, we provide a generic framework that covers deterministic distributed broadcast schemes in ad hoc networks. In this framework, the status of each node, forward or non-forward, is determined locally based on  $k$ -hop neighborhood information (for  $k = 2$  or  $3$ ). The broadcast packet can carry a small amount of broadcast state information such as recently visited nodes. Broadcast algorithms based on global network

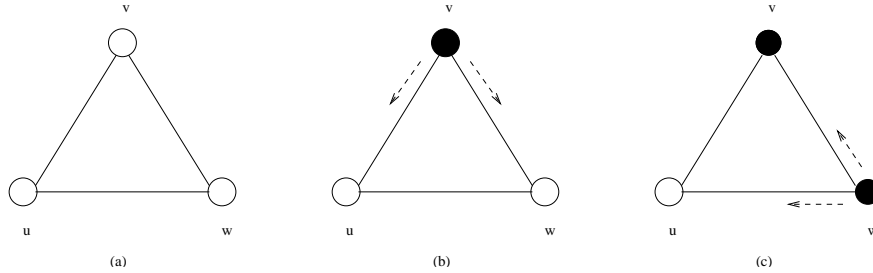


Figure 1: A sample ad hoc network with three different views.

topology [6] or pseudo-global network topology (that exhibits “sequentialized propagation”) [2, 4] do not provide scalability and are not included for further discussion. A comprehensive classification of broadcast schemes in ad hoc networks can be found in [26]. Under the objective of selecting a small CDS, the status of each node is decided in a decentralized manner based on a particular *view*, which is a snapshot of network state, including network topology and broadcast state, along time. Views can be sampled at different times and they can be *local views* that include connectivity and broadcast state of only nodes in the vicinity. In the generic framework, the status of a node can be decided by itself or by other nodes (say neighbors). Each node has the forwarding status by default like in flooding, and the status can be changed to non-forwarding if the proposed sufficient condition, called *coverage condition*, is met. In addition, such a condition can be easily checked locally around the node. Several existing broadcast algorithms can be viewed as special cases of the generic framework under local views with 2- or 3-hop neighborhood information. A comprehensive comparison among existing algorithms under the generic framework is conducted. Simulation results show that new algorithms under different local views, which are more efficient than existing ones, can be derived from the generic framework.

Although the efficiency of the coverage condition in producing a small CDS is confirmed by various simulation results, this scheme does not guarantee a constant approximation ratio in the worst case. Due to the special topological property of the ad hoc networks, constant approximation ratio can be achieved based on location information [9], global information [8], or global infrastructure (e.g., a spanning tree [22] or clusters [26]). The basic idea is to partition an ad hoc network into several regions, each region occupies exclusively a certain amount of geographical area, and select a constant number of nodes from each region to form a CDS. However, the importance of approximation ratio should not be over emphasized. For example, although the greedy algorithm proposed by Guha and Khuller [6] does not have a constant approximation ratio, it performs much better than several approaches with constant ratios on randomly generated networks. Another consideration is that the network partition process takes  $O(n)$  rounds in the worst case, which limits the scalability of these schemes. Except for location-based schemes, no approach can achieve both a constant approximation ratio and constant round of information exchanges. On the other hand, the coverage

condition can also be applied to a CDS generated by one the above schemes to further reduce the number of forward nodes.

The remainder of this paper is organized as follows: Section 2 gives preliminaries and our system model based on the notion of view. Section 3 proposes a generic sufficient condition called the coverage condition for the non-forward node. Discussions on the use of the coverage condition are given in Section 4. Section 5 provides a generic distributed broadcast protocol based on the coverage condition. Some of the existing broadcast protocols as special cases of the generic distributed broadcast protocol are given in Section 6. Simulation results are presented in Section 7. Section 8 concludes the paper.

## 2 Preliminaries

For a particular broadcasting, the status of each node can be determined before the broadcast process starts (called the *static approach*) or during the broadcast process (called the *dynamic approach*). The static approach is independent of the broadcast state information. It is also called *proactive*, since the status of each node is computed periodically as the network topology changes. The dynamic approach depends on both network topology and broadcast state information. It is also called *reactive* (or “on-the-fly”) as the status of each node is computed for each broadcast process.

In a formal term, an ad hoc network is represented by a unit disk graph  $G(t) = (V, E)$ , where two vertices (nodes) are connected if their geographical distance is within a given transmission range  $r$ . Note that  $G(t)$  is a function on time  $t$ . A *global view* with respect to a particular broadcast process is a snapshot of network topology and broadcast state. More formally,  $View(t) = (G(t), Pr(V, t))$ , where  $Pr(V, t)$  is a priority vector of nodes in  $V$  at time  $t$ . The status of each node is determined based on a particular view  $View(t)$ . We assume the network topology does not change during the broadcast period, so  $G(t)$  can be simply represented as  $G$ . The priority of each node  $v \in V$ ,  $Pr(v, t)$ , is a tuple  $(S(v, t), id(v))$ , where  $S(v, t)$  represents the forwarding status of  $v$  under  $View(t)$ , and  $id(v)$  is the distinct identifier of node  $v$  (other parameters such as node degree can be used in place of node id).

A node that has forwarded or has been designated to forward (i.e., determined under a previous view) is called a *visited node*; otherwise, it is an *un-visited node*.  $S(v, t) = 2$  is reserved for a visited node  $v$  and  $S(v, t) = 1$  for an un-visited node (i.e., a visited node has a higher priority than an un-visited node under the lexicographical order). In this case,  $Pr(v, t)$  is a monotonically increasing function along the time. In the subsequent discussion,  $t$  is omitted with an understanding that all terms are with respect to a particular view. An un-visited node is called a *forward node* if it is or would be determined to forward the broadcast packet under the current view; otherwise, it is called a *non-forward node*. Both visited/un-visited and forward/non-forward status are time sensitive. The forward node status under the current view will be

a visited node in the next view. The forward node status of each node can be changed; however, once the forwarding status has been distributed to other nodes (i.e., it is reflected in the next view and other nodes have learned the visited node status of this node), it cannot be changed.

Figure 1 shows an example of view changes during a broadcast process initiated from  $v$ , where the network topology remains unchanged and visited nodes are colored black. The priority vectors are the following:  $Pr(V) = (Pr(u), Pr(v), Pr(w)) = ((1, u), (1, v), (1, w))$  for Figure 1 (a),  $Pr(V) = ((1, u), (2, v), (1, w))$  for Figure 1 (b) and  $Pr(V) = ((1, u), (2, v), (2, w))$  for Figure 1 (c). The lexicographical order can be used to order nodes based on their priorities; e.g.,  $(1, w) > (1, v)$  and  $(2, v) > (1, w)$ .

In ad hoc networks, a *local view* at node  $v$  is a more realistic model to determine the node status of  $v$ . A view is local at node  $v$  if node  $v$  can only capture part of a view within its vicinity. Specifically, a local view,  $View' = (G', Pr(V'))$ , of  $View = (G, Pr(V))$  meets the following conditions:  $G'$  is a subgraph of  $G$  and  $Pr'(V) \leq Pr(V)$ ; that is, each element  $Pr'(v)$  is no more than the one in  $Pr(v)$ .  $Pr'(v)$  is defined as follows:  $Pr'(v) = Pr(v)$  if  $v \in V'$ ; otherwise,  $Pr'(v) = (S(v) = 0, id(v))$  (i.e., an invisible node under the local view has the lowest priority).

A *static view* is a view without any visited node. Applying the static approach (i.e., with static views only) to the example in Figure 1 (a), any node can be selected to form a forward node set. Node priority (i.e., node id) can be used to break a tie. Suppose  $w$  (the highest id among the three) is selected. When the source is  $w$ ,  $w$  alone forms a forward node set. When the source is  $v$ ,  $v$  and  $w$  form a forward node set. In the dynamic approach, not only topology information but also the distribution of visited nodes are used to select forward nodes.

Throughout the paper, it is assumed that each node only captures a local view, which is a subgraph of the original graph, and its priority vector is no more than that of the global view (i.e., an un-visited node will not be treated as a visited node). Note that any visited nodes are assumed to be connected under any local view, since they are all connected to the source. Five additional assumptions are used: (1) There is no error in packet transmission; that is, each message (broadcast packet or network state message) sent from a node will eventually reach its neighbors. (2) Location information of each node is not available. Location-based broadcasting has been extensively studied as in [14, 18, 19]. (3) Network topology is a connected graph without unidirectional links. A sublayer can be added [17, 23] to provide a bidirectional abstraction for unidirectional ad hoc networks. (4) All nodes have fresh topology information in their local views at the beginning of the broadcast period, and the network topology does not change during the broadcast period. Note that if the network topology changes during the broadcast period, no broadcast algorithm (including flooding) can ensure full coverage. Although some existing reliable broadcast protocols [1, 13] can be applied to guarantee full coverage through transmission redundancy and confirmation, they are beyond the scope of this paper. (5) The network is relatively sparse. For a dense ad hoc network, the *clustering approach* [11, 26] can be used to convert the dense graph to a sparse one.

### 3 The Generic Coverage Condition

In the generic distributed broadcast protocol, each node has the forwarding status by default like in flooding. However, the status of a node can be non-forwarding if the following sufficient condition, called *coverage condition*, is met. We start with the coverage condition where the status of all nodes are determined under one single view. The result is then extended to the case where the status of each node is determined under a distinct local view.

---

#### Coverage Condition:

Node  $v$  has a non-forwarding status if for any two neighbors  $u$  and  $w$ , a *replacement path* exists that connects  $u$  and  $w$  via several intermediate nodes (if any) with higher priorities than that of  $v$ .

---

The coverage condition indicates that when every pair of neighbors of  $v$  can be connected through nodes with higher priorities, node  $v$ , as the connecting node for its neighbors, can be replaced (i.e., can take the non-forwarding status). A replacement path may include some visited nodes that have the highest priorities. Note that “replacement” can be applied iteratively. To avoid possible “cyclic dependency” situations, a total order is defined among nodes based on  $Pr(v)$ . Intermediate nodes may not exist. In this case,  $u$  and  $w$  are directly connected. In a formal term, assume that  $v$  is a non-forward node. Let  $N(v)$  be the neighbor set of node  $v$ , then for any  $u, w \in N(v)$ , a replacement path  $(u, u_1, u_2, \dots, u_l, w)$  exists such that  $Pr(u_i) > Pr(v)$ . Next we define a special replacement path, called a *maximal replacement path*, such that all intermediate nodes (if any) are either forward nodes or visited nodes. That is, none of the nodes in the maximal replacement path can be replaced under the current view.

**Definition 1** Max-min node for  $(u, w, v)$ : A min node in a path is a node with the lowest priority. Assume  $\{P_i\}$  is the set of paths connecting  $u$  and  $w$  and each node in a path in the set has a higher priority than that of  $v$ . A max-min node in  $\{P_i\}$  is a node with the highest priority among all min nodes in  $\{P_i\}$ .

Next we define a procedure called MAXMIN to construct a maximal replacement path for  $v$  connecting  $u$  and  $w$ .

MAXMIN( $u, w, v$ ):

- 1: **if**  $u$  and  $w$  are directly connected **then return**  $\emptyset$ .
- 2: Find the max-min node  $x$  for  $(u, w, v)$ .
- 3: **return** path (MAXMIN( $u, x, v$ ),  $x$ , MAXMIN( $x, w, v$ )).

**Lemma 1** The procedure MAXMIN( $u, w, v$ ) will complete in finite steps and generate a maximal replacement path.

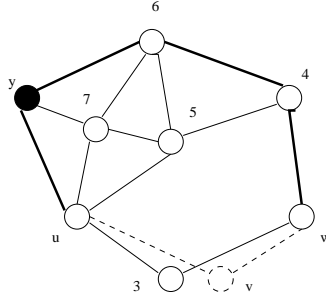


Figure 2: A maximal replacement path for  $v$  connecting  $u$  and  $w$ .

**Proof:** First, we show that all nodes generated by  $\text{MAXMIN}(u, w, v)$  are distinct. Based on the min node definition,  $x$  has the lowest priority among all nodes in replacement paths connecting  $u$  and  $x$  (and ones connecting  $x$  and  $w$ ). Therefore,  $x$  will not be selected as the max-min node in either  $\text{MAXMIN}(u, x, v)$  or  $\text{MAXMIN}(x, w, v)$ . To show that  $\text{MAXMIN}(u, x, v)$  and  $\text{MAXMIN}(x, w, v)$  have no common element, we assume that  $\text{MAXMIN}(u, x, v) = u_1, u_2, \dots, u_l$  and  $\text{MAXMIN}(x, w, v) = x_1, x_2, \dots, x_m$ . Suppose  $u_i = x_j$ , then  $(u, u_1, \dots, u_i, x_{j+1}, \dots, x_m, w)$  is a replacement path for  $v$  connecting  $u$  and  $w$ . The fact that all the nodes in this path have a higher priority than  $x$  contradicts the fact that  $x$  is a max-min node. Since each recursive call of the max-min procedure selects a distinct node, this process will complete in finite steps.

Next we show that  $x$  cannot be further replaced (i.e.,  $x$  will be a forward node under the current view if it is not a visited node). If  $x$  is replaced by path  $P$ , then  $(\text{MAXMIN}(u, x, v), P, \text{MAXMIN}(x, w, v))$  is another replacement path for  $v$  that connects  $u$  and  $w$  (if it is a walk with multiple occurrences of some nodes, multiple occurrences can be easily removed to form a path). Clearly, all the nodes in this path have higher priorities than  $x$  which contradicts the fact that  $x$  is a max-min node.  $\square$

Figure 2 shows a sample maximal replacement path constructed from the max-min procedure by including  $u$  and  $v$  at the two ends. In this example,  $id(v) = 2$ . Nodes with priorities lower than the one of  $v$  are not shown. Node 4 is the max-min node for  $(u, w, v)$ . Node 6 is the max-min node for  $(u, 4, v)$ , and visited node  $y$  is the max-min node for  $(u, 6, v)$ . Therefore, the maximal replacement path is  $(u, y, 6, 4, w)$ .

**Theorem 1** *Given a graph  $G = (V, E)$  that is connected but not a complete graph, the forward node set  $V'$  (including forward nodes and visited nodes), derived based on the coverage condition, forms a connected dominating set of  $G$ .*

**Proof:** We first show that  $V'$  forms a dominating set. Randomly select a node  $v \in V$ . We show that  $v$  is either in  $V'$  or adjacent to a node in  $V'$ . If  $v$  is a visited node or a neighbor of a visited node, the theorem holds; otherwise, if  $v$  is a forward node under the current view, the theorem also holds. For the remaining case, we will show that there exists a forward neighbor. Since  $v$  is a non-forward node, for any two neighbors

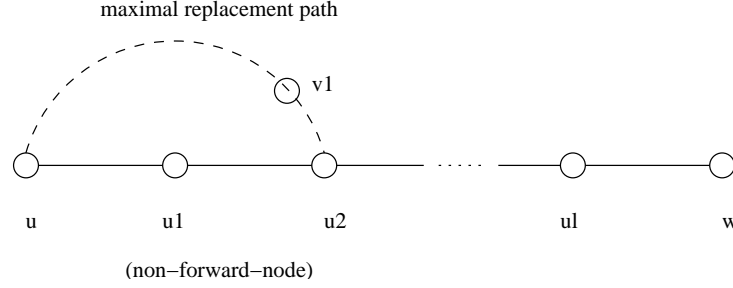


Figure 3: Maximal replacement path for  $u_l$ .

of  $v$ , there is a replacement path for  $v$  connecting these two neighbors. There exists at least one neighbor  $u$  of  $v$  such that there is  $w \in N(u)$ , but  $w \notin N(v) \cup \{v\}$  (otherwise,  $G$  is a complete graph). Let  $u$  be such a neighbor with the largest id. Clearly, there is no replacement path for  $u$  connecting  $v$  and  $w$  and, hence,  $u$  is a forward node.

Next we show that  $V'$  is connected. Randomly select two nodes  $u$  and  $w$  in  $V'$ . Assume that  $(u, u_1, u_2, \dots, u_l, w)$  is a path in  $G$  connecting  $u$  and  $w$ . Find a maximal replacement path for  $u_l$  that connects  $u$  to  $u_2$ . Assume that  $v_1$  is the last intermediate node of the maximal replacement path. Note that  $v_1 = u_1$  if  $u_1$  is not replaceable. Repeat the above process on  $(u_1, u_2, \dots, u_l, w)$  to replace  $u_2$  (see Figure 3). Eventually,  $u_1, u_2, \dots$ , and  $u_l$  are all replaced or skipped and the resultant path connects  $u$  and  $w$  with forward nodes and visited nodes only (if it is a walk with multiple occurrences of some nodes, multiple occurrences can be easily removed to form a path).  $\square$

Note that when the network is a complete graph, there is no need of a forward node. One transmission from the source reaches all the nodes. Theorem 1 shows the result under one particular view, i.e., each node takes the same view in deciding its status. Suppose each node  $v_i \in V$  decides its status under a distinct local view,  $View_i$ . The following theorem shows that Theorem 1 still holds.

**Theorem 2** *If each node  $v_i$  applies the coverage condition under a local view,  $View_i$ , Theorem 1 still holds.*

**Proof:** Let  $f_i(v_i)$  be a Boolean variable representing the forwarding status of node  $v_i$  under  $View_i = (G(v_i) = (V(v_i), E(v_i)), Pr_i(V))$ : 1 for forwarding and 0 for non-forwarding. Each  $G(v_i)$  is a subgraph of  $G$ .  $F = (f_1(v_1), f_2(v_2), \dots, f_n(v_n))$  captures the forwarding status of all nodes in the network under their corresponding local views. Define  $Views_{super} = (G_{super}, Pr_{super}(V))$ , where

$$G_{super} = (V_{super}, E_{super}) = (V(v_1) \cup V(v_2) \cup \dots \cup V(v_n), E(v_1) \cup E(v_2) \cup \dots \cup E(v_n))$$

and

$$Pr_{super}(v_i) = \max\{Pr_1(v_i), Pr_2(v_i), \dots, Pr_n(v_i)\}$$



Note that each  $Pr_l(v_i)$  has three potential forms/values: the priority of an invisible node  $(0, id(q))$ , the lowest priority; the priority of an un-visited node  $(1, id(q))$ ; and the priority of a visited node  $(2, id(v_i))$ , the highest priority. Because  $Pr_i(V) \leq Pr_{super}(V)$  and  $G_i$  is a subgraph of  $G_{super}$ ,  $f_{super}(v_i) \leq f_i(v_i)$  based on the coverage condition. Therefore,

$$F_{super} = (f_{super}(v_1), f_{super}(v_2), \dots, f_{super}(v_n)) \leq F = (f_1(v_1), f_2(v_2), \dots, f_n(v_n))$$

Applying Theorem 1 to  $Views_{super}$ , the forward node set under  $F_{super}$  forms a connected dominating set. Clearly, the forward node set under  $F$  also forms a connected dominating set.  $\square$

A node that takes the forward status under a global view must also take the same status under a local view, but not vice versa. Therefore, the forward node set under different local views is a superset of the one under the global collective view of all local views.

## 4 Discussion

We elaborate more on the coverage condition based on four important aspects of its application: (1) timing issue, (2) selection issue, (3) space issue, and (4) priority issue. To simplify the discussion, under a particular view, each visited node is colored black (called a black node) and all the other nodes are colored white. Note that a forward node is colored white until its status is visible to other nodes (i.e., after a view change).

### 4.1 Timing issue

A broadcast protocol is called *static* if the forward/non-forward status of each node is determined on the *static view* (i.e., with no visited node) only; otherwise, it is *dynamic*. The static broadcast protocol is a special case of the dynamic one. The difference is that the forward node set derived from static views can be used in any broadcasting while the one derived from dynamic views is normally used in a specific broadcasting. There are two types of dynamic algorithms: (1) *First-receipt*: the status is determined right after the first receipt of the broadcast packet. (2) *First-receipt-with-backoff*: the status is determined after a *backoff delay* of the first receipt of the broadcast packet. A backoff delay is used so that a node can learn more about the broadcast state from its forward neighbors. However, this is done at the cost of prolonging the completion time of the broadcast process.

Figure 4 shows two examples of forward node set on the same network: one without broadcast state based on the static version of the coverage condition (Figure 4 (a)) and one with broadcast state based on the dynamic version of the coverage condition (Figure 4 (b)). In the example with broadcast state, it is assumed that the up-stream broadcast state is piggybacked with the broadcast packet. The forward node set derived from Figure 4 (a) can also be interpreted as the one for any broadcasting. In Figure 4 (b), because nodes

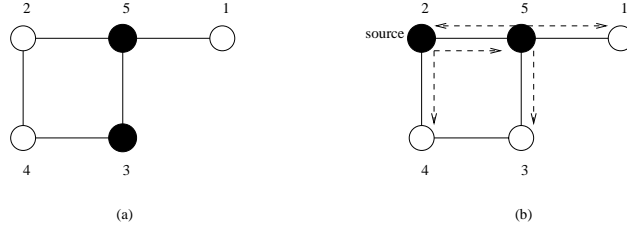


Figure 4: (a) Forward node set without broadcast state (static). (b) Forward node set with upstream broadcast state (dynamic) with node 2 being the source (visited node).

2 (source) and 5 are visited nodes, node 3 can conclude that it can be a non-forward node since two of its neighbors can be connected using node 2 (a black node). Note that if the status of node 3 is decided (as a forward node) before the broadcast process starts at node 2 or before it learns the broadcast state, it can still be changed to the non-forwarding status as long as it has not sent out its status (i.e., no other node has used the status of node 2 in its decision).

## 4.2 Selection issue

The coverage condition only states the condition under which a node  $v$  can be labelled non-forwarding. The selection issue deals with who should check this condition (and hence determine the status) for  $v$ . There are three choices: (1) *Self-pruning*. The status of  $v$  is determined by node  $v$  itself. (2) *Neighbor-designating*. The status of  $v$  is determined by some other nodes, say neighbors of  $v$ . (3) *Hybrid*. The status of  $v$  is determined by both  $v$  and neighbors of  $v$ .

In the self-pruning approach, each node  $v$  determines its status using the coverage condition. In the neighbor-designating approach, each node  $v$  determines the status of all its neighbors. In case a node has multiple status as selected by its neighbors, it will forward once (and only once) if at least one status is forwarding. This requirement, however, can be further relaxed. We can redefine the status function  $S(v, t)$  in the priority tuple  $Pr(v, t) = (S(v, t), id(v))$  as follows:  $S(v, t) = 2$  for visited node  $v$ ,  $S(v, t) = 1.5$  for an unvisited but designated node, and  $S(v, t) = 1$  for an unvisited and undesignated node. A designated node does not need to forward the packet if it meets the coverage condition. In the hybrid approach, each node determines the status of some of its neighbors and leaves other neighbors to determine their own status.

## 4.3 Space issue

A view consists of network topology and broadcast state information (visited status of some nodes). Network topology information is relatively long lived and can be collected through periodic “hello” messages

exchanged among neighbors. In an ad hoc network, it is too expensive to collect global network topology. A local view of network topology, in terms of *k-hop neighborhood information* (or simply *k-hop information*), is used as an approximation. The notion of *k-hop information* is often used liberally in literature, and its meaning varies in different circumstances. To simplify the discussion, we give a definition as follows.

**Definition 2** *Given a node  $v$ , its local view of network topology  $G_k(v)$  is said to contain  $k$ -hop information, if it takes at least  $k$  rounds of neighborhood information exchanges to build up.*

If the neighborhood information is collected via periodically exchanging “hello” messages, it takes  $k$  rounds for each node to collect its  $k$ -hop information. It is clearly impossible to collect up-to-date network topology information for large  $k$ ; therefore,  $k$  is usually a small integer such as 2 or 3 in ad hoc networks.

The maximum subgraph  $G_k(v)$  that can be derived from  $k$ -hop information is  $(N_k(v), E_k(v))$ , where  $N_k(v)$  denotes the set of nodes that are at most  $k$  hops away from node  $v$ . That is,  $N_0(v) = \{v\}$  and  $N_{k+1}(v) = (\bigcup_{u \in N_k(v)} N(u)) \cup N_k(v)$ , for  $k \geq 0$ .  $E_k(v)$  denotes the set of links between  $N_k(v)$  in  $G$ , excluding those links between  $k$ -hop neighbors. That is,  $E_k(v) = E \cap (N_{k-1}(v) \times N_k(v))$ . For example, if  $v$  has 1-hop information, then it knows all its neighbors, but not the links between these neighbors.

Broadcast state information is relatively short lived and cannot be collected through relatively infrequent “hello” messages. Instead, such information can be collected through the following two means: (1) *Snooped*. Each node can snoop the activities of its neighbors. When a neighbor forwards the broadcast packet, it becomes a visited node. (2) *Piggybacked*. When a node forwards the broadcast packet, it also attaches information of some visited nodes (including designated forward neighbors). There are two ways that a forward node  $v$  piggybacks visited node information: (a)  $v$  piggybacks information of visited nodes passed from its predecessor and (b)  $v$  piggybacks information about designated forward neighbors that have been selected by  $v$ ; that is, neighbors that will forward the packet. We normally assume that network topology information is not piggybacked, since the broadcast packet needs to be kept relatively small.

#### 4.4 Priority issue

The priority function used in the coverage condition can also affect the resulting forward node set. Based on the difficulty in collecting the priority values, the node properties that are used in the priority function can be divided into three categories:

**0-hop priority.** Node id,  $id(v)$ , represents a distinct identification of node  $v$  as used now in  $Pr(v)$ . Node id can be obtained without neighborhood topology information. Id’s of nodes in  $N_k(v)$  can be collected together with  $N_k(v)$  and no extra round of “hello” message exchange is needed. Therefore, it is the least expensive, but it is also the least efficient one in reducing the forward node set.

**1-hop priority.** Node degree,  $deg(v)$ , is defined as the number of  $v$ 's neighbors, i.e.,  $|N(v)|$  (or simply  $|N(v)|$ ). The higher the node degree of a node, the higher the priority. Node degree is based on 1-hop information. If  $k$ -hop information is collected together with node id, an extra round of information exchange is required before neighborhood information converges. Therefore,  $k$ -hop information plus node degree for each node in  $N_k(v)$  requires  $(k + 1)$ -hop information. Node degree is more efficient in reducing the forward node set, but it is also more expensive than 0-hop priority. When  $deg(v) = deg(u)$ , the id's of  $u$  and  $v$  are usually used to break a tie.

**2-hop priority.** Neighborhood connectivity ratio,  $ncr(v)$ , is the ratio of pairs of neighbors that are not directly connected to pairs of any neighbors. That is,

$$ncr(v) = 1 - \frac{\sum_{u \in N(v)} |N(u) \cap N(v)|}{deg(v)(deg(v) - 1)}$$

Again, the higher the  $ncr(v)$  value of node  $v$ , the higher the priority. Using  $ncr(v)$  as the priority value is the most efficient in reducing the forward node set, but it is also the most expensive. Collecting  $ncr(u)$  for each node  $u$  in  $N_k(v)$  requires  $(k + 2)$ -hop information. When  $ncr(v) = ncr(u)$ , node degrees followed by node id's of  $u$  and  $v$  are usually used to break a tie.

In neighbor-designating schemes, each designated node has one or more *designators* (i.e., nodes that request it to forward the broadcast packet). Attributes of these designators can be used as priority functions. Specifically, the minimal id of a node's designators is a 2-hop priority, which can be collected in a proactive manner. The transmission time of a node's first designator with respect to a specific broadcast packet is a 0-hop priority, which can be obtained without additional cost in a reactive manner.

## 5 A Generic Distributed Broadcast Scheme

Here we propose a generic distributed broadcast scheme based on the coverage condition. This is a dynamic approach in which a connected forward node set is constructed for a particular broadcast request, and it is dependent on the *location of the source and the progress of the broadcast process*. We assume that each node  $v$  determines its status and the status of some of its neighbors "on-the-fly" under a local view. The source node always forwards the packet. The approach can also be used in a static view where a connected forward node set is constructed independent of any particular broadcast process. We also assume that the broadcast packet that arrives at  $v$  carries information of  $h$  most recently visited nodes,  $v_1, v_2, \dots, v_h$ , and the set of designated forward neighbors,  $D(v_i)$ , selected at each  $v_i$  (usually for small  $h$  such as 1 or 2). Figure 5 shows a view at  $v$  with regard to the broadcast state information, assuming each node applied the first-receipt approach and, hence, each node has only one upstream link (with respect to the source). A general case is that each node has more than one upstream link (i.e., the node forwards the broadcast packet after receiving several copies of the packet) and a *reverse forwarding tree* is formed with root  $v$ .

---

**Algorithm 1** A Generic Distributed Broadcast Protocol (for each node  $v$ )

---

- 1: Periodically  $v$  exchanges “hello” messages with neighbors to update local network topology  $G_{\#}(v)$ .
  - 2:  $v$  updates priority information  $Pr$  based on snooped/piggybacked messages.
  - 3:  $v$  applies the coverage condition to determine its status.
  - 4: **If**  $v$  is a non-forward node **then stop**.
  - 5:  $v$  designates some neighbors as forward nodes if needed and updates its priority information  $Pr$ .
  - 6:  $v$  forwards the packet together with  $Pr$ .
- 

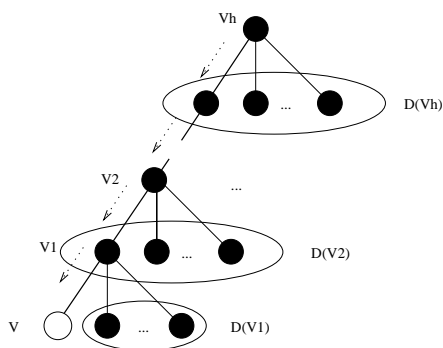


Figure 5: Broadcast state that includes visited nodes and their designated forward neighbors.

In Algorithm 1, the first two steps collect information to establish a local view. Step 1 collects network topology information of the view while Step 2 collects broadcast state information of the view. In Step 3, each node  $v$  determines its status based on the coverage condition. The process stops at Step 4 if  $v$  is a non-forward node; otherwise, the view is enhanced by selecting some forward neighbors at Step 5. In addition,  $v$  is changed to a visited node. Finally,  $v$  forwards the packet at Step 6. The default status for each node is forward, but non-designated forward node.

## 6 Special Cases

A large body of existing broadcast protocols can be considered as special cases of our generic distributed broadcast protocol. These special cases take one or more of the following approaches: (1) By skipping some of the steps in the scheme. (2) By using some special cases of the coverage condition at Step 3. (3) By applying a specific strategy in selecting designated forward nodes at Step 5.

One commonly used special case of the coverage condition uses a *coverage set*. A set  $C(v)$  is called a coverage set of  $v$  if the neighbor set can be “covered” by nodes in  $C(v)$ ; that is,  $N(v)$  is a subset of the union of neighbor sets of nodes in  $C(v)$ .

---

**Strong Coverage Condition:**

Node  $v$  has a non-forwarding status if it has a coverage set. In addition, the coverage set belongs to a connected component of the subgraph induced from nodes with higher priorities than that of  $v$ .

---

Clearly, the strong coverage condition is stronger than the original coverage condition, because the existence of a connected coverage set implies the existence of a replacement path for any two neighbors. In general, the original coverage condition is more costly to check than the strong coverage condition. When the original coverage condition is applied on  $k$ -hop neighborhood information with a constant  $k$ , its computation complexity is  $O(D^3)$ , where  $D$  is the density of network (i.e., the maximum number of nodes per unit area). While under the same circumstance, the computation complexity of the strong coverage condition is  $O(D^2)$  [24]. Obviously, the overhead is higher with larger  $D$ . Although appropriate density is necessary for network connectivity, a very dense network is inefficient in a shared media access scheme, because each node needs to contend with  $O(D)$  neighbors for the limited bandwidth. We assume that high density can be avoided by techniques such as adjustable transmitter range or clustering [11, 26], and therefore, the computation cost can be kept reasonably small.

The basic idea in selecting designated forward neighbors is that by designating some forward neighbors, other neighbors can take the non-forwarding status. Designated forward neighbors should be those covering at least one 2-hop neighbor of the current node (otherwise, they will not contribute in coverage). One extreme is to select a minimum number of designated forward neighbors so that other neighbors can take the non-forwarding status.

In the following, we examine several existing approaches as special cases of our generic distributed broadcast protocol. Special cases are grouped into static and dynamic. Within dynamic approaches, they are further classified as self-pruning, neighbor designating and hybrid. Some special cases of the coverage condition, that do not appear in any of the existing algorithms, are also discussed.

## 6.1 Static algorithms

The typical static algorithms are the generic distributed broadcast protocol with Steps 1 and 3.

**Wu and Li's algorithm.** Wu and Li [25] proposed a *marking process* to determine a set of *gateways* (i.e., forward nodes) that form a CDS: a node is marked as a gateway if it has two neighbors that are not directly connected. Two pruning rules are used to reduce the size of the resultant CDS. Based on pruning Rule 1, a gateway can become a non-gateway if all of its neighbors are also neighbors of another node, called *coverage node*, that has a higher priority. According to pruning Rule 2, a gateway can become a non-gateway if all of its neighbors are also neighbors of either of two coverage nodes that are directly connected and have

higher priorities. Two types of priority are used: node id and the combination of node degree and node id. In order to implement the marking process and pruning rules, 2-hop information (if each coverage node is a neighbor) or 3-hop information (if one coverage node is a neighbor's neighbor) is collected at each node.

**Dai and Wu's algorithm.** Dai and Wu [5] extended the previous algorithm by using a more generic pruning rule called Rule  $k$ : a gateway becomes a non-gateway if all of its neighbors are also neighbors of any one of  $k$  coverage nodes that are self-connected and have higher priorities. Rules 1 and 2 are special cases of Rule  $k$ , where  $k$  is restricted to 1 and 2, respectively. The connectivity requirement of coverage nodes requires information about nodes beyond 3 hops. However, Rule  $k$  can be implemented in a restricted way, which is as efficient as Rule 1 and more efficient than Rule 2, using either 2- or 3-hop information.

**Span.** Chen et al [3] proposed an approach, called Span, to construct a set of forward nodes called *coordinators*. A node  $v$  becomes a coordinator if it has two neighbors that are not directly connected, indirectly connected via one intermediate coordinator, or indirectly connected via two intermediate coordinators. Before a node changes its status from non-coordinator to coordinator, it waits for a backoff delay which is computed from its energy level, node degree, and neighborhood connectivity ratio. The backoff delay can be viewed as a priority value, such that nodes with shorter backoff delay period have a higher chance of becoming coordinators. Span cannot ensure full coverage, because two coordinators may simultaneously change back to non-coordinators and the remaining coordinators may not form a CDS. To conduct a fair comparison of Span and other broadcast algorithms, we use in this paper an enhanced version of Span, where a node becomes a coordinator if it has two neighbors that are not directly connected or indirectly connected via one or two intermediate coordinators with higher priority values. 3-hop information is needed to implement Span.

Both Wu and Li's algorithm and Dai and Wu's algorithm use the strong coverage condition and each coverage set consists of nodes with higher priorities. In Span, the coverage condition is used with two restrictions: no visited node is used and each replacement path is no more than three hops. Figure 6 (a) shows an example of the difference between the coverage condition and the strong coverage condition. Node 4 is a non-forward node under the coverage condition but is a forward node under the strong coverage condition. With local views, node 4 is non-forward under 3-hop information and forward under 2-hop information since the link (7,8) is invisible to node 4.

## 6.2 Dynamic and self-pruning algorithms

The dynamic and self-pruning algorithms are usually the generic distributed broadcast protocol with Steps 1, 2, 3, and 6.

**SBA.** Peng and Lu [15] proposed the Scalable Broadcast Algorithm (SBA) to reduce the number of forward nodes. Unlike the static algorithms, the status of a forward node is computed on-the-fly. When a node  $v$

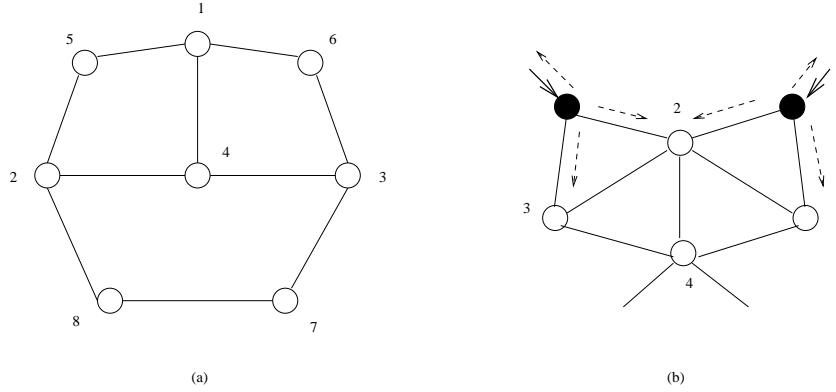


Figure 6: Two sample ad hoc networks.

receives a broadcast packet, instead of forwarding it immediately,  $v$  will wait for a backoff delay. For each neighbor  $u$  that has forwarded the packet, node  $v$  removes  $N(u)$  from  $N(v)$ . If  $N(v)$  does not become empty after the backoff delay, node  $v$  forwards the packet; otherwise, node  $v$  becomes a non-forward node. 2-hop information is used to implement SBA.

**Stojmenovic's algorithm.** Stojmenovic et al [19] extended the Wu and Li's algorithm in two ways: (1) Suppose every node knows its accurate geographic position, only 1-hop information is needed to implement the marking process and Rules 1 and 2. That is, each node only maintains a list of its neighbors and their geographic positions (connections among neighbors can be derived). (2) The number of forward nodes is further reduced by a neighbor elimination algorithm similar to the one used in SBA.

**LENWB.** Sucec and Marsic [20] proposed the Lightweight and Efficient Network-Wide Broadcast (LENWB) protocol, which also computes the forward node status on-the-fly. However, unlike in SBA and Stojmenovic's algorithm, the forward node status is determined when the broadcast packet is received for the first time. Whenever node  $v$  receives a broadcast packet from a neighbor  $u$ , it computes the set  $C$  of nodes that are connected to  $u$  via nodes that have higher priorities than  $v$ . If  $N(v)$  is contained in  $C$ , node  $v$  is a non-forward node; otherwise, it is a forward node. Similar to Dai and Wu's Rule  $k$ , the connectivity requirement also needs information about nodes beyond 3 hops; however, a restricted implementation can be done using 2- or 3-hop information.

Peng and Lu's and Stojmenovic's approaches are similar and use a special case of the strong coverage condition, where all coverage nodes are neighbors. That is, if each neighbor is either a visited node or a neighbor of a visited node, the corresponding node is a non-forward node. Therefore, the neighbor set is covered by a set of (connected) visited nodes. The first-receipt-with-backoff approach is used. Sucec and Marsic's approach also uses the strong coverage condition with a coverage set consisting of one visited node (black node) and the rest un-visited but higher priority nodes. In this approach, the first-receipt approach is



adopted.

Figure 6 (b) shows a case of neighbor coverage that cannot be covered by Peng and Lu's and Stojmenovic's approaches. After node 2 has two visited neighbors, neighbor 4 is still not covered based on both Peng and Lu's and Stojmenovic's approaches. However, using the strong coverage condition, node 2 is a non-forward node, because its neighbor set is covered by white nodes 3, 4 and two black nodes. Note that the two black nodes are viewed as connected in node 2's local view.

### 6.3 Dynamic and neighbor designating algorithms

The typical dynamic and neighbor designating algorithms are the generic distributed broadcast protocol with Steps 1, 2, 4, 5, and 6. All of the following approaches adopt the greedy strategy where a minimum set of designated forward nodes is selected so that the other neighbors can take the non-forward status.

**Dominant pruning.** Lim and Kim [10] provided two broadcast algorithms. One of them is based on simple self-pruning, which can be viewed as the first-receipt version of SBA. The other one is based on dominant pruning (DP). The DP algorithm uses 2-hop information to compute the forward node set of each node. Specifically, if  $u$  is the last forward node and  $v$  is designated as the next forward node,  $v$  selects its local forward node set from  $X = N(v) - N(u)$  to cover 2-hop neighbors in  $Y = N_2(v) - N(u) - N(v)$ . The local forward node set is selected with a simple greedy algorithm as in the set coverage problem. That is, each node  $w$  in  $X$  calculates its effective node degree  $deg_e(w) = |N(w) \cap Y|$ . A node  $w_1$  with the maximum  $deg_e(w_1)$  is first selected,  $w_1$  is removed from  $X$  and  $N(w_1)$  is removed from  $Y$ . If  $Y$  is not empty, each node re-computes its effective node degree and another node  $w_2$  with the maximum  $deg_e(w_2)$  is selected. This procedure is repeated until  $Y$  becomes empty and  $\{w_1, w_2, \dots\}$  forms a local forward node set that covers  $N_2(v)$ .

**Multipoint relays.** Qayyum et al [16] proposed selecting multipoint relays (MPRs) as forward nodes to propagate link state messages in their optimized link state routing (OLSR) protocol. The MPRs are selected from 1-hop neighbors to cover 2-hop neighbors, with a greedy algorithm similar to the one used in DP. Visited nodes are not considered in the selection of MPRs and, therefore, the entire set of 2-hop neighbors must be covered. MPR can be viewed as a static version of DP and is maintained in a proactive manner. The difference is that a relaxed neighbor-designating requirement is applied to MPR. If an MPR receives a broadcast packet first from a neighbor that is not its designator, it does not need to forward this packet because its neighbor set is covered by MPRs designated by that neighbor. Here the transmission time serves as a priority function to avoid cyclic dependence.

**Lou and Wu's algorithm.** Lou and Wu [12] extended the DP algorithm by further reducing the number of 2-hop neighbors to be covered by 1-hop neighbors. Two algorithms, total dominant pruning (TDP) and partial dominant pruning (PDP), are proposed. TDP requires the last forward node  $u$  piggyback  $N(u)$  along

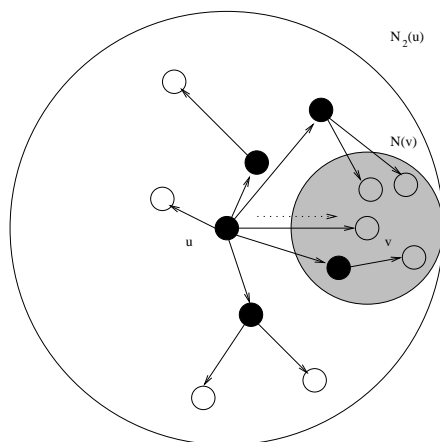


Figure 7: Neighbor-designating algorithms.

with the broadcast packet. With this information, the next forward node  $v$  can remove  $N_2(u)$ , instead of  $N(u)$  in DP, from  $N_2(v)$ . PDP, without using the piggybacking technique, directly extracts the neighbors of the common neighbors of  $u$  and  $v$  (i.e., neighbors of nodes in  $N(u) \cap N(v)$ ) from  $N_2(v)$ . Simulation results show that PDP avoids the extra cost in TDP introduced by piggybacking 2-hop information with the broadcast packet, but achieves almost the same performance improvement.

Both dominant pruning and MPR are based on using some 1-hop neighbors, as designated forward neighbors, to cover all 2-hop neighbors so that all remaining 1-hop neighbors can be non-forwarding. MPR uses the static approach while the other algorithms use the dynamic first-receipt approach. Lou and Wu's algorithm can be considered as DP with a better greedy algorithm in selecting the coverage set. Figure 7 shows how these approaches use the strong coverage condition: Suppose  $u$  is the current node and node  $v$  is any neighbor that is not selected as a forward neighbor. Since  $u$  and the selected designated forward neighbors cover all the 2-hop neighbors of  $u$  which include 1-hop neighbors of  $v$ , node  $v$  is covered by a set of (connected) visited nodes.

#### 6.4 Dynamic and hybrid algorithms

We consider here a hybrid of self-pruning and neighbor-designating algorithms. The first-receipt approach is still used. Upon receiving a broadcast packet from  $u$  with designated forward node set  $D(u)$  selected by  $u$ ,  $v$  uses the following steps: If  $v$  is not a designated forward node and  $v$  has not sent the packet before, then  $v$  applies the coverage condition to determine its status. If  $v$  is a forward node (self selected or designated),  $v$  selects a neighbor  $w \notin u \cup D(u)$  as its designated forward node (if any) based on a certain priority scheme. A neighbor that covers some nodes in  $N_2(v)$  is selected with either the lowest id or the maximum effective

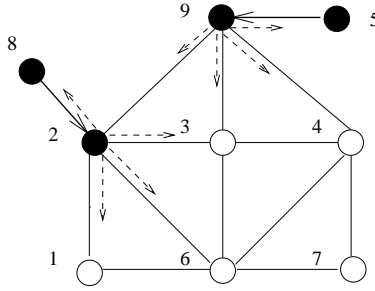


Figure 8: A sample ad hoc network for different selection policies.

node degree (with respect to uncovered nodes in  $\mathcal{N}_2(v)$ ). Node id is used to break a tie in node degree. Then  $v$  forwards the packet together with  $D(v) = \{w\}$ . Note that the selected forward neighbor should cover at least one 2-hop neighbor. In this hybrid approach, each node  $v$  only uses 2-hop information.

Consider the example in Figure 8 and suppose nodes 2 and 9 are forwarding the packet to its neighbors. Using self-pruning, nodes 4 and 6 will be forward nodes and nodes 1 and 3 will be non-forward nodes based on the coverage condition with 2-hop information. It is assumed nodes 1 and 6 receive their first copy of the packet from node 2 and node 4 from node 9. Node 3 receives its first copy of the packet from either node 2 or node 9. Using the proposed hybrid approach with node degree as the priority, node 2 is selected as a designated forward node by node 9 and node 6 by node 2. Note that nodes 2 and 9 do not know each other's forwarding status and, hence, there is no coordination in selecting their designated forward nodes. Node 4 is no longer a forward node, since nodes 2 and 9 are visited nodes under the local view of node 4 (passed from node 9). If node id is used as the priority in the hybrid approach, node 2 is selected by node 9. Node 3 is selected by node 2, since node 1 does not cover any 2-hop neighbor of node 2. Once node 3 receives the packet, it will pick node 4 to cover node 7. Using the neighbor-designating approach, node 9 selects node 2 first followed by node 4 to complete the 2-hop coverage. Similarly, node 2 selects node 6 and then node 9.

## 7 Simulation

A comprehensive simulation study has been conducted to answer two questions related to the generic distributed broadcast protocol: (1) Among different implementation options, is there a best combination in terms of cost-effectiveness? (2) Can new algorithms be derived from the generic scheme that outperform the existing ones and if so, what is the degree of performance increase?

The simulation program consists of an ad hoc network topology generator and a discrete event broadcasting simulator. To generate a random ad hoc network,  $n$  hosts are randomly placed in a restricted  $100 \times 100$  area. The transmitter range is adjusted according to a given average node degree  $d$  to produce exactly  $\frac{nd}{2}$

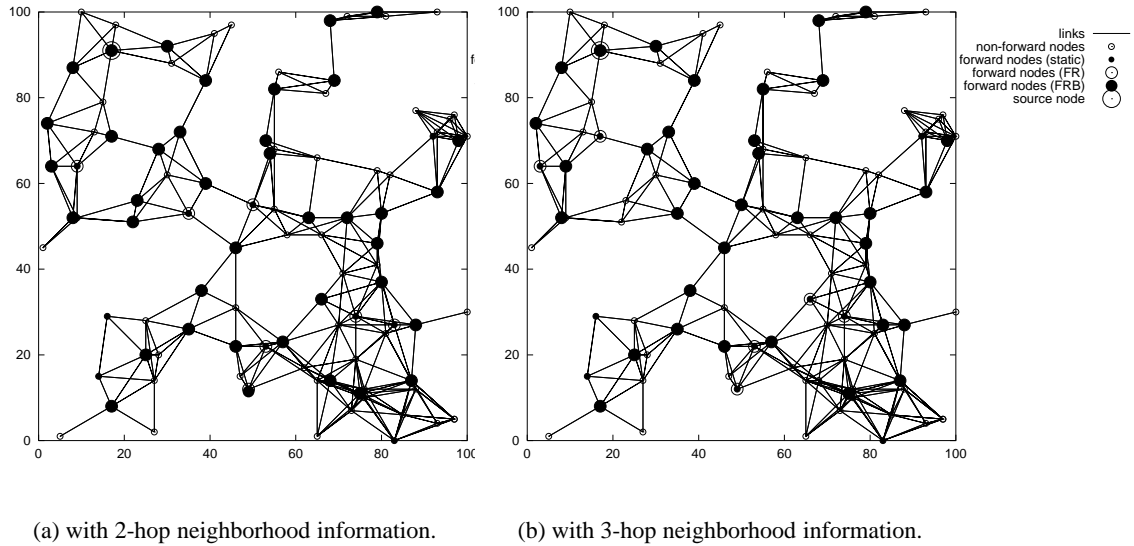


Figure 9: Broadcasting on a sample ad hoc network of 100 nodes. The numbers of forward nodes are (a) 49, 45, 41 and (b) 46, 42, 36 for the static, first-receipt, and first-receipt-with-backoff algorithms, respectively.

links in the corresponding unit disk graph. Two average node degrees are used in the simulation, one for relatively sparse networks ( $d = 6$ ) and another for relatively dense networks ( $d = 18$ ). Networks that are not connected are discarded. Performances of the special cases of the generic distributed broadcast protocol are compared in terms of the average number of the forward nodes under different network configurations. For each configuration, the simulation is repeated until the 90% confidence interval of the average value is within  $\pm 1\%$ . Figure 9 shows several forward node sets derived from several broadcast algorithms on a sample ad hoc network.

## 7.1 Implementation options

Many existing broadcast algorithms use 2-hop information and node id as priority, because they have the lowest cost in view update. Unless otherwise specified, they are used in the following simulations.

**Timing issue.** Figure 10 compares performances of different broadcast algorithms, in terms of the size of the forward node set, where each node makes the forward/non-forward decision proactively (Static), after the first receipt of the broadcast packet (FR), after the first receipt with a random backoff delay (FRB), or after the first receipt with a backoff delay that is proportional to the inverse of node degree (FRBD). The static algorithm requires less computation and no extra end-to-end delay, but also produces more forward nodes. The FR algorithm causes no extra end-to-end delay, but recomputes the forwarding/non-forwarding status

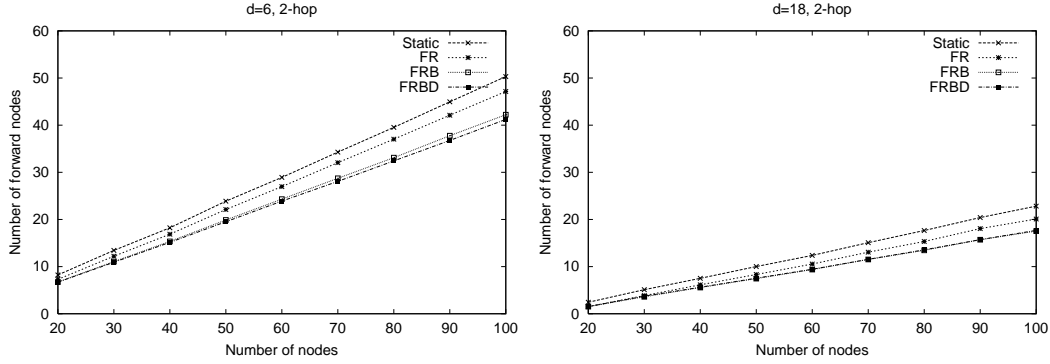


Figure 10: Performances of broadcast algorithms with different timing options. The average node degree is 6 (left) and 18 (right).

for each broadcasting. FR produces fewer forward nodes than the static algorithm. The two algorithms with backoff delays recompute node status for each broadcasting and cause extra end-to-end delay. They produce the smallest forward node set, and between them, FRBD is slightly better than FRB.

Since the computation time is negligible in a broadcast process, the dynamic algorithm is more desirable than the static one. Among the dynamic algorithms, FR is appropriate for highly delay-sensitive applications, and FRBD is appropriate for less delay-sensitive applications.

**Selection issue.** Figure 11 compares performances of different algorithms, where the coverage condition is applied via self-pruning (SP), neighbor-designating (ND), and two hybrid schemes: one that designates a neighbor with the highest degree (MaxDeg) and the other that designates a neighbor with the lowest id (MinPri). In the neighbor-designating and hybrid schemes, we use the strict rule that every designated node becomes forward node. In relatively sparse networks, the sequence from the worst performance to the best performance is MinPri, ND, SP and MaxDeg. MinPri is the worst, which suggests that designating a neighbor with the lowest priority produces more redundancy than the expected elimination effect. Performances of ND, SP and MaxDeg stay close. MaxDeg is slightly better, because it designates some nodes with large degrees and small id's, which can be used in replacement paths of nodes that have larger id's and are originally hard to replace. In relatively dense networks, when the number of nodes is small ( $n \leq 50$ ), ND, MinPri and MaxPri stay close and perform better than SP. This is because when the network diameter is small, most broadcast processes complete in 2 hops. In this case, a “centralized” selecting algorithm as used in ND is more effective than the “decentralized” algorithm used in SP. However, in relatively larger networks ( $n = 100$ ), ND is worse than MinPri and even worse than MaxDeg and SP, because different forward nodes may designate different 1-hop neighbors to cover their common 2-hop neighbors, which causes redundancy in the forward node set.

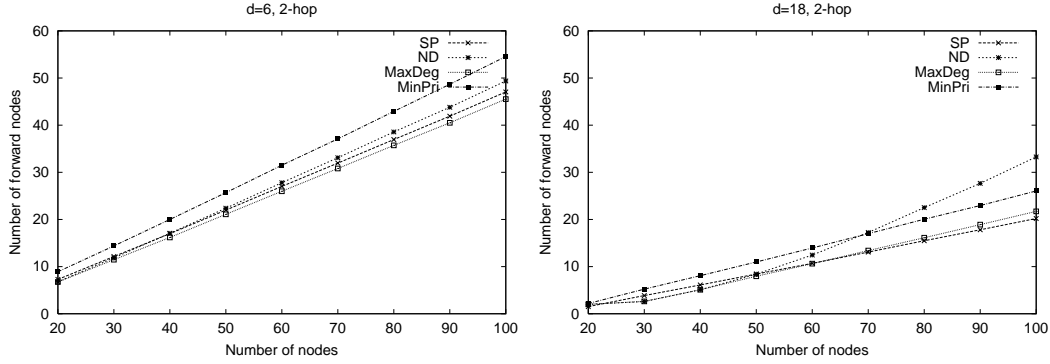


Figure 11: Performances of dynamic (first-receipt) algorithms with different selection options. The average node degree is 6 (left) and 18 (right).

The ND algorithm has the lowest computation cost, because only designated nodes need to compute the forward node set. But ND performs poorly in relatively dense networks. SP, MaxDeg and MinPri have almost the same computation cost, because all nodes are required to check the coverage condition. Among them, MinPri is the worst in all circumstances, MaxDeg is the best in relatively sparse networks, and SP is the best in relatively dense networks. Because the computation cost is not as important as the communication cost, MaxDeg and SP are preferable over other algorithms. Note that MaxDeg is a new algorithm derived from the generic framework.

**Space issue.** Figure 12 compares performances of different algorithms that are based on 2-hop, 3-hop, 4-hop, and 5-hop information. Also, performance under global information (Global) is also included as a lower bound. Although the performance progressively improves as the hop count increases, the difference becomes marginal as the hop count increases. In fact, algorithms based on 2- and 3-hop information do not perform significantly worse than the one based on the global information. Considering the cost in gathering neighborhood information, algorithms based on 4-, 5-hop, or global information are not cost-effective compared with the ones based on 2- or 3-hop information.

**Priority issue.** Figure 13 compares performances of different algorithms using node id (ID), node degree (Degree), and neighborhood connectivity ratio (NCR) as priority values. ID requires no extra maintenance cost but produces more forward nodes. Degree has higher maintenance cost and produces less forward nodes than ID. NCR has the highest maintenance cost and produces the smallest forward node set. Note that in relatively sparse networks, Degree is much better than ID and is very close to NCR. This is because node degree and neighborhood connectivity ratio are better indicators in selecting those nodes in some “critical” positions that are essential in replacement paths of other nodes. When node id is used as the priority, many “non-critical” nodes become forward nodes only because they have higher id’s. In relatively dense networks, all three metrics stay very close. Since there are few critical positions in a dense network, the importance of a

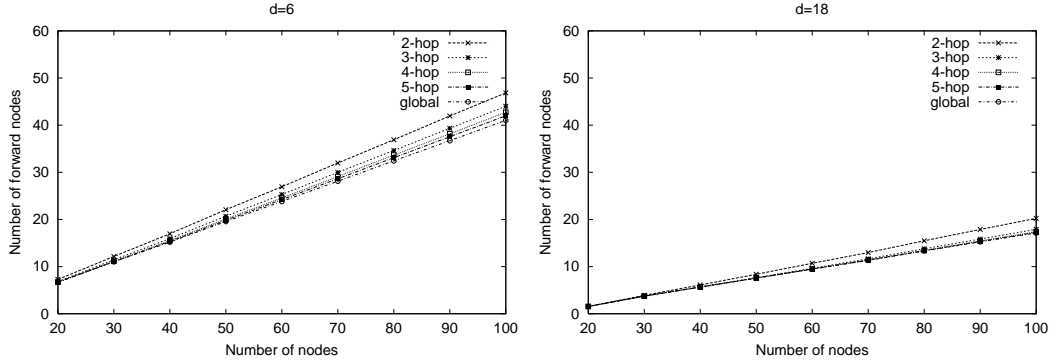


Figure 12: Performances of dynamic self-pruning algorithms based on different local views. The average node degree is 6 (left) and 18 (right).

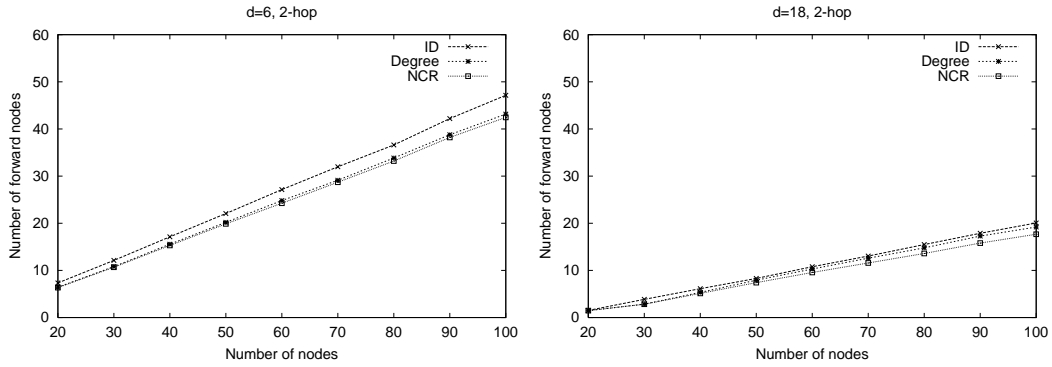


Figure 13: Performances of dynamic self-pruning algorithms using different priority values. The average node degree is 6 (left) and 18 (right).

good indicator is reduced. Considering the cost of collecting and maintaining degree and NCR information, Degree in relatively dense networks and NCR in general has the worst cost-effectiveness. Tradeoffs must be made between performance and maintenance cost in selecting ID and Degree in relatively sparse networks.

Overall, there is no single combination of implementation options that is the best for all circumstances. Fine tuning is needed to achieve better tradeoff between performance and overhead based on the types of ad hoc networks and applications.

Table 1: Existing distributed broadcast algorithms compared in the simulation.

Category	Self-pruning	Neighbor-designating
Static	Rule $k$ , Span	MPR
First-receipt	LENWB	DP, PDP
First-receipt-with-backoff	SBA	-

## 7.2 Special cases

Here we compare performances of several existing special cases of the generic framework, including Dai and Wu’s algorithm (Rule- $k$ ), the enhanced Span (Span), MPR, LENWB, dominant pruning (DP), partial dominant pruning (PDP), and SBA. These algorithms can be divided into static and dynamic algorithms, and dynamic algorithms can be further divided into first-receipt and first-receipt-with-backoff algorithms. Each category, except the last one, contains both self-pruning and neighbor-designating algorithms, as shown in Table 1. For the sake of fairness, only algorithms under the same category are compared. Three new algorithms derived from the generic framework using the coverage condition, one for each category, are simulated and compared with existing algorithms. The corresponding performance data of these new algorithms are labelled “Generic” in the result diagrams.

**Static algorithms.** Figure 14 compares four static broadcast algorithms. All algorithms except MPR use NCR as the priority value, as it is the original configuration of Span. In MPR, the first designator’s transmission time is used to as the priority function in reducing the number of forward nodes. The sequence from the worst performance to the best performance is MPR, Span, Rule  $k$ , and Generic. MPR is less efficient in relative dense networks, because of un-coordinated forward node sets designated by different nodes. Span is slightly worse than Rule  $k$ , because of its restriction on the length of each replacement path. Generic performs slightly better than Rule  $k$ , because it uses the original coverage condition and has no constraint on the lengths of replacement paths; while Rule  $k$  only uses the strong coverage condition.

**First-receipt algorithms.** Figure 15 compares three first-receipt broadcast algorithms. All algorithms use node degree as priority values, as it is the original configuration of LENWB. The sequence from the worst performance to the best performance is DP, PDP, LENWB, and Generic. Both DP and PDP are much worse than the other two algorithms, because neighbor-designating in general is worse than self-pruning, and cannot take advantage of the 1-hop priority. PDP is better than DP, since it has fewer 2-hop neighbors to cover than DP does. LENWB is slightly worse than Generic, and can be viewed as a good approximation of Generic. Note that LENWB uses less broadcast state information than Generic. In LENWB, only the last visited node is used in checking the strong coverage condition. In Generic, each node also knows the second last visited node that is piggybacked in the broadcast packet. However, this extra broadcast state information



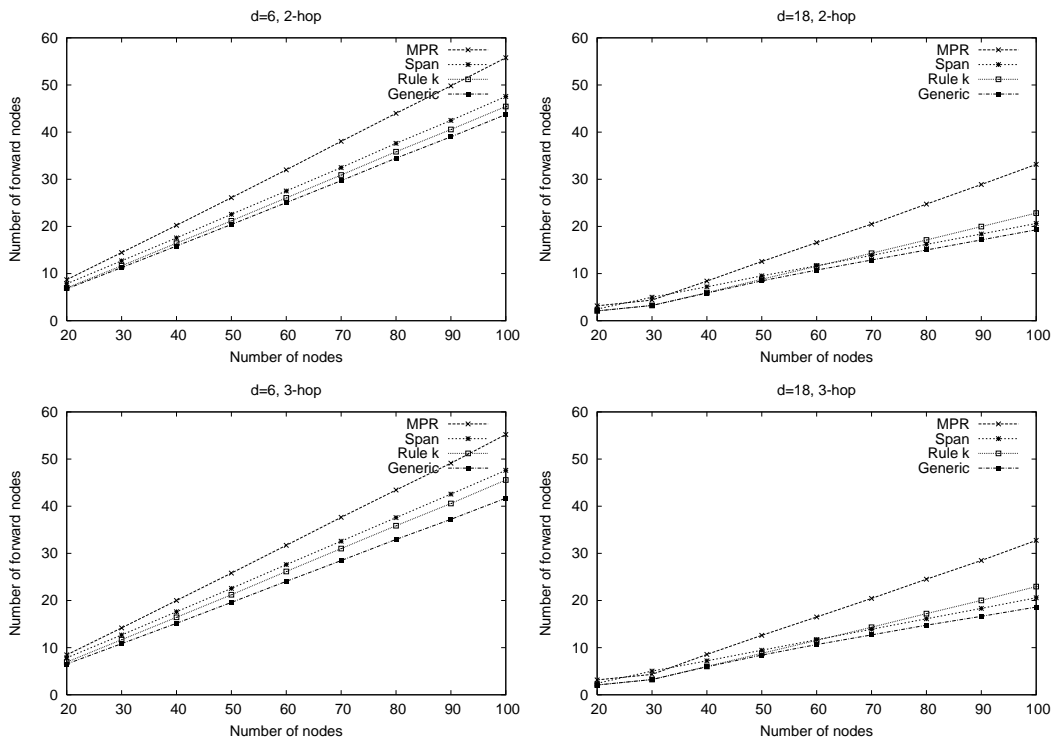


Figure 14: Performances of several static broadcast algorithms based on 2-hop (the upper row) and 3-hop (the lower row) neighborhood information. The average node degree is 6 (the left column) and 18 (the right column).

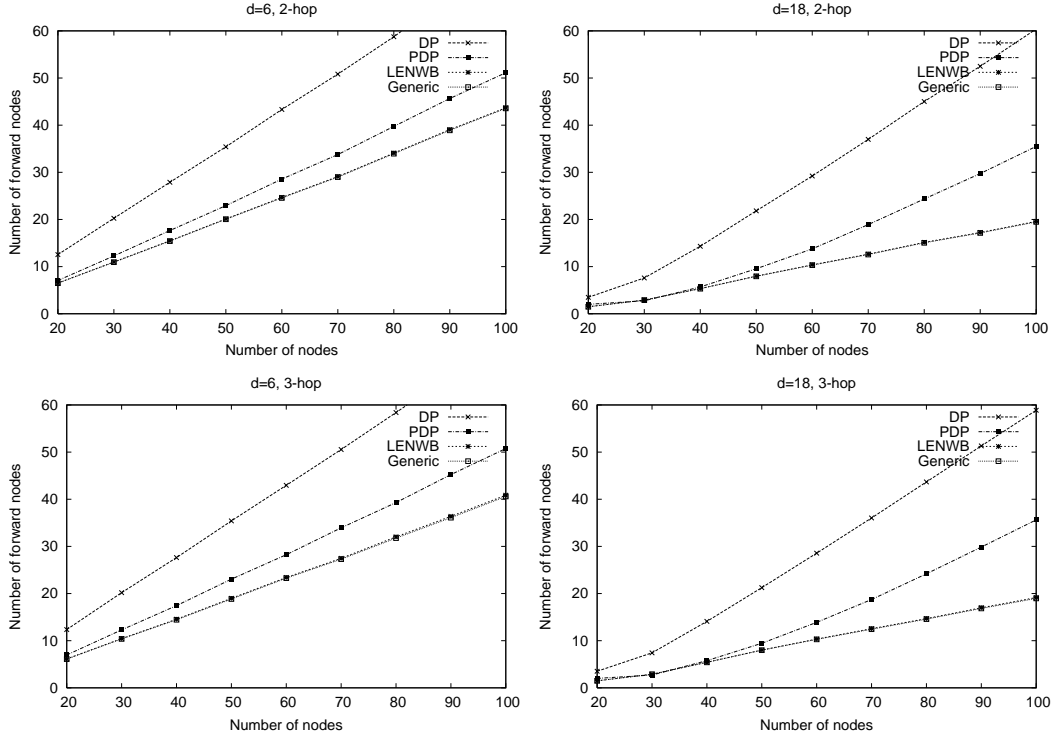


Figure 15: Performances of several first-receipt broadcast algorithms based on 2-hop (the upper row) and 3-hop (the lower row) neighborhood information. The average node degree is 6 (the left column) and 18 (the right column).

has little impact on performance. A similar observation has also been reported in [24].

**First-receipt-with-backoff algorithms.** Figure 16 compares two first-receipt-with-backoff algorithms. Generic significantly outperforms SBA, because SBA requires direct neighbor set coverage, while Generic allows indirect coverage. More specifically, a node does not need to forward a broadcast packet even if some of its neighbors are not directly covered by any visited node, but are indirectly connected to a visited node via several intermediate nodes with higher priorities.

Overall, within each category, the generic algorithm performs better than existing self-pruning algorithms, which in turn, perform better than existing neighbor designating algorithms.

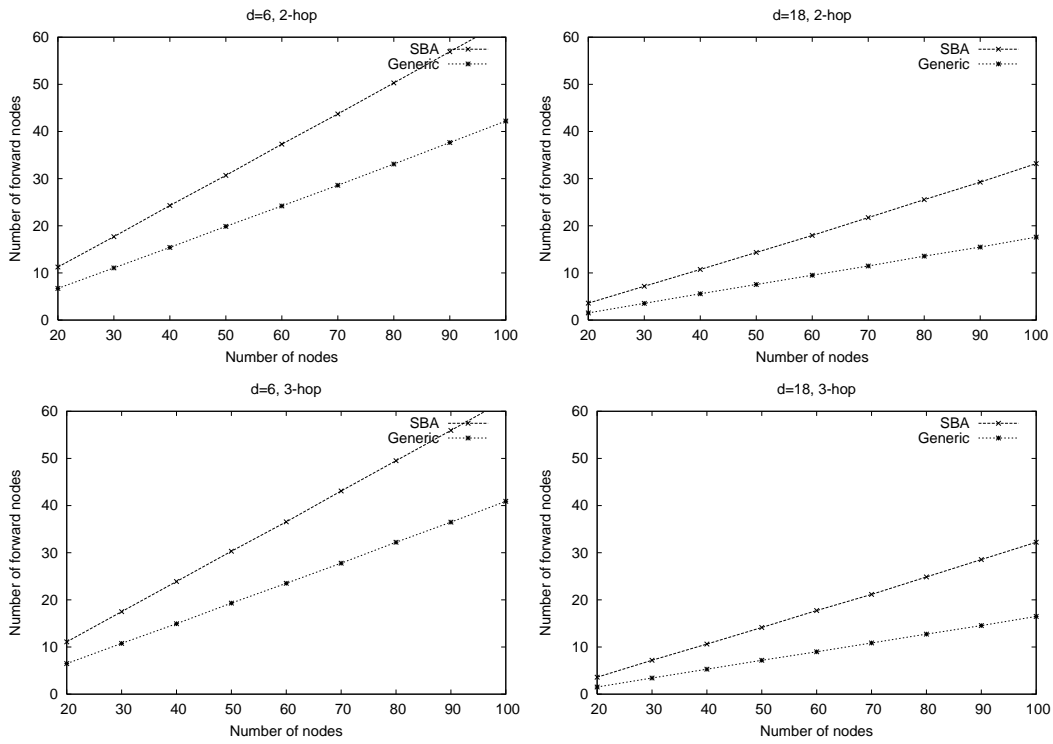


Figure 16: Performances of two first-receipt-with-backoff broadcast algorithms based on 2-hop (the upper row) and 3-hop (the lower row) neighborhood information. The average node degree is 6 (the left column) and 18 (the right column).

## 8 Conclusion

A generic framework of distributed broadcasting in ad hoc networks has been proposed and its correctness has been shown. Four important implementation issues, namely timing, selection, space, and priority, have been discussed and their impacts on broadcast efficiency have been examined. Nine existing broadcast algorithms, which represent a broad spectrum of state-of-art distributed broadcast techniques in ad hoc networks, have been shown to be special cases of the generic framework. Simulation results show that, by adjusting the four implementation options, the generic distributed broadcast protocol can be well adapted to different configurations of ad hoc networks and upper layer applications. We have also shown that several new algorithms can be derived from the generic framework, and that these algorithms produce smaller forward node sets than existing broadcast algorithms under the same requirement of neighborhood information.

## References

- [1] S. Alagar and S. Venkatesan. Reliable broadcast in mobile wireless networks. In *Proc. of MILCOM'95*, pages 236–240, 1995.
- [2] K. M. Alzoubi, P.-J. Wan, and O. Frieder. Distributed heuristics for connected dominating sets in wireless ad hoc networks. *Journal of Communications and Networks*, 4(1):22–29, Mar. 2002.
- [3] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *ACM Wireless Networks Journal*, 8(5):481–494, Sep. 2002.
- [4] Y. P. Chen and L. Liestmen. Approximating minimum size weakly-connected dominating sets for clustering mobile ad hoc networks. In *Proceedings of the third ACM international symposium on Mobile ad hoc networking & computing (MobiHoc)*, pages 165–172, 2002.
- [5] F. Dai and J. Wu. Distributed dominant pruning in ad hoc wireless networks. *accepted to appear in Proceedings of IEEE International Conference on Communications (ICC)*, May 2003.
- [6] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374–387, Apr. 1998.
- [7] Z. J. Haas, J. Y. Halpern, and L. Li. Gossip-based ad hoc routing. In *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 3, pages 1707–1716, June 2002.

- [8] H. B. Hunt, M. V. Marathe, V. Radhakrishnan, S. S. Ravi, D. J. Rosenkrantz, and R. E. Stearns. NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. *J. Algorithms*, 26(2):238–274, 1998.
- [9] W. H. Liao, Y. C. Tseng, and J. P. Sheu. GRID: A fully location-aware routing protocol for mobile ad hoc networks. *Telecommunication Systems*, 18:37–60, 2001.
- [10] H. Lim and C. Kim. Flooding in wireless ad hoc networks. *Computer Communications Journal*, 24(3-4):353–363, 2001.
- [11] C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 15(7):1265–1275, 1996.
- [12] W. Lou and J. Wu. On reducing broadcast redundancy in ad hoc wireless networks. *IEEE Transactions on Mobile Computing*, 1(2):111–123, Apr.-June 2002.
- [13] E. Pagani and G. P. Rossi. Providing reliable and fault-tolerant broadcast delivery in mobile ad hoc networks. *Mobile Networks and Applications*, 4:175–192, 1999.
- [14] A. Pelc. *Broadcasting in Radio Networks*. in Handbook of Wireless Networks and Mobile Computing, Edited by I. Stojmenovic, John Wiley & Sons, Inc, 2002.
- [15] W. Peng and X. Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. In *Proceedings of the first ACM international symposium on Mobile and ad hoc networking & computing (MobiHoc)*, pages 129–130, 2000.
- [16] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying for flooding broadcast message in mobile wireless networks. volume 9, page 298, Jan. 2002.
- [17] V. Ramasubramanian, R. Chandra, and D. Mosse. Providing a bidirectional abstraction for unidirectional ad hoc networks. In *Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 3, pages 1258–1267, June 2002.
- [18] I. Stojmenovic. *Handbook of Wireless Networks and Mobile Computing*, chapter Location Updates for Efficient Routing in Ad Hoc Networks, pages 451–471. Edited by I. Stojmenovic, John Wiley & Sons, Inc., Feb. 2002.
- [19] 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*, 13(1):14–25, Jan. 2002.
- [20] J. Sucec and I. Marsic. An efficient distributed network-wide broadcast algorithm for mobile ad hoc networks. CAIP Technical Report 248, Rutgers University, Sep. 2000.

- [21] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. *Wireless Networks*, 8(2/3):153–167, Mar.-May 2002.
- [22] P. J. Wan, K. Alzoubi, and O. Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. *Proc. of IEEE INFOCOM'2002*, 3:1597–1604, June 2002.
- [23] J. Wu. Extended dominating-set-based routing in ad hoc wireless networks with unidirectional links. *IEEE Transactions on Parallel and Distributed Systems*, 13(9):866–881, Sept. 2002.
- [24] J. Wu and F. Dai. Broadcasting in ad hoc networks based on self-pruning. *accept to appear in Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Mar./Apr. 2003.
- [25] J. Wu and H. Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proc. of the 3rd Int'l Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (Dial M)*, pages 7–14, 1999.
- [26] J. Wu and W. Lou. Forward-node-set-based broadcast in clustered mobile ad hoc networks. *accepted to appear in Wireless Communication and Mobile Computing*, 2003.