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

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 itself (self-pruning) or by other nodes (neighbor-designating). Node status can be determined at different snapshots of network state along time (called views) without causing problems in broadcast coverage. 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. This work is an extension to an early work in which only self-pruning methods are discussed [16].*

1 Introduction

Broadcasting is a special routing process of transmitting a packet so that each node in an ad hoc wireless network (or simply ad hoc 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. Flooding ensures the coverage, 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. In Figure 1 (b), when node v broadcasts a packet, 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* [14], in which redundant packets cause contention and collision.

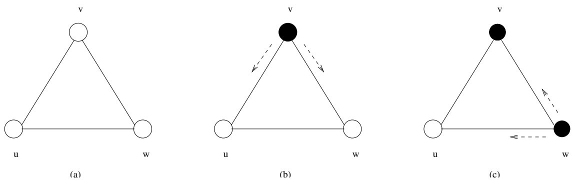

Figure 1. Three different views.

We focus on the deterministic approach to find a forward node set, which can be selected statically (independent of any broadcast process) [2, 3, 10, 17] or dynamically (during a particular broadcast process) [5, 7, 9, 13]. 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 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, forwarding or non-forwarding, 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 topology [4] or pseudo-global network topology (that exhibits "sequentialized propagation") [1] 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 [18]. Under the objective of selecting a small CDS, the status of each node is decided in a decentralized manner based on

---

*This work was supported in part by NSF grant CCR 9900646 and grant ANI 0073736. Email:{jie,fdai}@cse.fau.edu.

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 (*self-pruning*) or by other nodes (*neighbor-designating*). 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. 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.

The generic broadcast scheme is an extension to an early generic framework for self-pruning methods [16]. The extended framework support both self-pruning and neighbor-designating approaches. In addition, the new view-based model is more effective in formalizing reactive processes.

2 Preliminaries

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 that all nodes have fresh topology information at the beginning of the broadcast period, and 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. For example, in Figure 1 (a), $Pr(V) = (Pr(u), Pr(v), Pr(w)) = ((1, u), (1, v), (1, w))$, and in Figure 1 (b), $Pr(V) = ((1, u), (2, v), (1, w))$, where visited nodes are colored black. The lexicographical order can be used to order nodes based on their priorities; e.g., $(1, w) > (1, v)$ and $(2, v) > (1, w)$. 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.

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).

Throughout the paper, it is assumed that each node only captures a local view. Note that any visited nodes are assumed to be connected under any local view, since they are all connected to the source. Four 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 [8, 12]. (3) Network topology is a connected graph without unidirectional links. A sublayer can be added [11, 15] to provide a bidirectional abstraction for unidirectional ad hoc networks. (4) The network is relatively sparse. For a dense ad hoc network, the *clustering approach* [6, 18] 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 forward 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.

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-forward status). A replacement path may include some visited nodes, which 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 $(u, u_1, u_2, ..., u_l, w)$ exists such that $Pr(u_i) > Pr(v)$.

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.*

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 was proved in [16] under the assumption that the priority of one node is the same to all its observers. This assumption holds only 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. $F = (f_1(v_1), f_2(v_2), ..., f_n(v_n))$ captures the forwarding status of all nodes in the network under their corresponding local views. Define $View_{super} = (G_{super}, Pr_{super}(V))$, where $G_{super} = (V_{super}, E_{super}) = (V(v_1) \cup V(v_2) \cup ... \cup V(v_n), E(v_1) \cup E(v_2) \cup ... \cup E(v_n))$ and $Pr_{super}(v_i) = max\{Pr_1(v_i), Pr_2(v_i), ..., Pr_n(v_i)\}$.

Each $Pr_l(v_i)$ has three potential forms/values: the priority of an invisible node $(0, id(v_i))$, the lowest priority; the priority of an un-visited node $(1, id(v_i))$; 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, $F_{super} = (f_{super}(v_1), f_{super}(v_2), ..., f_{super}(v_n)) \leq F = (f_1(v_1), f_2(v_2), ..., f_n(v_n))$. Applying Theorem 1 to $View_{super}$, 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. □

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.

4 Discussion

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*. 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.

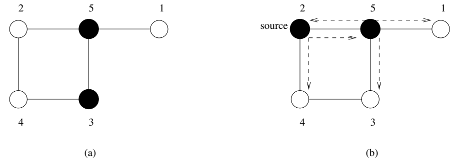

Figure 2. Two broadcast processes.

Figure 2 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 2 (a)) and one with broadcast state based on the dynamic version of the coverage condition (Figure 2 (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 2 (a) can also be interpreted as the one for any broadcasting. In Figure 2 (b), because nodes 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).

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 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.

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*) for a small k, is used as an approximation. We define k-hop information for a node v as the local view of network topology $G_k(v)$ that takes at least k rounds of neighborhood information exchanges to build up. 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). We normally assume that network topology information is not piggybacked, since the broadcast packet needs to be kept relatively small.

Priority issue. The priority function used in the coverage condition can also affect the resultant forward node set. Based on the difficulty in collecting the priority values, node properties used in the priority function can be divided into three categories: (1) 0-hop priority such as node id, denote as $id(v)$. (2) 1-hop priority such as node degree, $deg(v)$, which is defined as the number of v's neighbors. When $deg(v) = deg(u)$, the id's of u and v are usually used to break a tie. (3) 2-hop priority such as neighborhood connectivity ratio, $ncr(v)$, which is defined as the ratio of pairs of neighbors that are not directly connected to pairs of any neighbors. 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). Properties of these designators can be used as priority functions. Specifically, the transmission time of a node's first designator with respect to a specific broadcast packet is a 0-hop priority.

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 and designated forward nodes selected by the last visited node.

Algorithm 1 Generic Distributed Broadcast Protocol

```
Periodically each node $v$ exchanges "hello" messages with neighbors to update local network topology $G_k(v)$.
$v$ updates priority information $Pr$ based on snooped and piggybacked messages.
$v$ applies the coverage condition to determine its status.
If $v$ is a non-forward node then stop.
$v$ designates some neighbors as forward nodes if needed and updates its priority information $Pr$.
$v$ forwards the packet together with $Pr$.
```

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 a 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 is that of using 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 maximum number of nodes per unit area, while the computation complexity of the strong coverage condition is $O(D^2)$ [16].

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.

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 [17] 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.

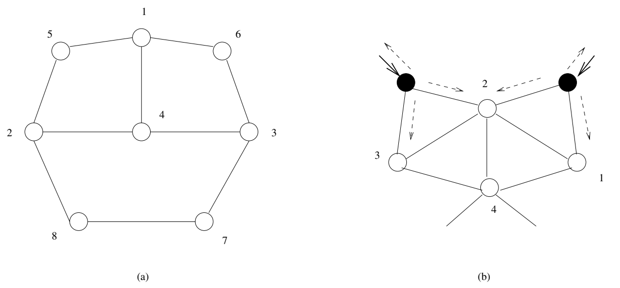

Figure 3. Two sample ad hoc networks.

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 [3] extended the previous algorithm by using a more generic pruning rule called Rule k without introducing additional cost: 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.

Span: Chen et al [2] 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. To ensure connectivity, all intermediate coordinators should have higher priorities. 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 original coverage condition is used with two restrictions: no visited node is used and each replacement path is no more than three hops. Figure 3 (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.

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 [9] 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 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.

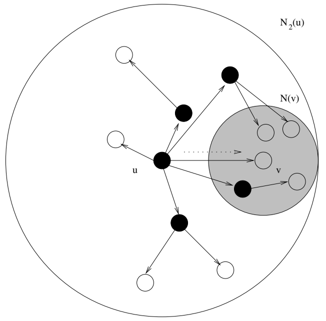

Figure 4. Neighbor-designating algorithms.

LENWB: Sucec and Marsic [13] proposed the Lightweight and Efficient Network-Wide Broadcast (LENWB) protocol, which also computes the forward node status on-the-fly. However, unlike in SBA, 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 Rule k, the connectivity requirement also needs information about nodes beyond 3 hops; however, a restricted implementation can be done using 2-hop information.

In SBA, the neighbor set is covered by a set of (connected) visited nodes. The first-receipt-with-backoff approach is used. In LENWB, the neighbor set is covered by one visited node (black node) and several un-visited but higher priority nodes. In this approach, the first-receipt approach is adopted. Figure 3 (b) shows a case of neighbor coverage that cannot be covered by SBA. After node 2 has two visited neighbors, neighbor 4 is still not covered based on SBA. 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.

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 [5] 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). In each node v, the DP algorithm uses a greedy approach to compute the forward node set of each node on-the-fly. 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 $Y = N_2(v) - N(u) - N(v)$, where $N_2(v) = \bigcup_{w \in N(v)} N(w)$ is the set of v's 2-hop neighbors.

Lou and Wu's algorithm: Lou and Wu [7] extended the DP algorithm by further reducing the number of 2-hop

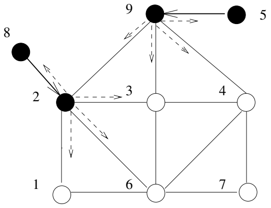

Figure 5. Different selection policies.

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 $N_2(u)$ along 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 $N_2(v)$.

Multipoint relays: Qayyum et al [10] proposed selecting multipoint relays (MPRs) from 1-hop neighbors to cover 2-hop neighbors. Visited nodes are not considered in the selection of MPRs and, therefore, the entire set of 2-hop neighbors must be covered. 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.

Both DP 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. TDP and PDP can be considered as DP with a better greedy algorithm in selecting the coverage set. Figure 4 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.

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 node degree (with respect to uncovered nodes in $N_2(v)$). 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 5 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.

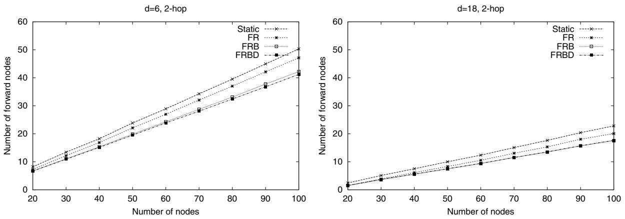

Figure 6. Various timing options.

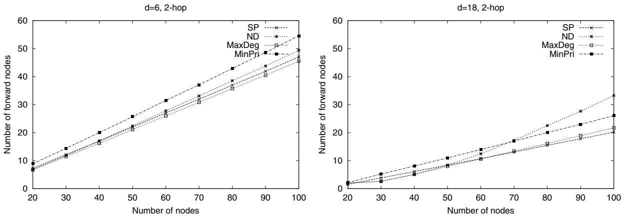

Figure 7. Various selection options.

7 Simulation

We evaluated the effect of various implementation issues, several new special cases of the generic scheme, and compared them with existing protocols. To generate a random ad hoc network, n hosts are randomly placed in a restricted area. The transmitter range is adjusted according to a given average node degree d to produce exactly $\frac{nd}{2}$ links. Two average node degrees are used, one for relatively sparse networks ($d = 6$) and another for relatively dense networks ($d = 18$). Networks that are not connected are discarded. Performances of various special cases are compared in terms of the average size of resultant CDS's. The default configuration uses 2-hop information and node id as priority. Each simulation is repeated until the 90% confidence interval of mean is within $\pm 1\%$.

Timing issue. Figure 6 compares performances of static, first-receipt (FR), and two first-receipt-with-delay algo-

Table 1. Simulated algorithms.

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

rithms that use random backoff delay (FRB) a backoff delay that is proportional to the inverse of node degree (FRBD), recpectively. 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 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. Because 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.

Priority issue. Simulation results in [16] show that, in relatively sparse networks, using $deg(v)$ as the priority of each node v is much better than $id(v)$ and is very close to $ncr(v)$. In relatively dense networks, all three metrics stay very close. Considering the cost of collecting and maintaining $deg(v)$ and $ncr(v)$, $deg(v)$ in relatively dense networks and $ncr(v)$ in general has the worst cost-effectiveness. Tradeoffs must be made between performance and maintenance cost in selecting $id(v)$ and $deg(v)$ 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.

Selection issue. Figure 7 compares performances of the self-pruning (SP), neighbor-designating (ND), and two hybrid algorithm: 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. Note that MaxDeg is a new algorithm derived from the generic framework.

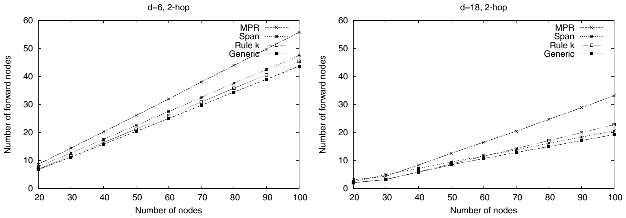

Figure 8. Static algorithms.

Space issue. Simulation results in [16] show that 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.

We also compare performances of several existing special cases of the generic framework, as shown in Table 1.

Static algorithms. Figure 8 compares four static broadcast algorithms and a new static algorithm derived from the generic framework labelled "Generic". All algorithms except MPR use $ncr(v)$ as the priority value of each node v, 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 9 compares three first-receipt broadcast algorithms and a new first-receipt algorithm labelled "Generic". 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

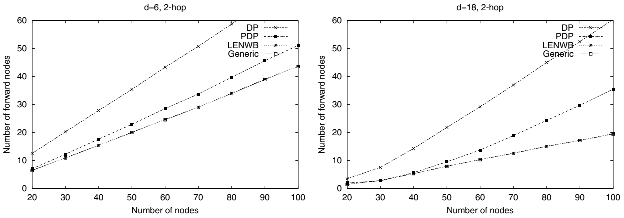

Figure 9. First-receipt algorithms.

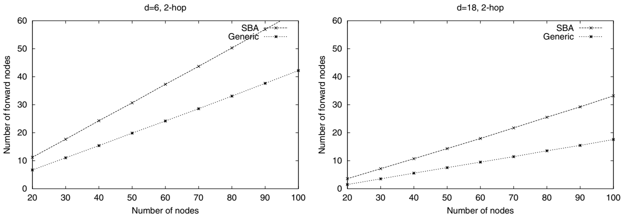

Figure 10. First-receipt-with-backoff algorithms.

Generic, each node also knows the second last visited node that is piggybacked in the broadcast packet. However, this extra broadcast state information has little impact on performance.

First-receipt-with-backoff algorithms. Figure 10 compares SBA and a new first-receipt-with-backoff algorithm labelled "Generic". 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.

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 examined. Seven 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] 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.

[2] 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.

[3] F. Dai and J. Wu. Distributed dominant pruning in ad hoc wireless networks. *accpeted to appear in Proc. of ICC*, May 2003.

[4] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374–387, Apr. 1998.

[5] H. Lim and C. Kim. Flooding in wireless ad hoc networks. *Computer Communications Journal*, 24(3-4):353–363, 2001.

[6] C. R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 15(7):1265–1275, 1996.

[7] 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.

[8] A. Pelc. *Broadcasting in Radio Networks*. in Handbook of Wireless Networks and Mobile Computing, Edited by I. Stojmenovic, John Wiley & Sons, Inc, 2002.

[9] W. Peng and X. Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. In *Proc. of MobiHoc*, pages 129–130, 2000.

[10] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying for flooding broadcast message in mobile wireless networks. In *Proc. of HICSS-35*, volume 9, page 298, Jan. 2002.

[11] V. Ramasubramanian, R. Chandra, and D. Mosse. Providing a bidirectional abstraction for unidirectional ad hoc networks. In *Proc. of IEEE INFOCOM*, volume 3, pages 1258–1267, June 2002.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] J. Wu and F. Dai. Broadcasting in ad hoc networks based on self-pruning. *accept to appear in Proc. of IEEE INFOCOM*, Mar./Apr. 2003.

[17] J. Wu and H. Li. On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proc. of DialM*, pages 7–14, 1999.

[18] 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.