Localized Broadcasting in Mobile Ad Hoc Networks Using Neighbor Designation

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

This work was supported in part by NSF grants CCR 9900646, ANI 0073736, and EIA 0130806.

I. Introduction

A mobile ad hoc network (or ad hoc network) [13] enables wireless communications between participating mobile nodes without the assistance of any base station. Two nodes that are out of one another's transmission range need the support of intermediate nodes which relay messages to set up a communication between each other. The broadcast operation is the most fundamental role in ad hoc networks because of the broadcasting nature of radio transmission: When a sender transmits a packet, all nodes within the sender's transmission range will be affected by this transmission. This is usually referred to as the *promiscuous receive mode.* The advantage is that one packet can be received by all neighbors; the disadvantage is that it interferes with the sending and receiving of other transmissions, creating *exposed terminal problem*, that is, an outgoing transmission collides with an incoming transmission, and *hidden terminal problem*, that is, two incoming transmissions collide with each other. Broadcast operation has extensive applications, such as when used in the route query process in routing protocols [20], [33], [37], when sending error messages to erase invalid routes [32], or when used as an efficient mechanism for reliable multicast in highly dynamic wireless networks [17].

In general, broadcasting refers to a process of transmitting a packet so that each node in a network receives a copy of this packet. *Flooding* is the most simple approach for broadcasting where every node in the network forwards the packet exactly once. Flooding ensures the full coverage of all the network, that is, the broadcast packet is guaranteed to be sent to every node in the network, providing the network is static and connected and the MAC layer of the communication channel is error-free during the broadcast process. However, flooding generates many redundant transmissions. Figure 1 shows a sample network with five nodes. When node v broadcasts a packet, nodes u, w and x will receive the packet. u, w and x will then forward the packet and lastly y will also broadcast the packet. Apparently, there is much broadcast redundancy for blind flooding in this case. Transmitting the broadcast packet only by nodes v and u is enough for a broadcast operation. When the size of the network increases and the network becomes dense, more transmission redundancy will be introduced and these transmissions are likely to trigger considerable transmission collision and contention. This is a serious *broadcast storm problem* [31] that

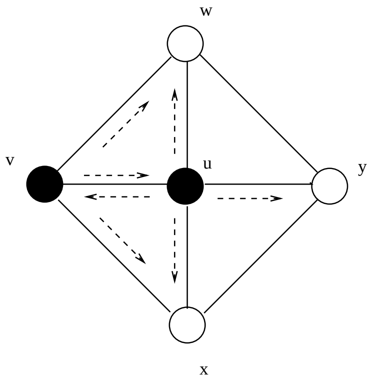

Fig. 1. A sample ad hoc network with five nodes.

finally collapses the whole network.

The broadcast storm problem can be avoided by reducing the number of nodes that retransmit the broadcast packet. Ni et al classified the broadcast algorithms into two categories: *probabilistic approach* and *deterministic approach* [31]. Counter-based, distance-based, and location-based schemes belong to probabilistic approaches. For the deterministic approach, Wu and Lou [52] further classified it into four classes: *global*, *quasi-global*, *quasi-local*, and *local* algorithms. In [49], Wu and Dai classified the local broadcast algorithms into three classes: self-pruning, neighbor-designating, and hybrid algorithms. (see Figure 2)

We first classify the basic broadcast algorithms in Section II. Then, in Section III, we describe several neighbor-designating-based broadcast algorithms. We discuss three extensions of the neighbor-designating-based broadcast algorithms in Section IV. In the last section, we summarize the topic of neighbor-designating-based broadcast algorithms.

II. Classification

This section discusses in detail three levels of classification of broadcast algorithms.

A. Probabilistic algorithms

The probabilistic approach for a broadcast operation is as follows: Upon receiving a broadcast packet, each node forwards the packet with probability p. The value p is determined by relevant information gathered at each node. Although the probabilistic approach provides a good stochastic result, it cannot guarantee the full coverage. In [31], other prob-

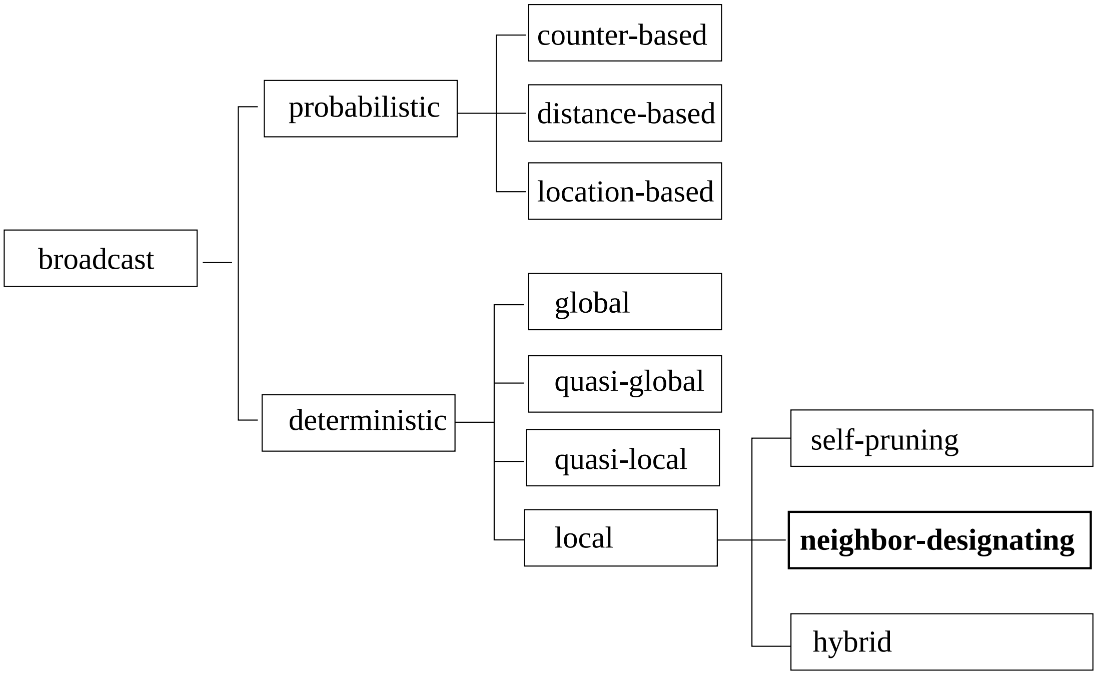

Fig. 2. Classification of broadcast algorithms.

abilistic approaches were also discussed:

Counter-based scheme. Upon the reception of a previously unknown packet, the node initiates a waiting timer and a counter. The counter increases one for each received redundant packet. When the waiting timer expires, if the counter is larger than a threshold value, the node will not rebroadcast the packet; otherwise, the node will broadcast it.

Distance-based scheme. Upon the reception of a previously unknown packet, the node initiates a waiting timer. Before the waiting timer expires, the node checks the location of the senders of each received packet. If any sender is closer than a threshold distance value, the node will not rebroadcast the packet. Otherwise, the node rebroadcasts it when the waiting timer expires.

Location-based scheme. Upon the reception of a previously unknown packet, the node initiates a waiting timer and accumulates the coverage area that has been covered by the arrived packet. When the waiting timer expires, if the accumulated coverage area is larger than a threshold value, the node will not rebroadcast the packet. Otherwise, the node will broadcast it.

Other enhancements to the above probabilistic algorithms are discussed in [5], [15], [44].

B. Deterministic algorithms

The deterministic approaches provide full coverage of the network for a broadcast operation, that is, only a subset of nodes forward the broadcast packet and the remaining nodes are adjacent to the nodes that forward the packet. The nodes that forward the broadcast packet form a *forward node set* for a particular broadcast operation. Basically, a forward node set is a *connected dominating set* (CDS). A *dominating set* (DS) is a subset of nodes such that every node in the graph is either in the set or is adjacent to a node in the set. If the subgraph induced from a DS of the network is connected, the DS is a CDS. Finding a *minimum connected dominating set* (MCDS) in a given graph is NP-complete; in a unit disk graph, it has also been proved to be NP-complete [30]. There are in general two models of neighbor set information: *neighbor set without node positions* and *neighbor set with node positions* (obtained through GPS or other means). The latter model "trivializes" the approximation problem of CDS. That is, approximation algorithms with a constant approximation ratio can be easily derived. On the other hand, finding an approximation algorithm with a small constant approximation ratio is still a challenging issue in the absence of global network information [4]. 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).

The CDS of a graph can be constructed with global or local information. The distinction between global and local is not a clear-cut situation. Through several rounds of sequential information exchanges, global information can be assembled based on local information only. However, sequential information propagation can be costly. There are four types of broadcast protocols [52]:

Global. Broadcast protocols, centralized or distributed, are based on global state information. The most widely used global broadcast protocol is based on Guha and Khuller's approximation algorithm [14] and has been used in protocol design by Das et al [10]: All nodes are initially colored white. The node with the maximum node degree is selected and colored black, all its neighbors are colored gray. An iterative selection process runs until there is no white node left. Select a gray node that has the maximum number of white neighbors, color the selected node black and its white neighbors gray. The resultant set of black nodes is a CDS. This algorithm is centralized and works well except for some extreme cases. A modified algorithm selects the gray node u and its neighbor v if they can cause the maximum number of white nodes to change color to gray when both u and v are changed to black. The modified algorithm guarantees an approximation ratio $O(\ln \Delta)$ under any random graph, where Δ is the maximum node degree of the network. Therefore, this algorithm can be used as a lower bound of the MCDS.

Quasi-global. Distributed broadcast protocols are based on partial global state information. Unlike the global broadcast protocol, the quasi-global broadcast protocol does not need to collect the whole global state. Only partial global state information is collected, typically with the help of a global infrastructure such as a spanning tree. The protocol proposed in [3], [45] fits into this category: A spanning tree is first constructed starting from the selected root (through an election process), a *maximal independent set* (MIS) is constructed level by level down the tree. An *independent set* (IS) is a set in which no two nodes are neighbors. An MIS is an IS in which any other node in the network is a neighbor of a node in the set. Nodes in the MIS are colored black. Clearly, an MIS is a DS. Specifically, nodes are labelled according to a topological sorting order of the tree. Then nodes are labelled based on their positions in the order starting from the root v. All nodes are white initially. The root v is marked black first and other nodes are marked black unless there is a black neighbor. Each parent of a black node in the tree acts as a *connector* by marking gray. Black and gray nodes form a CDS. This *spanning-tree-based CDS broadcasting* (STCDS) generates a CDS with a constant approximation ratio of 8 (i.e., the size of the CDS is at most 8 times the size of the MCDS). Also, other than tree level information needed to determine the topological sorting order of each node, no other global state information is distributed. However, like the MCDS, the STCDS requires $O(diam)$ sequential rounds, because both the spanning tree construction and status marking process are serialized. In addition, the STCDS does not support locality of maintenance. Movements of hosts may trigger the re-construction of the whole spanning tree (as will be shown in the next section).

Quasi-local. Distributed broadcast protocols are based on mainly local state information and occasional partial global state information. The cluster approach falls into the quasi-local model. Cluster structure is a two-level hierarchical structure and it is formed by first electing a clusterhead and, then, its neighbors joining in the cluster as non-clusterhead members. There are many clustering approaches [7], [8], [11], [12], [23]. A simple one, called the *lowest-ID cluster algorithm* (LID), initializes all nodes white. When a white node finds itself having the lowest ID among all its white neighbors, it becomes a clusterhead and colors itself black. All its white neighbors join in the cluster and change their colors to gray. This iterative process continues until there is no white node left. The black nodes form the set of clusterheads. Each gray node belongs to one and only one clusterhead. That clusterhead is called the dominator of the gray node. The clusterhead and its dominated gray neighbors form a cluster. The LID may exhibit sequential propagation, as happens when the network is a linear chain with decreasing IDs from one end to the other end (this is the reason this approach is called quasi-local), resulting in $O(diam)$ rounds of information exchanges. However, this situation rarely happens. In the average case, the cluster formation can be considered as a localized process. Clusterheads form a DS, but not a CDS. Clusterheads and gateways together form a CDS.

Another variation of cluster approach proposed by Sinha, Sivakumar and Bharghavan [41], called *core broadcast* (CB), also includes the selection process of forward nodes: Initially all nodes are white. A white node determines its dominator by selecting its black neighbor that has the maximum number of nodes that regard this black node as their dominators. In case there is no black neighbor, the white node selects the node (white or gray) with the maximum node degree within its 1-hop neighborhood (including itself) as its dominator. After the white node has chosen its dominator, it colors itself gray if it is not selected as a dominator by itself or by its neighbors; otherwise, it marks itself black if it has been selected as a dominator. The coloring process continues until no white node is left. Eventually, all the black nodes become cores. In the core broadcast, each node computes its forward node set. A node's forward node set includes all its black neighbors. It also includes those gray neighbors that either have a black neighbor that is not covered by the forward node set or have a gray neighbor whose dominator is not covered by the forward node set. The core broadcast requires only the nodes in the forward node set relay the broadcast so it reduces the broadcast redundancy. The set of cores, like the set of clusterheads, is a DS of the network. While the set of clusterheads is also an IS, the set of cores does not have this property since two cores may be neighbors.

Local. Distributed broadcast protocols are based on solely local state information. The local broadcast protocol is based on solely local information without exhibiting any sequential propagation of state information. It also supports locality of maintenance. However, although this approach is competitive in the average case, it does not guarantee performance in the worst case such as a constant approximation ratio. Wu and Li's [51] *marking process* is a local broadcast protocol: All nodes are initially white. A node marks itself black only when it has two unconnected neighbors. After the marking process, the black nodes form a CDS. Rules 1 and 2 aim to remove redundant nodes from the CDS. Rule 1 allows a black node u to change its color black to white if it can find another black node v, with $id(u) < id(v)$, to cover all u's neighbors. For Rule 2, a black node u changes itself to white if there exist two connected nodes v and w, with $id(u) = min\{id(u), id(v), id(w)\}$, that can collectively cover all u's neighbors. Recently, Dai and Wu [9] have extended Rules 1 and 2 to Rule k to further reduce nodes in the CDS without increasing the computational complexity. If a black node u can be covered by k connected black nodes and $id(u)$ is smaller than any ID of these k nodes, then u can change itself to white. A constant number of rounds (2 or 3 depending on the implementation) are needed to construct a CDS and its maintenance. Wu and Li's approach has been applied to broadcasting in [43] where only black nodes (besides the source) forward the broadcast packet.

C. Local algorithms

Wu and Dai [49] proposed a deterministic generic distributed broadcast scheme and classified the local broadcast algorithms into three categories, called *self-pruning*, *neighbor-designating*, and *hybrid* broadcasting approaches. In all these schemes, the status of each node is determined in a decentralized manner based on node's current *local view*. A *view* is a snapshot of network state, including network topology and broadcast state, along time. A node can utilize its K-hop neighborhood information to build its local view. 1-hop and 2-hop neighborhood information are the most common cases. Also, the broadcast packet can carry some broadcast state information, such as the next selected node to forward the packet, the recently visited nodes and their neighbor sets. The status of each node can be determined by itself or by its neighbor. For a specific node, the upstream node that has sent a broadcast packet to this node is viewed as a *forwarded node*; the downstream node selected by this node to forward the broadcast packet is viewed as *forward node*; the downstream node that is designated not to forward the packet is viewed as a *non-forward node*. Notice that the node status under the current view will be changed in the next view, that is, a forward node in current view will be a forwarded node in the next view.

The generic distributed broadcast scheme constructs a CDS for a particular broadcast that depends on the location of the source and the progress of the broadcast process. Each node v determines its status and the status of some of its neighbors under a current local view. Each node has the forwarding status by default, and the status can be changed to non-forwarding if the *coverage condition* is met. Coverage condition is described as follows:

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 the one of v.

The coverage condition indicates that when every pair of neighbors of v can be connected through other nodes, node v, as the connecting node for its neighbors, can be replaced (i.e., can take the non-forwarding status). Note that "replacement" can be applied iteratively. To avoid possible "cyclic dependency" situations, a total priority order needs to be defined among all nodes, such as node ID or a pair of node degree and node ID.

In self-pruning approaches, a node will resign its role of forwarding the broadcast packet by "itself" if the replacement path from the source can be found for each of its neighbors. Nodes in each replacement can be either forwarded nodes or nodes with higher priorities. In the neighbor-designating broadcasting approaches, a node can determine its neighbor's forwarding/non-forwarding status, that is, a forward node selected by its upstream sender updates its view when it receives the packet and determines its status and its neighbors' status consequently. In the hybrid approaches, both self-pruning and neighbor-designating methods are applied to determine a node's status. The marking process discussed in the last section is an example of self-pruning, and MPR is an example of neighbor-designating.

Figure 3 shows the coverage condition that applies to the self-pruning and neighbor-

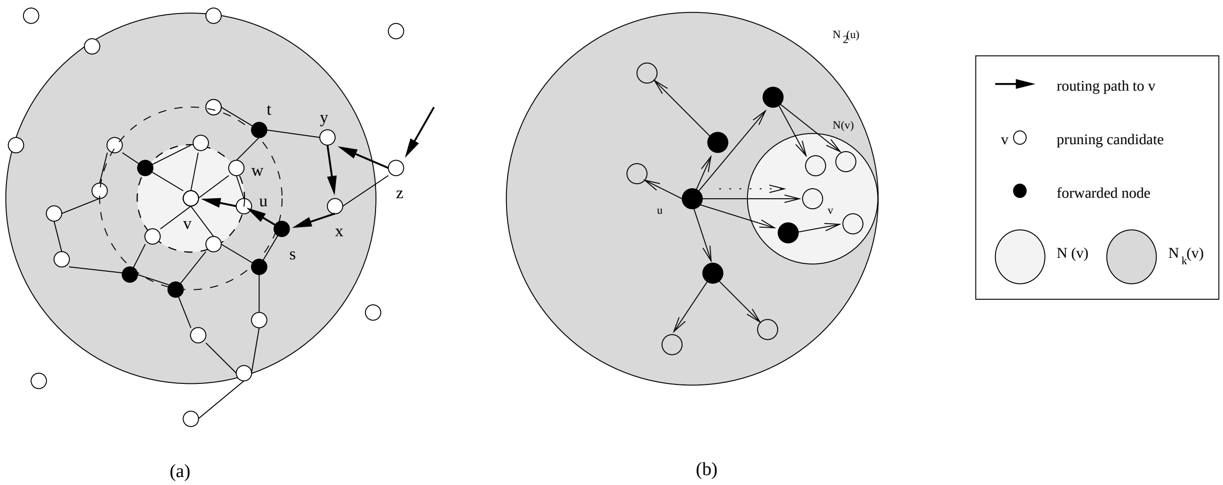

Fig. 3. Coverage condition for (a) self-pruning algorithm and (b) neighbor-designating algorithm.

designating approaches. In Figure 3 (a), suppose v has received a packet from u. If v's 1-hop neighbors are all covered by a set of forwarded nodes (black) or some white nodes with higher priorities, v can drop its role to relay the packet. Figure 3 (b) shows how the neighbor-designating approach works. 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) coverage nodes.

In the subsequent discussion, we only consider the neighbor-designating algorithms that utilize local information. The following assumptions are also used: (1) The transition is error free; 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 [34], [42], [43]. (3) Network topology is a connected graph without unidirectional links. A sub-layer can be added [40], [46] 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 can ensure full coverage, except a special mobility management mechanism is used, such as the one in [50]

III. Neighbor-Designating Broadcast Algorithms

In this section, we describe some algorithms which belong to the category of neighbor-designating broadcast. Each algorithm adopts the heuristic strategy where a minimum number of forward nodes are selected so that other neighbors can take the non-forward status.

A. Forward Node Selection Process

Theoretically, a MANET is represented as a unit disk graph $G(t)$ =(V, E), where the node set V represents a set of wireless mobile nodes and the edge set E represents a set of bi-directional links between the neighboring nodes. Two nodes are considered neighbors if and only if their geographic distance is less than the transmission range r. We use $N_k(v)$ to represent the k-hop neighbor set of v, where nodes in the set are no more than k hops further from v. $N_k(v)$ includes v itself. ($N_1(v)$, 1-hop neighbor set, can be simply represented as $N(v)$.) u's *k-hop node set* $H_k(u)$ consists of all nodes that are exactly k hops away from u. $N_k(u)$ and $H_k(u)$ have the following relationships: (1) $N_k(u) = N_{k-1}(u) \cup H_k(u)$; (2) $N_{k-1}(u) \cap H_k(u) = \phi$. If S is a node set, $N(S)$ is the union of the neighbor sets of every node in S, that is, $N(S) = \cup_{\forall w \in S} N(w)$.

For the instance a node u broadcasts a packet, u selects a subset from X to cover U by using the greedy algorithm in the set coverage problem [29]. The greedy algorithm, called forward node set selection process, works as follows:

1. Each node w in X calculates its effective node degree $deg_e(w) = |N(w) \cap U|$.
2. 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 U. A tie is broken by using node ID.
3. If U 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.
4. Repeat steps 2 and 3 until U becomes empty.
5. The node set $\{w_1, w_2, ...\}$ forms a forward node set.

B. Multipoint Relays

Qayyum et al [39] proposed selected multipoint relays (MPRs) as forward nodes to propagate link state messages in their optimized link state routing (OLSR) protocol. The

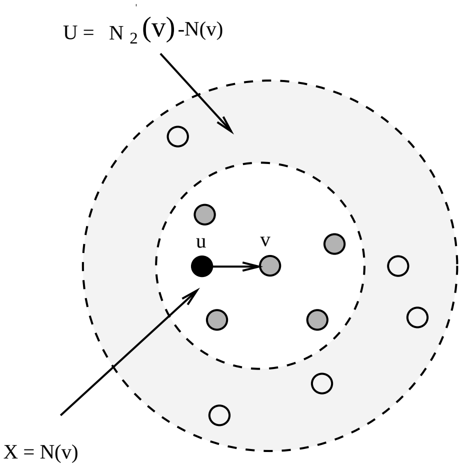

Fig. 4. Illustration of multiple relays (MPR).

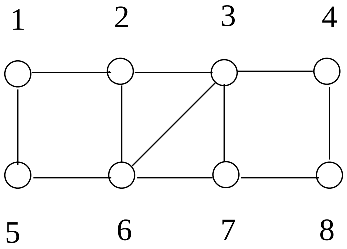

Fig. 5. A sample network.

MPRs are selected from 1-hop neighbors to cover the entire set of 2-hop neighbors. (Figure 4). The MPRs are selected as follows: If u intends to forward a packet, u uses the forward node set selection process to select its forward node set from $X = N(u)$ to cover 2-hop neighbors in $U = N_2(u)$. The selected forward nodes become MPRs.

In the sample network shown in Figure 5, when node 3 uses MPR algorithm, $U = N_2(3)=\{1, 2, 3, 4, 5, 6, 7, 8\}$ and $X = N(3)=\{2, 3, 4, 6, 7\}$, node 3 selects nodes $2, 4$ and 6 as its forward nodes.

When sending a broadcast packet, each selected MPRs runs a restricted forward node selection process applied in [19]: if an MPR v first receives a broadcast packet from a neighbor that does not designate v as a forward node, v does not forward this packet even if v may be selected as a forward node later by another neighbor.

In Figure 5, node 3 selects nodes $2, 4$ and 6 as MPRs; node 6 selects nodes $2, 3$ and 7 as MPRs. Suppose node 2 starts a broadcast, node 7 may receive the broadcast packet from both nodes 3 and 6. If node 7 first receives a packet from node 3, and then receives a duplicated broadcast packet from node 6, then because node 3 does not designate node 7 to forward the broadcast, node 7 will not forward the broadcast packet later when it receives the same broadcast packet from node 6 and has been designated to forward the packet by node 6 as an MPR. On the other hand, if node 7 first receives a packet from node 6, node 7 will forward the broadcast packet.

In [1], the requirement for an MPR to forward the packet is more restricted. An MPR will become a forward node if its node ID is smaller than all its neighbors or if it is selected as an MPR by its neighbor with the minimal node ID.

In the same network shown in Figure 5, when applying the rule proposed in [1], node 7 will never forward a broadcast packet whether the packet is from node 3 or 6 since node 3, which is the neighbor of node 7 with the smallest node ID, does not select node 7 as an MPR.

In [47], Wu further extends the algorithm in [1] by introducing the concept of *free neighbor*. Node u is a free neighbor of v if v is not the neighbor of u with the smallest node ID. The neighbors of free neighbors can be removed from the 2-hop neighbors of a node before it uses the forward node selection algorithm to designate its MPRs. Another extension rule proposed in [47] is that a node which has a smaller ID than all its neighbors and also has two unconnected neighbors becomes a forward node. This extension rule will be more effective in relatively sparse networks.

In Figure 5, we investigate node 3's neighbors. Nodes 2 and 6 are node 3's free neighbors, therefore, $N(2)$=$\{1, 2, 3, 6\}$ and $N(6)$=$\{2, 3, 5, 6, 7\}$ can be removed from $N_2(3)$. After that, node 6 only selects node 4 as its forward node. For node 7, since its free neighbors are 6 and 8 and $N_2(7) - N(6) - N(8)$ are empty, node 7 will not select any forward node.

C. Dominant Pruning

Lim and Kim [22] provided a dominant pruning algorithm (DP). Unlike the MPR, the DP excludes the neighbors of the upstream forwarded node from the current node's 2-hop neighbor set. In Figure 6, suppose u sends a packet and u selects v as its forward node. Since the neighbor set of u, $N(u)$, has been covered by u and the neighbor set of v, $N(v)$, will be covered by v, v does not need to select other forward nodes to cover

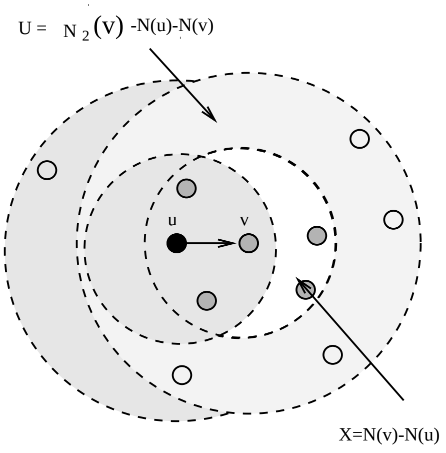

Fig. 6. Illustration of dominant pruning (DP).

them again. Therefore, v can determine its forward node set from $X = N(v)$ to cover $U{=}N_2(v) - N(u) - N(v)$. The forward node selection process from X to cover U is the same as above.

For the sample network shown in Figure 5, suppose node 3 receives a packet from node 6. When node 3 uses DP algorithm, $U{=}N_2(3) - N(6) - N(3){=}\{1, 8\}$ and $X = N(2){=}\{2, 4, 6, 7\}$. Node 3 selects node 2 and 4 as its forward nodes.

Peng and Lu [36] proposed an ad hoc broadcast protocol (AHBP) algorithm similar to the DP. In their algorithm, forward nodes are called BRGs (Broadcast Relay Gateways). BRGs, using the same forward node selection process to determine their downstream BRGs, will forward the broadcast packet and inform their designated BRGs. The AHBP considers the case of the mobility of the node. When v receives a packet from u that is not listed in its neighbor set, v assumes itself to be a designated BRG and rebroadcasts the packet.

D. Total Dominant Pruning and Partial Dominant Pruning

Lou and Wu [25] proposed total dominant pruning (TDP) and partial dominant pruning (PDP) to extend the DP by further reducing the number of 2-hop neighbors that need covered by 1-hop neighbors. The TDP requires the upstream forwarded node u piggyback $N_2(u)$ along with the broadcast packet. With this information, the selected forward node v can remove $N_2(u)$, instead of $N(u)$ in DP, from $N_2(v)$ (Figure 7), that is, $U{=}\ N_2(v){-}N_2(u)$.

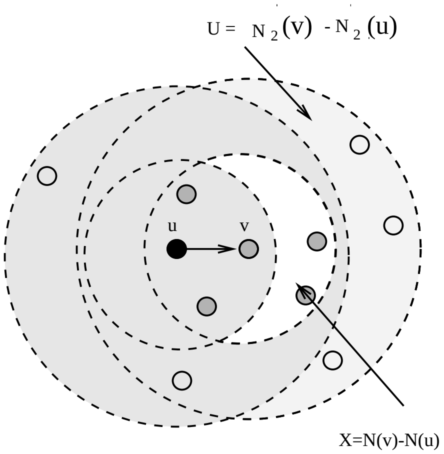

Fig. 7. Illustration of total dominant pruning (TDP).

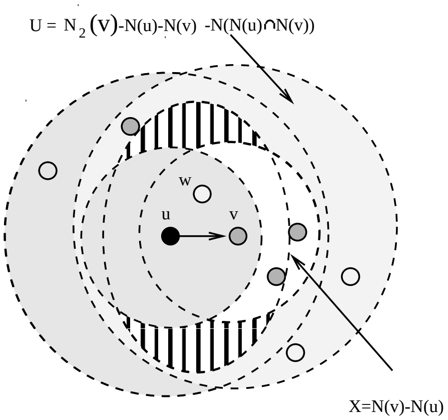

Fig. 8. Illustration of partial dominant pruning (PDP).

X can also update to $U(u) - N(v)$. In the PDP, the broadcast packet does not attach the upstream forwarded node's 2-hop neighbor set. When receiving the packet from node u, node v extracts the neighbors of the common neighbors of u and v (i.e., neighbors of nodes in $N(u) \cap N(v)$) from $N_2(v)$ since these nodes are covered by u's forward node set $F(u)$. Therefore, the uncovered 2-hop neighbor set U becomes $N_2(v) - N(u) - N(v) - N(N(u) \cap N(v))$ (Figure 8). The selection process for both TDP and PDP is the same as before, that is , selecting forward nodes from X to cover U.

For the sample network shown in Figure 5, suppose node 3 receives a packet from node 6. When node 3 uses TDP algorithm, $U = N_2(3) - N_2(6) = \phi$ and $X = N(3) - N(6) = \{4\}$. Node 3 does not select any node as its forward node. When node 3 uses PDP algorithm,

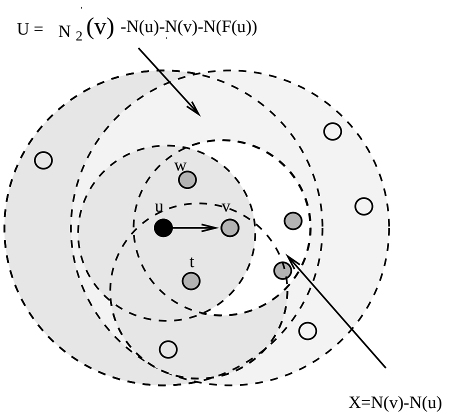

Fig. 9. Illustration of CDS-based broadcasting (CDSB).

$U = N_2(3) - N(6) - N(3) - N(N(3) \cap N(6)) = \{8\}$ and $X = N(3) - N(2) = \{4\}$. Node 3 selects node 4 as its forward node.

E. CDS-based Broadcast Algorithm

Peng and Lu [35] proposed a CDS-based broadcast algorithm (CDSB). It considers not only the sender of the broadcast packet but also the forward nodes with lower node IDs that are selected by the sender to determine a selected forward node's forward node set. For a sender u, suppose u selects nodes t, v, w $(id(t) < id(v) < id(w))$ as its forward nodes. When nodes t, v, w receive the packet, t updates its uncovered 2-hop neighbor set $U(t) = N_2(t) - N(u) - N(t)$; v updates its uncovered 2-hop neighbor set $U(v) = N_2(v) - N(u) - N(t) - N(v)$ because $N(t)$ is covered by t. Likewise, w's uncovered 2-hop neighbor set is $U(w) = N_2(w) - N(u) - N(t) - N(v) - N(w)$ (Figure 9). Notice that v may not forward the packet if $U(v)$ is empty.

For the sample network shown in Figure 5, suppose node 6 is the source, it selects nodes $2, 3$ and 7 as forward nodes. The forward nodes are piggybacked with the broadcast packet. When 2 receives a packet from node 6, it updates $U(2) = N_2(2) - N(6) - N(2) = \{4\}$ and $X(3) = N(2) = \{1, 2, 3, 6\}$. Node 2 selects node 3 as its forward node. When 3 receives the packet from node 6, it updates $U(3) = N_2(3) - N(6) - N(3) - N(2) = \{8\}$ and $X(3) = N(3) = \{2, 3, 4, 6, 7\}$. Node 3 selects node 4 as its forward node. When 7 receives the packet from node 6, it updates $U(7) = N_2(7) - N(6) - N(7) - N(2) - N(3) = \phi$ and $X(7) = N(7) = \{3, 6, 7, 8\}$. Node 7 does not select any node as its forward node.

IV. Other Extensions

We describe three extensions that also applied the neighbor-designating approach for broadcasting. The first broadcast approach is based on the cluster structure, the second is a generic K-hop zone-based algorithm, and the third considers the reliability issue.

A. Cluster-Based Broadcast Algorithm

Although cluster-based broadcast algorithms are not local algorithms, they usually work well with local state information and low time delay. Basically, the clustered network converts any "dense" network to a "sparse" one consisting of clusterheads only since clusterheads form a DS of the network. Moreover, clusterheads and gateways form a CDS of the network. Therefore, this is enough to fulfill a broadcast operation when all clusterheads and gateways forward a broadcast packet.

Alzoubi et al [2] proposed a cluster-based message-optimal CDS algorithm. In this algorithm, a CDS is constructed locally in a constant approximation ratio with message complexity in $O(n)$, where n is the size of the network. This message complexity reaches the lower bound in the network with n nodes. It was formed with two steps: In the first step, clusterheads are determined by the lowest-ID clustering algorithm. A clusterhead knows all its 2-hop and 3-hop clusterheads with two rounds of neighborhood information exchanges. In the second step, each clusterhead selects a node to connect each 2-hop clusterhead and a pair of nodes to connect each 3-hop clusterhead. All the clusterheads and selected nodes form a CDS of the network.

A cluster-based broadcast algorithm is proposed in [24] based on the coverage of the neighbor set. A clusterhead v's *coverage set* $C(v)$ is a set of clusterheads that are in a specific coverage area of v. It can be a *3-hop coverage set*, which includes all the clusterheads in $N_3(v)$, or a *2.5-hop coverage set*, which includes all the clusterheads in $N_2(v)$ and the clusterheads that have members in $N_2(v)$. Figure 10 illustrates a clusterhead v's 3-hop and 2.5-hop coverage sets. In this network, the clusterhead of c' is in v's 3-hop coverage set, but not in v's 2.5-hop coverage set. In general, the size of a clusterhead's 2.5 hop coverage set is less than that of its 3 hop coverage set. Therefore, the cost of maintaining the 2.5-hop coverage set can be less than that of the 3-hop coverage set.

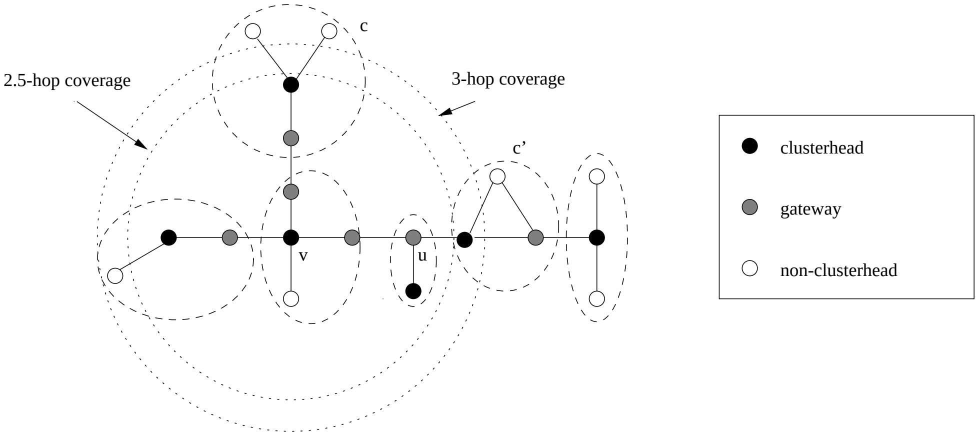

Fig. 10. 3-hop and 2.5-hop coverage area

Each clusterhead gathers the information of its coverage set by exchanging neighborhood information with its neighbors. Figure 11(a) illustrates construction of a coverage set with 3-hop and 2.5-hop coverage set. For 3-hop coverage set, node 6 sends a message M1 which contains its 1-hop clusterhead neighbors: $CH\_HOP1(6) = \{2*\}$ and 8 sends a message M2: $CH\_HOP1(8) = \{2*, 3\}$. Here, * indicates the clusterhead of the cluster that the node belongs to. Likewise, nodes 5 and 7 send messages M3,M5: $CH\_HOP1(5) = \{1*\}$ and $CH\_HOP1(7) = \{1, 4*\}$. After receiving M1 and M2, node 7 may form a message M6 which contains its 2-hop clusterhead neighbors and associated gateways: If node 6 is selected, $CH\_HOP2(7) = \{2[6], 3[8]\}$; if node 8 is selected, $CH\_HOP2(7) = \{2[8], 3[8]\}$. Here, $CH\_HOP2(u) = \{v[w], ...\}$ means that clusterhead u connects to clusterhead v via w. Node 5 also sends message M4: $CH\_HOP2(5) = \{2[6]\}$. By receiving M3, M4, M5, M6, node 1 can build its local view as Figure 11(b) or (c). For the 2.5 hop coverage set, node 1 builds its local view as Figure 11(d). Here, clusterhead 3 is unknown to node 1.

The clusterhead needs to select forward nodes to connect each clusterhead in its coverage set. In [2], each clusterhead randomly selects one or two nodes to connect its adjacent clusterheads. In [26], a greedy algorithm similar to forward node selection process is applied when a clusterhead v receives a broadcast packet p from its upstream clusterhead u. Suppose p is a new packet for v and p also attaches u's forward node set $F(u)$ and u's coverage set $C(u)$. v can update $C(v) = C(v) - C(u) - \{u\}$ because all the clusterheads in $C(u) \cup \{u\}$ are covered by $F(u)$, and they do not need to be covered again. v selects the

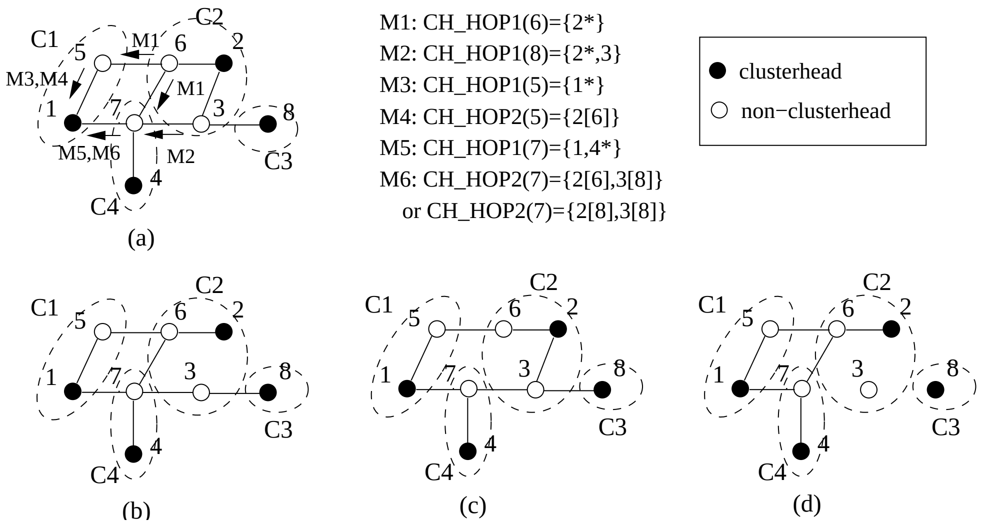

Fig. 11. Illustration of constructing a coverage set

gateway that connected the maximum number of clusterheads in $C(v)$, puts the gateway into $F(v)$, removes the connected clusterheads from $C(v)$; This process repeats until $C(v)$ become empty. The selection process works for both 3-hop and 2.5-hop coverage sets. When the 2.5-hop coverage set is used, $F(u)$ may cover some extra clusterheads in addition to $C(u) \cup \{u\}$, that is, if clusterhead v is 3 hops away from u, and u uses a path (u, f, r, v) to deliver the broadcast packet to v, clusterheads in $N(r)$ also receive the broadcast packet. These clusterheads can also be excluded from $C(v)$. Therefore, the updated $C(v)$ $= C(v) - C(u) - \{u\} - N(r)$.

The cluster-based broadcast algorithm consists of the following steps: (1) If the source is not a clusterhead, it sends the broadcast packet p to its clusterhead. (2) When a clusterhead receives p from its upstream clusterhead sender for the first time, it uses the above forward node selection process to choose forward nodes that cover all the clusterheads in its coverage set. The coverage set of this clusterhead, together with its selected forward nodes, are piggybacked with p for the forwarding purpose. (3) For a non-clusterhead node that receives p for the first time, if it is a forward node, it relays p; otherwise, it does nothing.

Figure 12 illustrates the broadcast process in a cluster-based CDS with 2.5-hop coverage set. Suppose the source is node 1. Node 1's 2.5-hop coverage set $C(1)$ is $\{2, 3\}$, it selects

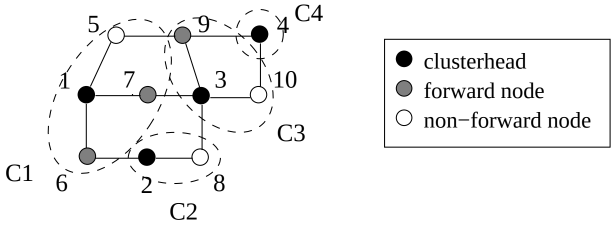

Fig. 12. Illustration of cluster-based broadcast with 2.5-hop coverage set.

nodes 6 and 7 to forward the packet to clusterheads 2 and 3. The broadcast packet piggybacks the forward node set $F(1) = \{6, 7\}$ and the 2.5-hop coverage set $C(1) = \{2, 3\}$. When clusterhead 2 receives the broadcast packet from clusterhead 1, it updates $C(2) = C(2) - C(1) - \{1\} = \{1, 3\} - \{2, 3\} - \{1\} = \phi$; then, it only locally broadcasts the packet. When clusterhead 3 receives the packet from clusterhead 1, it updates $C(3) = C(3) - C(1) - \{1\} = \{1, 2, 4\} - \{2, 3\} - \{1\} = \{4\}$; therefore, clusterhead 3 selects node 9 to forward the packet to clusterhead 4. $F(3) = \{9\}$ and $C(3) = \{1, 2, 4\}$ are piggybacked with the packet. After clusterhead 4 receives the packet, it only locally broadcasts the packet since all clusterheads in $C(4)$ have received the packet. In total, 7 nodes (nodes 1, 2, 3, 4, 6, 7 and 9) will forward the packets.

B. K*-hop Zone-Based Algorithm*

A node's *K-hop zone* consists of all the nodes within K hops from the given node. For a given network, K can be set from 0 to the diameter of the network. One extreme case is $K = 0$, that is, nodes in the network have no neighbor set information, or the neighbor set information is most likely out-of-date most of the time because of the high mobility of the nodes. The only possible strategy for routing is flooding. The other extreme case is that K is equal to the diameter of the network, that is, each node knows the global information of the network. Therefore, the optimum solution can be determined in this circumstance.

A K-hop cluster-based algorithm is proposed in [21]: Each node gets its K-hop neighbor set information. A cluster is composed of all nodes within K hops from a given node. Each node belongs to one cluster. When a broadcast occurs, only border nodes, which are exactly K hops away from the sender, will relay the broadcast. A similar connectivity-

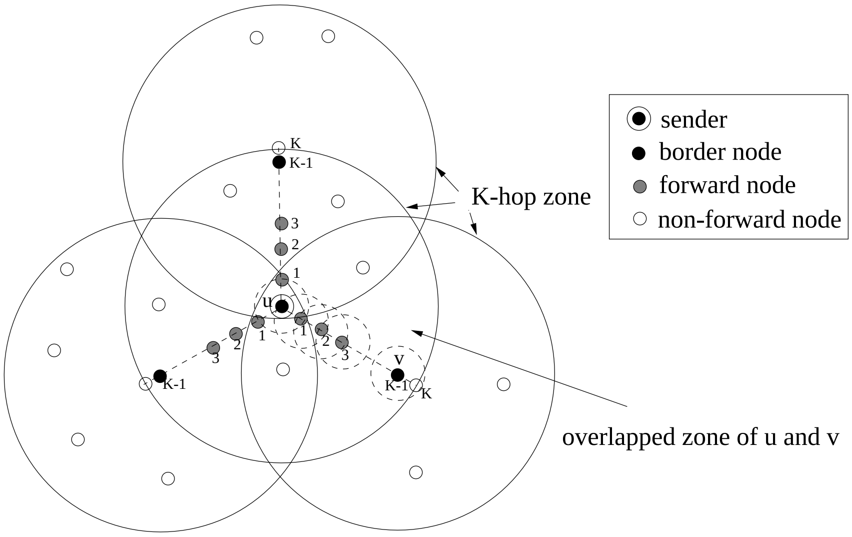

Fig. 13. Illustration of the K-hop zone-based broadcast protocol.

based K-hop clustering algorithm is proposed in [6]. One main disadvantage of these algorithms is that the overlapped area of two neighboring sender's K-hop neighbor sets cause much redundancy when K is large.

Lou and Wu [28] proposed a generic K-hop zone-based broadcast protocol. A generic K-hop zone-based broadcast protocol broadcasts a packet in four steps: (1) The sender uses the forward node set selection process to select its forward node set to cover its K-hop zone. It broadcasts the packet which attaches the forward node set. (2) A forward node (excluding the border nodes) forwards the packet when it first receives the packet. (3) A non-forward node receives the packet but does not forward it. (4) A border node, which is a selected forward node exactly $K-1$ hops away from the source, after receiving the packet, becomes a new sender to disseminate the packet. Nodes that are within $K-1$ hops of the sender can be excluded from the K-hop zone of the border node. In addition, nodes that are within $K-1$ hops of a border node with lower ID can also be excluded from the K-hop zone of a border node with higher ID if the two border nodes are within $K-1$ hops. Fig. 13 illustrates the K-hop zone-based broadcast protocol. The source u selects forward nodes to cover its K-hop zone. Gray nodes are forward nodes that just relay the broadcast packet. Black nodes are the border nodes that have their own K-hop zones. White nodes are the non-forward nodes that only receive the packet.

The forward node set selection process is described as follows: A sender u computes its forward node set to cover all the nodes in its K-hop zone: In each iteration, u selects some nodes in $H_k(u)$ to cover all the nodes in $H_{k+1}(u)$, where $0 \leq k \leq K-1$. Specifically, u itself covers all nodes in $H_1(u)$; some selected nodes in $H_1(u)$ cover all nodes in $H_2(u)$,..., until some selected nodes in $H_{K-1}(u)$ cover all nodes in $H_K(u)$. In each iteration, the selection criterion is that the node with the maximum number of uncovered neighbors is selected first. A tie is broken by node ID. All selected nodes form u's forward node set $F(u)$.

For the sample network shown in Figure 5, suppose node 1 is the source and it has 3-hop neighbor set information. $H_0(1) = \{1\}$, $H_1(1) = \{2, 5\}$, $H_2(1) = \{3, 6\}$ and $H_3(1)$ $= \{4, 7\}$. Node 1 covers $N_1(1)$. Nodes 2, 5 are first selected from $H_1(1)$ to cover all nodes in $H_2(1)$; then node 3 is selected to cover nodes in $H_3(1)$. Therefore, $F(1)$=$\{2, 3, 5\}$ and node 3 is a border node. Then node 3 becomes a new sender. Among nodes in node 3's 3-hop zone, node 4 will be selected to cover node 8.

C. Reliable Broadcast Algorithm

The traditional reliable broadcast protocols can be classified into two categories. The first category enforces strong reliability guarantees which provide an atomic operation for the successful delivery of a message to all the nodes [16]. The disadvantage is its poor scalability even in a very stable network. The second category is based on the feedback mechanisms of acknowledgement (ACK) or negative acknowledgement (NACK). It can also be classified as *sender initiated* and *receiver initiated* approaches [38]. In the sender initiated approach, the receiver acknowledges each message it receives. The ACKs are unicasted to the sender who maintains all the records for all receivers to confirm the success of the delivery. Only missing packets are retransmitted, either to individual requested receivers, or to all receivers. However, the requirement of sending ACKs in response to the reception of a packet for all receivers may cause channel congestion and packet collision, which is called ACK *implosion* [18]. Moreover, the amount of records that the sender must maintain to track the receiver set may also grow large. In the receiver initiated approach, the receiver is responsible for reliable delivery. Each receiver maintains reception records and requests repairs via a $NACK$ when errors happen. Several strategies can be applied for the receiver initiated approaches, such as sender-oriented, flat-receiver-oriented and hierarchical-receiver-oriented approaches. The problem of the receiver initiated approach is the long end-to-end delay since the sender must wait for the next broadcast packet to determine if the previous one is successfully delivered or not. Therefore, it can be applied only when the sender has many packets to be sent.

Lou and Wu [27] proposed a reliable broadcast algorithm, called *double-covered broadcast algorithm*, which aims to reduce broadcast redundancy by decreasing the number of forward nodes but still providing high delivery ratio for each broadcast packet in a dynamic environment. The algorithm utilizes the method in which the sender overhears the retransmission of the forward nodes to avoid the ACK implosion problem. Also, the algorithm guarantees that each node is covered by at least two transmissions so that it can avoid a single error due to the transmission collision. Moreover, the algorithm does not suffer the disadvantage of the receiver-initiated approach which needs a much longer delay to detect a missed packet.

The double-covered broadcast algorithm works as follows: When a sender broadcasts a packet, it selects a subset of 1-hop neighbors as its forward nodes to forward the broadcast based on a greedy approach. The selected forward nodes satisfy two requirements: (1) They cover all the nodes within 2 hops of the sender. (2) The sender's 1-hop neighbors are either forward nodes or non-forward nodes but covered by at least two neighbors, once by the sender itself and once by one of the selected forward nodes. After receiving the broadcast packet, each forward node records the packet, computes its forward nodes and re-broadcasts the packet as a new sender. The retransmissions of the forward nodes are received by the sender as the acknowledgement of receiving the packet. The non-forward 1-hop neighbors of the sender do not acknowledge receipt of the broadcast. The sender waits for a predefined duration to overhear the rebroadcasting from its forward nodes. If the sender fails to detect all its forward nodes retransmitting during this duration, it assumes that a transmission failure has occurred for this broadcast because of the transmission error or because the missed forward nodes are out of its transmission range. The sender then re-sends the packet until all forward nodes are retransmitted or the maximum number of retries is reached.

Like most of the other neighbor designating algorithms, the DCB supposes each node

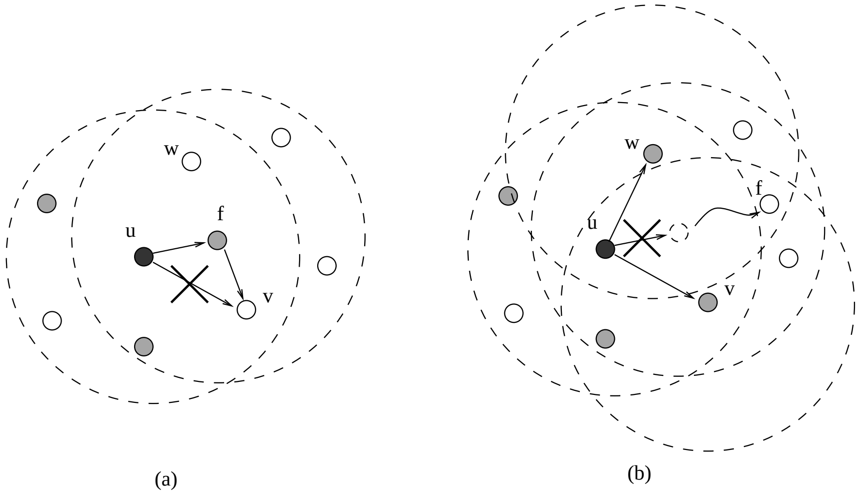

Fig. 14. An illustration of transmission errors: (1) a transmission error occurs at a non-forward node n. (2) alternative forward nodes m and n are selected to cover the area that is supposed to be covered by the missed forward node f.

knows its 1-hop and 2-hop neighbor sets $N(v)$ and $N_2(v)$. The forward node set selection process executes at each forward node to determine its own forward node set. We consider the two cases where a node v determines its forward node set $F(v)$: (1) v is the source of the broadcast: v uses FNSSP algorithm to find $F(v)$ in $X = N(v)$ to cover $U = N_2(v)$. (2) v is a selected forward node to relay the broadcast packet: Suppose v has already received the packet from a node set $V(v)$ and each node w in $V(v)$ has its own forward node set $F(w)$. v uses FNSSP algorithm to find $F(v)$ in $X = N(v) - V(v) - \cup_{\forall w \in V(v)} F(w)$ to cover $U = N_2(v) - N(V(v)) - \cup_{\forall w \in V(v)} N(F(w))$.

For the sample network in Figure 5, we uses the DCB algorithm to select each node's forward node set. In case 1, we suppose node 3 is the source, then node 3 selects nodes $2, 4$ and 6 as its forward nodes. In case 2, we consider node 6 when it receives a packet from node 3. $U(6) = N_2(6) - N(3) - N(2) - N(4) = \{5\}$, node 6 selects node 5 as its forward node.

A node may fail to receive the broadcast packet from its neighbors because of a transmission collision with other neighbors, the high transmission error rate of the radio channel, or the out-of-range movement of the node. Each non-forward node has been at least covered by two forwarding nodes; even if the non-forward node missed the packet from one, it still has a second chance to receive the packet from the other one (Figure 14 (a)). The reception of a forward node needs to be confirmed since forward nodes are the key nodes in the network that need to relay the broadcast packet. The loss of the reception may cause the transmission error to propagate. If the sender does not detect the forward node's retransmission signal, the sender will select alternative forward nodes to cover the coverage area of the missed forward node and resend the broadcast packet until it receives the confirmation from its forward nodes or the maximum times of retries is reached (Figure 14 (b)).

V. Summary

In this chapter, we described several broadcast algorithms that use the neighbor-designating approach. More complicated algorithms based on neighbor designating approach are also introduced. The forward node selection process is the basis for all the broadcast algorithms discussed here. The optimal solution is NP-complete and more theoretical discussion about the optimal MCDS solution can be found in [2], [14], [30], [45]. The algorithm discussed above are all heuristic algorithms. Except for the cluster-based broadcast algorithm, all other algorithms mentioned here have no constant approximate ratio to the optimal solution. Therefore, they will show poor performance for broadcast operation when the network becomes extremely dense. Recently, some efforts, such as [48], have been made to extend various local algorithms to be effectively used in dense networks.

References

[1] C. Adjih, P. Jacquet, and L. Viennot. Computing connected dominated sets with multipoint relays. *http://www.inria.fr/rrrt/rr-4597.html*, 2002.

[2] K. M. Alzoubi, P. J. Wan, and O. Frieder. Message-optimal connected dominating sets in mobile ad hoc networks. *Proc. of 3rd ACM Int'l Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC'2002)*, pages 157–164, 2002.

[3] K. M. Alzoubi, P. J. Wan, and O. Frieder. New distributed algorithm for connected dominating set in wireless ad hoc networks. *Proc. of 35th Hawaii Int'l Conf. on System Sciences (HICSS-35)*, pages 3881–3887, Jan. 2002.

[4] G. Calinescu, I. Mandoiu, P. J. Wan, and A. Zelikovsky. Selecting forwarding neighbors in wireless ad hoc networks. *Proc. of ACM DIALM'2001*, pages 34–43, Dec. 2001.

[5] J. Cartigny and D. Simplot. Border node retransmission based probabilistic broadcast protocols in ad-hoc networks. *Proc. of HICSS-36*, Jan. 2003.

[6] G. Chen, F. G. Nocetti, J. S. Gonzalez, and I. Stojmenovic. Connectivity based k-hop clustering in wireless networks. *Proc. of 35th Hawaii Int'l Conf. on System Sciences (HICSS-35)*, Jan. 2002.

[7] C. C. Chiang and M. Gerla. Routing and multicast in multihop, mobile wireless networks. *Proc. of IEEE Int'l Conf. on Universal Personal Communications*, 2:546–551, 1997.

[8] C. C. Chiang, H. K. Wu, W. Liu, and M. Gerla. Routing in clustered multihop, mobile wireless networks with fading channel. *Proc. of IEEE Singapore Int'l Conf. on Networks*, pages 197–211, 1996.

[9] F. Dai and J. Wu. Distributed dominant pruning in ad hoc wireless networks. *Proc. of IEEE 2003 International Conference on Communications (ICC 2003)*, 1:353–357, May 2003.

[10] B. Das, R. Sivakumar, and V. Bharghavan. Routing in ad-hoc networks using a spine. *Proc. of the 6th Int'l Conf. on Computer communications and Networks (ICCCN'97)*, pages 1–20, Sept. 1997.

[11] A. Ephremides, J. E. Wieselthier, and D. J. Baker. A design concept for reliable mobile radio networks with frequency hopping signaling. *Proc. of the IEEE*, 75(1):56–73, 1987.

[12] M. Gerla and J. Tsai. Multicluster, mobile, multimedia radio network. *ACM/Baltzer Wireless Networks*, 1:255–265, 1995.

[13] IEFT MANET Working Group. http://www.ietf.org/html.charters/manet-charter.html.

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

[15] Z. J. Haas, J. Y. Halpern, and L. Li. Gossip-based ad hoc routing. *Proc. of INFOCOM 2002*, 3:1707–1716, June 2002.

[16] V. Hadzilacos and S. Toueg. *Fault-tolerant broadcasts and related problems*, chapter Distributed System, pages 97–145. Addison-Wesley, 2 edition, 1993.

[17] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath. Flooding for reliable multicast in multi-hop ad hoc networks. *Proc. of ACM DIALM'99*, pages 64–71, Aug. 1999.

[18] M. Impett, M. S. Corson, and V. Park. A receiver-oriented approach to reliable broadcast ad hoc networks. *Proc. of Wireless Communications and Networking Conference (WCNC'2000)*, 1:117–122, 2000.

[19] P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot. Optimized link state routing protocol. draft-ietf-manet-olsr-07.txt, Nov. 2002.

[20] D. B. Johnson and D. A. Maltz. *Mobile Computing*, chapter Dynamic source routing in ad-hoc wireless networks, pages 153–181. Kluwer Academic Publishers, Boston, MA, Imielinski and Korth edition, 1996.

[21] D. Kim, S. Ha, and Y. Choi. K-hop cluster-based dynamic source routing in wireless ad-hoc packet radio network. *Proc. of VTC'98*, pages 224–228, 1998.

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

[23] C. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 15:151–162, 1999.

[24] W. Lou and J. Wu. Efficient broadcast with forward node set in clustered mobile ad hoc networks. *Proc. of the 11th Int'l Conf. on Computer Communications and Networks (ICCCN'2002)*, pages 398–403, Oct. 2002.

[25] W. Lou and J. Wu. On reducing broadcast redundancy in ad hoc wireless networks. *IEEE Trans. on Mobile Computing*, 1(2):111–123, April-June 2002.

[26] W. Lou and J. Wu. A cluster-based backbone infrastructure for broadcasting in manets. *Proc. of IPDPS'2003, Workshop of WMAN*, April 2003.

[27] W. Lou and J. Wu. A double-covered broadcast algorithm in mobile ad hoc networks. *accepted to appear in Proc. of IEEE Global Telecommunications Conference (GLOBECOM 2003)*, 2003.

[28] W. Lou and J. Wu. A k-hop zone-based broadcast protocol in mobile ad hoc networks. 2003. Under preparation.

[29] L. Lovasz. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 13:383–390, 1975.

[30] M. V. Marathe, H. Breu, H. B. Hunt III, S. S. Ravi, and D. J. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, 25:59–68, 1995.

[31] S. Ni, Y. Tseng, Y. Chen, and J. Sheu. The broadcast storm problem in a mobile ad hoc network. *Proc. of ACM/IEEE MOBICOM'99*, pages 151–162, Aug. 1999.

[32] V. D. Park and M. S. Corson. Temporally-ordered routing algorithm (TORA) version 1: Functional specification. *Internet Draft*, 1997.

[33] M. R. Pearlman and Z. J. Haas. Determining the optimal configuration of the zone routing protocol. *IEEE Journal on Selected Areas in Communications*, 17(8):1395–1414, Feb. 1999.

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

[35] W. Peng and X. Lu. Efficient broadcast in mobile ad hoc networks using connected dominating sets. *Journal of Software*, 1999.

[36] W. Peng and X. Lu. AHBP: An efficient broadcast protocol for mobile and hoc networks. *Journal of Science and Technology*, 2002.

[37] C. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. *Proc. of 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, pages 90–100, Feb. 1999. New Orleans, LA.

[38] D. G. Petitt. Reliable multicast protocol design choices. *Proc. of MILCOM'97*, 1:242–246, 1997.

[39] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying for flooding broadcast message in mobile wireless networks. *Proc. of 35th Hawaii Int'l Conf. on System Sciences (HICSS-35)*, pages 3898–3907, Jan. 2002.

[40] 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.

[41] P. Sinha, R. Sivakumar, and V. Bharghavan. Enhancing ad hoc routing with dynamic virtual infrastructures. *Proc. of IEEE INFOCOM'2001*, 3:1763–1772, April 2001.

[42] 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.

[43] I. Stojmenovic, S. 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.

[44] Y.-C. Tseng, S.-Y. Ni, and E.-Y. Shih. Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network. *IEEE Trans. on Computers*, 52(5):545–557, May 2003.

[45] 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.

[46] 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.

[47] J. Wu. An enhanced approach to determine a small forward node set based on multipoint relay. *accepted to appear in VTC 2003*, 2003.

[48] J. Wu and F. Dai. Distributed formation of a virtual backbone in ad hoc networks using adjustaval transmission ranges. 2003. Under preparation.

[49] J. Wu and F. Dai. A generic distributed broadcast scheme in ad hoc wireless networks. *Proc. of ICDCS 2003*, pages 460–468, May 2003.

[50] J. Wu and F. Dai. Mobility management and its applications in efficient broadcasting in mobile ad hoc networks. 2003. Under preparation.

[51] J. Wu and H. Li. On calculating connected dominating sets for efficient routing in ad hoc wireless networks. *Proc. of ACM DIALM'99*, pages 7–14, Aug. 1999.

[52] J. Wu and W. Lou. Forward-node-set-based broadcast in clustered mobile ad hoc networks. *Wireless Networks and Mobile Computing, a Special Issue on Algorithmic, Geometric, Graph, Combinatorial, and Vector Aspects*, 3(2):155–173, 2003.