Forward-Node-Set-Based Broadcast in Clustered Mobile Ad Hoc Networks *

Jie Wu and Wei Lou
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431

Abstract

A taxonomy of broadcast protocols in mobile ad hoc networks (MANETs) is given where protocols are classified into four groups: global, quasi-global, quasi-local, and local. The taxonomy also divides protocols based on the nature of algorithms: probabilistic and deterministic. The locality of maintenance also plays an important role in evaluating the protocol. An important objective in designing a broadcast protocol is to reduce broadcast redundancy to save scarce resources such as energy and bandwidth and to avoid the broadcast storm problem. This objective should be achieved without introducing excessive overhead and time delay, measured by the sequential rounds of information exchanges. This is done by choosing a small forward node set that forms a connected dominating set (CDS) to carry out a broadcast process. In this paper, a clustered network model is proposed in which each node is a clusterhead in the clustered architecture. Clusterheads are connected by carefully selecting non-clusterhead nodes locally at each clusterhead to connect clusterheads within 2.5 hops, a novel notion proposed in this paper. Information of neighbor clusterheads are piggybacked with the broadcast packet to further reduce each forward node set. It is shown that this approach is quasi-local with locality of maintenance. In addition, this approach has a constant approximation ratio to the minimum connected dominating set (MCDS) and generates a small forward node set in the average case. Comparisons are also done through simulation with representative protocols from each of the four groups of protocols based on the proposed taxonomy.

Key words: Broadcast, cluster, dominating set, forward node set, MANET, taxonomy.

---

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

1 Introduction

A mobile ad hoc network (MANET) is a totally on-the-fly network used to support the notion of "any time, any place" service. It is an infrastructureless network that consists of a collection of wireless mobile hosts to form a temporary network without the aid of any centralized administration or standard support services [24]. In general, a MANET can be represented as a unit disk graph [6] $G = (V, E)$, where V represents a set of wireless mobile hosts (nodes) and E represents a set of links between the neighbors, assuming all hosts have the same transmission range r. Two hosts are neighbors if and only if they are within each other's transmission range. Therefore, the connections of hosts are based on geographic distances of hosts. The inherent limitations of the MANET, such as scarce resources and dynamic topologies, require any routing protocol designed for such an environment be simple, efficient and robust. The way that the packets are transmitted in the MANET is quite different from that in the wired network, because when a node sends a packet, all its neighbors will receive that packet under the promiscuous receive mode.

Broadcasting a packet to the entire network is a basic operation and has extensive applications in the MANET. For example, broadcasting is used in the route discovery process in several routing protocols [13, 20, 22], when advertising an error message to erase invalid routes from the routing table [19], or as an efficient mechanism for the reliable multicast in a fast-moving MANET [12]. In a broadcast process, the source and a subset of nodes form a *flood tree* such that any other node in the network is adjacent to a node in the tree. Nodes on the tree are called *forward nodes* and form a *connected dominating set* (CDS). In graph terminology, a *dominating set* (DS) is a subset of vertices such that every vertex in the graph is either in the set or has a link to a vertex in the set. If the vertices in a DS are also connected, it is called a connected dominating set (CDS). CDS can be used as a backbone of the network to support the broadcast process. Only nodes in the CDS forward the broadcast packet while nodes that are not in the CDS receive the broadcast packet without forwarding it. Finding a minimum CDS (MCDS) in a given graph is NP-complete and it has also been proved to be NP-complete in a unit disk graph [17]. Even when a minimum flooding tree is identified, maintaining such a tree in a mobile environment is a costly operation for practical use. The straightforward approach for broadcast is *blind flooding*, in which each node is obligated to re-broadcast the packet whenever it receives a packet for the first time. Blind flooding will generate many redundant transmissions. These redundant transmissions may cause a serious problem, referred as the *broadcast storm problem* [18], in which redundant packets cause communication congestion and contention.

Many broadcast algorithms have been proposed in [1, 8, 14, 18, 21, 23, 26, 28, 30]. However, different assumptions and models have been used, making the comparison among them difficult. In this paper, a taxonomy of broadcast protocols in the MANET is first given where protocols are classified into four groups: *global, quasi-global, quasi-local,* and *local.* The taxonomy also divides protocols based on the nature of algorithms: *probabilistic* and *deterministic.* The *locality of maintenance* also plays an important role in evaluating the protocol. The knowledge of additional information such as neighbor positions also affects the level of difficulty in designing an efficient protocol. An important objective in designing a broadcast protocol is to reduce broadcast redundancy to save scarce resources such as energy and bandwidth and to avoid the broadcast storm problem. That is done by determining a small forward node set. This objective should be achieved without excessive overhead and time delay, measured in terms of sequential rounds of information exchange. The measure of performance is based on either the worst case (in terms of approximation ratio to MCDS) or the average case (mainly through simulation).

Typical global [8, 27] and quasi-global [1] broadcast protocols use either global or partial global information to derive a small forward node set in both the worst and average cases. However, these protocols are costly in terms of the number of rounds of sequential information propagation needed to distribute state information (in the global approach) or to establish a global infrastructure such as a spanning tree (in the quasi-global approach). In addition, both protocols do not support locality of maintenance. Local broadcast protocols [14, 16, 30] use aggressive greedy approaches and are relatively efficient in obtaining a small forward node set in the average case. The merit of this approach is that it is totally localized and the flooding tree is constructed "on-the-fly" during the broadcast process and, hence, the maintenance cost is minimized. However, they cannot guarantee a reasonable bound of the forward node set in the worst case (e.g., without a constant approximation ratio).

In this paper, we study a quasi-local broadcast protocol based on the *clustered network* which is a 2-level hierarchical network. In each cluster, a clusterhead directly connects to all members. Basically, the clustered network converts any "dense" network to a "sparse" one consisting of clusterheads only. The proposed broadcast protocol consists of the following steps: First, the network is partitioned into clusters, each of which is controlled by a clusterhead. Then, each clusterhead applies a greedy approach to select its forward node set locally to cover all clusterheads within its vicinity (say within the 3-hop coverage area as shown in Figure 1 (a)). A non-clusterhead node just relays the broadcast packet if it is selected as a forward node or else it does nothing. The information about the clusterheads that will receive the broadcast packet from the forwarding clusterhead is also piggybacked along the broadcast packet so that the receiving clusterhead can further reduce its coverage area (this method is called *pruning technique*). After all the clusterheads

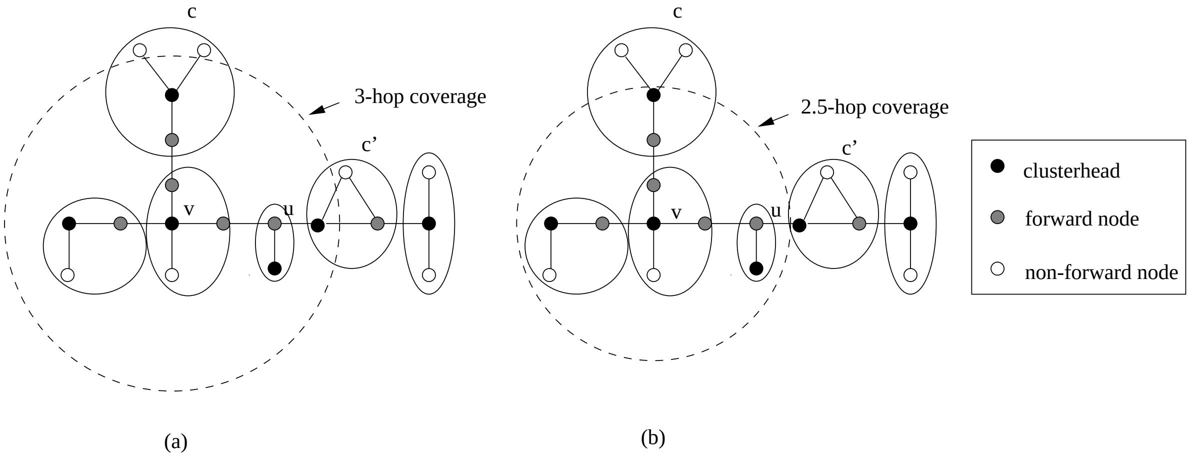


Figure 1: The coverage area of node v: (a) 3-hop and (b) 2.5-hop.

have received the packet and have forwarded it, the broadcast process terminates. The clustered network also ensures locality of maintenance when a slightly different cluster construction strategy is applied.

In addition to the good performance in the average case compared with other quasi-local and local protocols, we prove that the size of forward node set has a constant approximation ratio to the minimum connected dominant set (MCDS). A novel notion of 2.5-hop coverage is introduced where *each clusterhead just covers the clusterheads that have members within 2 hops.* That is, only partial 3-hop clusterheads need to be covered without sacrificing clusterhead connectivity. In Figure 1 (b), the clusterhead of c is covered by v, but not clusterhead of c'. This model extends a result by Chlamtac and Farago [5] where all 3-hop clusterheads need to be covered to ensure the connectivity of clusterheads. Simulation shows that the 2.5-hop coverage method has its comparable performance in terms of the size of the forward node set to the 3-hop coverage method when the pruning technique is used. Extensive simulation has been done to compare the proposed protocol with representative protocols from each of the four groups of protocols based on the proposed taxonomy. As differ from the traditional approximation model where only the worst case situation is considered, we provide a comprehensive analysis by including the average case situation as well.

The paper is organized as follows: A taxonomy of broadcast protocols in the MANET and related work are given in Section 2. Section 3 presents our protocol for the clustered network. In Section 4, we compare the performance of different protocols through simulations. Finally, Section 5 concludes the paper.

2 Taxonomy and Related Work

A taxonomy of broadcast protocols is given in the section, followed by a review of related work based on the taxonomy.

2.1 Taxonomy

We start with a taxonomy of broadcast protocols in the MANET. This taxonomy serves for two purposes: (1) to offer a platform of fair comparisons among protocols under different assumptions and models, and (2) to place our approach according to the taxonomy in order to clearly state its contributions and limitations. To have a clear focus, we limit our attention to two objectives:

1. A small set of forward nodes (or CDS).
2. Low overhead in both state information collection and maintenance and broadcast protocol itself.

The objective (1) can be measured either by the worst case or the average case in terms of *approximation ratio*. The measurement (2) is largely an uncharted field (except [1, 29] where traditional complexity models are used), partly because of the lack of a precise model for measurement. Mobility of hosts in MANETs adds another dimension of difficulty in the measurement. Here we use the *number of sequential rounds* needed to establish a CDS and any relevant infrastructure as a measurement for objective (2). We use the *locality of maintenance* to measure the adaptiveness of a protocol in the mobile environment. We believe that counting the number of communication operations alone is not sufficient. For example, it is much better to have 2-round synchronous information exchanges among n hosts (totally 2 synchronous rounds with $2n$ communication operations) than 1-round sequential information exchange among n hosts (totally n sequential rounds with n communication operations). In addition, communication operations (send and receive) are more expensive than computational operations (e.g., computational steps in a greedy algorithm without communication). Many local broadcast protocols require knowledge of 2-hop neighborhood information which can be collected through 2-round send operations at each host; that is, each host sends only two packets in two rounds. Although the second packet with a size of $O(\Delta)$ (Δ is the maximum node degree of the network) is larger than the first one $O(1)$, its size is insignificant when the graph is relatively sparse or a compression method is applied. Balancing objectives (1) and (2) is an art. Even within objective (1), the trade-off between the worst case and the average case is not a trivial task. The traditional approximation model has its limit, because it only focuses on the worst case (which rarely happens) and ignores the average case.

Our first classification is based on the type of algorithms used: *probabilistic* or *deterministic*. The probabilistic approach [5, 18] usually meets both objectives without introducing much overhead. However, with a low probability, one or more essential properties may not be met. For example, the generated DS is not guaranteed to be connected as in [5] or the network is not guaranteed to be covered by the selected forward nodes as in [18] (i.e., the forward nodes may not form a DS). The deterministic approach, however, guarantees all essential properties such as connectivity and coverage.

Our second classification is based on the amount of state information used in the algorithm: *global* or *local*. Note that 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 could be costly and this can be measured in terms of rounds. In the subsequent discussion, we focus only on deterministic solutions. Four types of broadcast protocols are discussed based on the second classification:

1. **Global**: broadcast protocol, centralized or distributed, is based on global state information.
2. **Quasi-global**: distributed broadcast protocol is based on partial global state information.
3. **Quasi-local**: distributed broadcast protocol is based on mainly local state information and occasional partial global state information.
4. **Local**: distributed broadcast protocol is based on solely local state information.

One common misconception is about the difference between the *centralized algorithm* where the source node determines the complete broadcast process and the *distributed algorithm* where the broadcast process is determined distributed at each node. We will show later that any model can be implemented in a distributed way. Also, any model with local information can be implemented in a centralized way once sufficient amount of information is collected at the source through information exchanges. To provide a uniform model, we assume that initially each node knows only its neighbor set, global or partial global state information is collected later through information exchanges (in terms of rounds) among neighboring nodes.

There are in general two models of neighbor set information: *neighbor set without host positions* and *neighbor set with host positions* (obtained through GPS). 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 [2]. In the subsequent discussion, it is assumed that neighbor set information does not include neighbor positions.

2.2 Related Work

We will review some of the representative broadcast protocols for each of the four groups of protocols based on the proposed taxonomy.

The most widely used global broadcast protocol is based on Guha and Khuller's approximation algorithm [11] and has been used in protocol design by Das et al [8, 27]: All nodes are initially colored white. The node with the maximum node degree is selected and colored black, all its neighbors are colored gray. A recursive selection process runs until no white node exists: 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 if u and its neighbor v can cause the maximum white neighbor nodes change color to gray. The modified algorithm guarantees an approximation ratio $O(\ln \Delta)$ under any random graph, where Δ is the maximum node degree of the network. Simulation results show that the result of the former algorithm is close to the MCDS under the unit disk graph. Therefore, this algorithm can be used as a lower bound of the MCDS and it is simply called MCDS.

Like any centralized algorithm, the MCDS can be implemented in a distributed way: A root is selected first through an election process and the node with the maximum node degree is the winner. The neighbors of the root are gray and the one with the maximum number of white neighbors is selected by the root. The newly selected gray node changes its color black and reports to the root about its neighbor status (i.e., status of nodes that have just changed from white to gray). This process repeats until none of gray nodes have white neighbors. Another simple way is for each node to broadcast its link status, and then, run the MCDS at each node to determine its state. This distributed MCDS requires $O(diam)$ sequential rounds, where $diam$ is the diameter of the network. In addition, it does not support locality of maintenance – any state change has to be broadcasted in the whole network.

Unlike global broadcast protocol, quasi-global broadcast protocol does not need to collect the whole global state information. Only partial global state information is collected, typically with the help of a global infrastructure such as a ST. The protocol proposed in [1, 29] fits into this category: A spanning tree is first constructed starting from the selected root (through an election process), a *maximum 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. A maximum independent set is a set in which any other node in the network is a neighbor of a node in the set. Nodes in the maximum independent set 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 connected dominating set. This protocol, called *spanning-tree-based CDS broadcasting* (STCDS), has a constant approximation ratio of 8. 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).

The cluster approach falls into the quasi-local model. Clusters are formed by first electing a clusterhead and, then, its neighbors joining in the cluster as non-clusterhead members. There are many clustering approaches [3, 4, 9, 10, 15]. A simple one, called the *lowest-id cluster algorithm* (LID), initializes all nodes to be 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. The process continues until there is no white node. 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 it happens when the network is a linear chain with decreasing ids from one end to the other end (this is why this approach is called quasi-local), resulting $O(diam)$ rounds of information exchanges. However, this situation rarely happens. In the average case, the cluster formation can be considered as a localized process (as will be shown in the section on simulation).

Other clustering formations are possible: Clusters can be formed to be overlapped or disjointed; clusterheads can be elected by choosing the node with the lowest node id (LID) or the one with the highest node degree (HD) [3, 10]. Clusterheads form a DS, but not a CDS. In fact, clusterheads form an MIS. For a CDS, each clusterhead selects some cluster members, called forward nodes, to connect adjacent clusters. Sinha, Sivakumar and Bharghavan [26] propose a variation of the cluster approach, called *core broadcast* (CB) which also includes the selection process of forward nodes: Initially all nodes are white. A white node determines its dominator by selecting its black neighbor which 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 (probably selected by itself). The coloring process continues until no white node exists. 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 which is not covered by the forward node set or have a gray neighbor whose dominator is not covered by the forward node set. A node is said to be covered by a forward node set if it is a member of the forward node set or a neighbor of any member of 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.

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 [30] *making 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 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. The marking process together with Rules 1 and 2 is simply labelled as MPR1&2. Recently, Dai and Wu [7] extend Rules 1 and 2 to Rule k to further reduce nodes in the CDS without increasing the computational complexity of MPR1&2. 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 [28] where only black nodes (except the source) forward the broadcast packet.

3 Forward-Node-Set-Based Broadcast Protocol (FNSB)

In this section, we first review a preliminary work based on the local model that motivates the proposed *forward-node-set-based broadcast protocol* (FNSB). The FNSB is quasi-local based on the clustered network and it is presented in three subsections: cluster formation and neighbor clusterhead information gathering, forward node set selection and reduction, and broadcast process and its termination. The FNSB is then illustrated with an example, together with other broadcast protocols.

3.1 Preliminary work

In [14], Lim and Kim provide two local algorithms based on the pruning technique to select the forward node set: *self pruning* (SP) and *dominant pruning* (DP). The SP algorithm only exploits the knowledge of directly connected neighbor information. A node does not re-broadcast a packet if all its neighbors have been covered by the previous transmission. The idea of self-pruning (SP) is also discussed in [21, 28]. The DP algorithm uses 2-hop neighbor information to compute each node's forward node set. Specifically, if v is the sender and u is the receiver, u selects its forward node set from $N(u) - N(v)$ to cover 2-hop neighbors in $N^2(u) - N(u) - N(v)$, where $N^k(u)$ represents the k-hop neighbor set of u. A similar forward node selection algorithm is also proposed in [23]. The forward node set is selected in such a way that they cover all the nodes within 2 hops.

Lou and Wu [16] extend Lim and Kim's dominant pruning technique. Two algorithms, *total dominant pruning* (TDP) and *partial dominant pruning* (PDP), are proposed. These approaches utilize the forwarding node's 2-hop neighbor set information to further reduce broadcast redundancies. The TDP requires the sender piggyback its 2-hop neighbor set information along the broadcast packet. With this information, the receiver can prune all the nodes in the sender's 2-hop neighbor set from the receiver's 2-hop neighbor set that needs to be covered. Apparently, the TDP will generate a smaller forward node set than the DP, but it also introduces some overhead when the broadcast packet piggybacks the 2-hop neighborhood information. The PDP, without using the piggybacking technique, directly extracts the neighbors of the common neighbors of both sender and receiver from the receiver's 2-hop neighbor set (see Figure 2 where w can be removed from the 2-hop coverage set of u without sacrificing clusterhead connectivity). Simulation results in [16]

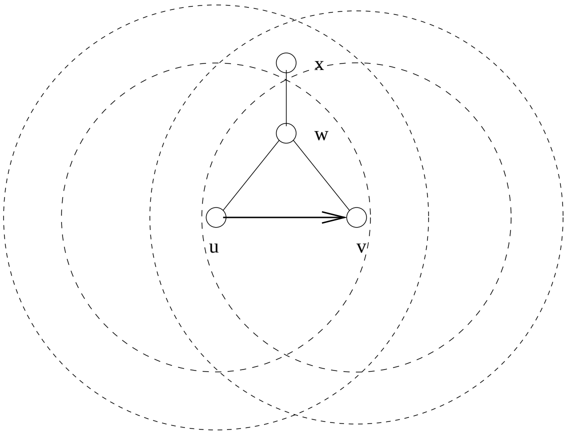


Figure 2: Eliminating neighbors of common neighbors in PDP.

show that the PDP algorithm avoids the extra cost as in the TDP introduced by piggybacking 2-hop neighborhood information with the broadcast packet, but achieves almost the same performance improvement. Note that the pruning approach based on neighbor position rather than 2-hop neighbor set can also be used [28]. In Figure 2, once v determines that the distance between its neighbor w and the incoming node u is less than the transmission radius, there is no need to cover w. However, neighbor position alone is not sufficient to detect neighbors of common neighbors (w in Figure 2). Therefore, neighbor position information only is weaker than 2-hop neighbor set information.

Both the TDP and PDP do not work well in some extreme cases when the network is dense. Consider the example in Figure 3 (a) where all 1-hop neighbors of v are on the circle of C (with radius r centered at v) and all 2-hop neighbors of v are on the circle of C' (with radius $2r$ center at v). Clearly, each 2-hop neighbor of v, u, on the circle of C' can only be covered by exactly one 1-hop neighbor of v, w, on the circle of C whose position is exactly on the line connecting v and u. When the number of nodes on C' increases infinitively, the number of selected forward nodes on C is also infinitive (See Figure 3 (a)). Therefore, the approximation ratio is $O(n)$, where n is the number of hosts in the network. In fact, 15 nodes are sufficient to cover all 1-hop and 2-hop neighbors of v (see Figure 3 (b)). The above case motivates our work in this paper: We apply the idea of TDP and PDP to the clustered network consists of clusterheads only. The clustered network basically converts a given dense graph to a spare graph. We will show that this approach is quasi-local with localized maintenance. In addition, it has a constant approximation ratio and a small CDS in the average case.

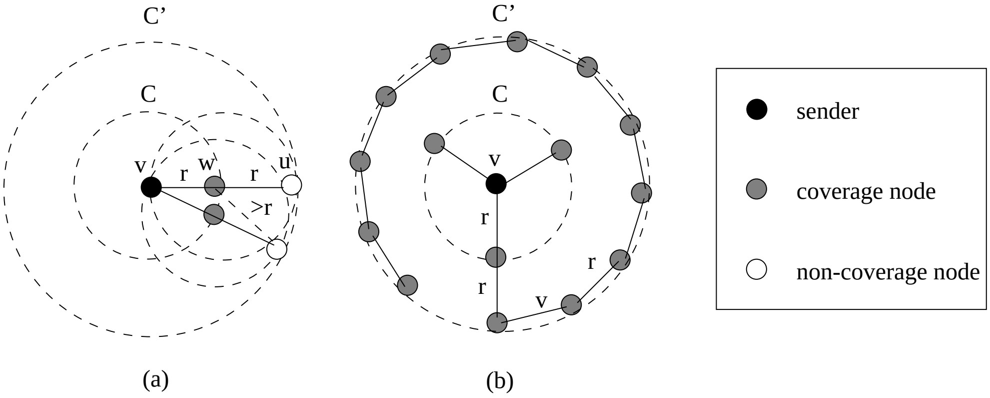


Figure 3: A example of the worst case: (a) node v needs an infinitive number of 1-hop forward nodes to cover its 2-hop neighbors and (b) a solution with 12 coverage nodes.

3.2 Cluster formation and neighbor clusterhead information gathering

In the forward-node-set-based-broadcast protocol (FNSB), the network is partitioned into disjoint clusters by the lowest-id cluster algorithm (LID). Figure 4 (a) shows the result of applying the algorithm in a MANET with 10 nodes. The pure clustered network does not support locality of maintenance; however, it can be made localized if a slightly different cluster construction strategy is applied: Once a cluster is formed, a non-clusterhead node never challenges the current clusterhead. If a clusterhead moves into an existing cluster, the coming clusterhead will give up its role of clusterhead. If a clusterhead moves out of a cluster, the remaining non-clusterhead nodes in this cluster determines their new clusters. A node that has clusterhead neighbors will take the adjacent clusterhead that has the smallest id as its new clusterhead and joins in that cluster. For those nodes that have no clusterhead neighbors, the cluster formation process is applied among those nodes to form new clusters. Thus, the clusters can be mobility adaptive and the changes in a cluster can be limited in a restricted area. A similar approach called least-clusterhead-change clustering algorithm (LCC) has been adopted in [3]. The LID has a constant approximation ratio if the forward node set is carefully selected locally at each clusterhead (as will be seen later in Theorem 3).

Note that the above strategy cannot be applied in the STCDS [1], where a special maximum independent set has to be maintained in order to ensure a approximation ratio of 8: In the STCDS, for any black node, there exists a 2-hop black node. Consider a linear network where nodes are marked black and gray in an alternate fashion with the first node in the chain being black. When the second node moves out followed by two new nodes move in to form a new chain, the hop count

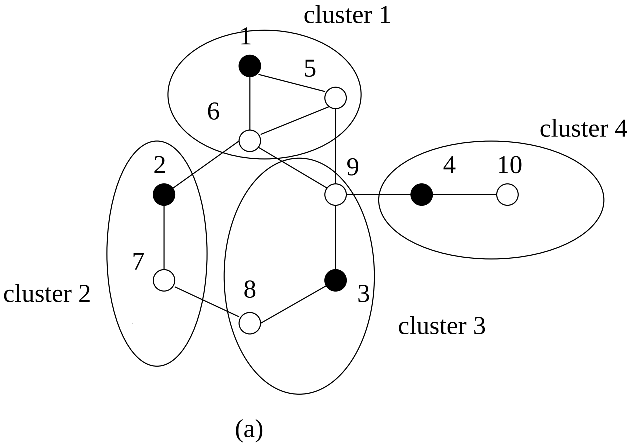


Figure 4: A sample network: (a) the network is partitioned into clusters by the LID and (b) a spanning tree (rooted at 6) by the STCDS.

between first one and third nodes (both are black) becomes three. A global re-marking process is unavoidable. Therefore, the STCDS does not exhibit the property of the locality of maintenance.

There are two ways to define a clusterhead v's *neighbor clusterhead set* $C(v)$:

1. **3-hop coverage**: Each clusterhead covers all clusterheads within 3 hops. That is, $C(v)$ includes all clusterheads in $N^3(v)$.
2. **2.5-hop coverage**: Each clusterhead covers all clusterheads in 2 hops and some clusterheads that are 3 hops away. That is, $C(v)$ includes all clusterheads in $N^2(v)$ and the clusterheads that have non-clusterhead members in $N^2(v)$.

A node is called a *1-hop (2-hop) gateway* if it is used to connect a neighbor clusterhead that is separated by 2 hops (3 hops). Sometimes a node can be a 1-hop gateway and a 2-hop gateway at the same time. From the information periodically sent by neighboring 1-hop and 2-hop gateways, a clusterhead v can compute its neighbor clusterhead set $C(v)$. The clusterheads in $C(v)$ are grouped into two classes: (a) the clusterheads that are 2 hops away from v, represented by $C^2(v)$ and (b) the clusterheads that are 3 hops away from v, represented by $C^3(v)$. Therefore, $C(v) = C^2(v) \cup C^3(v)$. Notice that $C^2(v)$ is the same under both methods, but not for $C^3(v)$. In Figure 4 (a), based on the 3-hop coverage method, $C(1) = \{2, 3, 4\}$, $C^2(1) = \{2\}$, and $C^3(1) = \{3, 4\}$; while based on the 2.5-hop coverage method, $C(1) = \{2, 3\}$, $C^2(1) = \{2\}$, and $C^3(1) = \{3\}$. For clusterhead 3, $C(3) = \{1, 2, 4\}$ under both methods where $C^2(3) = \{4\}$ and $C^3(3) = \{1, 2\}$.

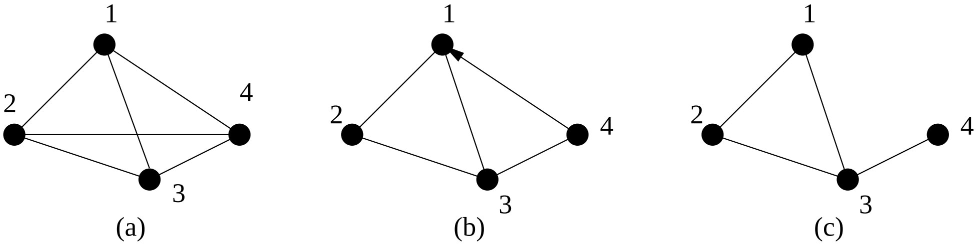


Figure 5: The cluster graph and adjacent cluster graph of the sample network in Figure 4(a): (a) the cluster graph based on the 3-hop coverage method, (b) the cluster graph based on the 2.5-hop coverage method, and (c) the adjacent cluster graph.

To gather neighbor cluster information, each node exchanges information with its neighbors. Each node needs only two sends after the formation of the cluster. Under the 2.5-hop coverage method, since each 2-hop neighbor belongs to only one cluster and it knows its clusterhead after the cluster formation process, these two sends are equivalent to the 2-hop neighborhood information gathering process in terms of the size of each packet (one with $O(\Delta)$ and the other with $O(\Delta^2)$). Under the 3-hop coverage method, each 2-hop neighbor may have multiple adjacent clusterheads that belong to different clusters (as 2-hop neighbor u of v in Figure 1). In addition, maintenance of 3-hop neighbor information costs more than 2.5-hop information.

A *cluster graph* G' is constructed from clusterheads in the given clustered network G: Each vertex of G' comes from a clusterhead in G, and each directed link (v, w) of G' is from clusterhead v to a clusterhead w ($w \in C(v)$). For the 3-hop coverage method, all the clusterheads within $N^3(v)$ will be included in $C(v)$. Therefore, the corresponding cluster graph G' will be an undirected graph. That is, for any two clusterheads v and w, if $w \in C(v)$, then $v \in C(w)$. Both directed links (v, w) and (w, v) exist in graph G'. But with the 2.5-hop coverage method, the clusterhead v builds its clusterheads set $C(v)$ that consists of clusterheads in $N^2(v)$ and clusterheads of those non-clusterhead members that are in $N^2(v)$. In this case, there may exist two clusterheads v and w, where $w \in C(v)$, but $v \notin C(w)$. For the sample network in Figure 4 (a), the cluster graphs under both methods are shown in Figures 5 (a) and (b). For both cases, the following theorem is true for the cluster graph $G^{'}$.

Theorem 1: *The cluster graph $G^{'}$, built by using either the 3-hop or 2.5-hop coverage method from a connected graph G, is a strongly connected graph.*

Proof : A graph is a strongly connected graph if and only if there exists a path for any two

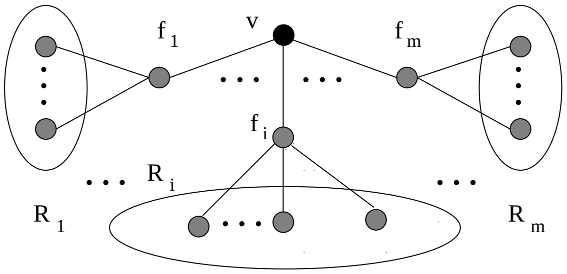


Figure 6: A 3-level tree rooted at clusterhead v: $\{< f_i, R_i >\}$.

vertices in the graph. Construct a *adjacent cluster graph* $G^{''}$ from the given clustered network G: Each cluster in G is considered as a vertex in $G^{''}$, if two clusters $c(u)$ and $c(v)$ in G are neighbors (i.e., a link exists in G that connects a node in cluster $c(u)$ and a node in cluster $c(v)$), a link exists between the corresponding vertices in $G^{''}$ (See Figure 5 (c) as an example). Because G is a connected graph, for any two vertices $c(u)$ and $c(v)$ in $G^{''}$, there exists a path $P = (u, u_1, u_2, ..., v)$ in G. The path traverses the clusters $(c(u), c(u_1), c(u_2), ..., c(v))$ in $G^{''}$. It means that a path exists in $G^{''}$ that connects $c(u)$ and $c(v)$. Therefore, $G^{''}$ is a strongly connected graph.

It can be easily seen that adjacent cluster graph $G^{''}$ is a subgraph of cluster graph $G^{'}$ based on either the 3-hop or the 2.5-hop coverage method. That is, the cluster graph $G^{'}$ contains the adjacent cluster graph $G^{''}$. Therefore, adding some links (directed or undirected) to the adjacent cluster graph $G^{''}$ can build the cluster graph $G^{'}$ without changing its strongly connected property. Since $G^{''}$ is strongly connected, $G^{'}$ is also strongly connected. $\square$

This theorem extends a result by Chlamtac and Farago [5] where all 3-hop clusterheads need to be covered to ensure the connectivity of clusterheads.

3.3 Forward node set selection and reduction

Clusterhead v's forward node set is a subset of gateways by which v can connect to the clusterheads in $C(v)$. Host v connects to a clusterhead in $C^2(v)$ via a 1-hop gateway and it connects to a clusterhead in $C^3(v)$ with two 2-hop gateways. The forward node set of v is computed on-demand to cover all the clusterheads in $C(v)$. Notice that since the 3-hop coverage and the 2.5-hop coverage generate different $C(v)$s, the corresponding forward node sets are also different.

We use a greedy algorithm to determine the forward node set at each node. Note that the forward node set of each node is centrally computed at the node and is piggybacked at the broadcast packet forwarded by the node. When a clusterhead v receives a packet from another clusterhead u via u's forward node set $F(u)$, v selects a minimum number of gateways to form its forward node set $F(v)$, through which v can cover all the clusterheads in $C(v)$. The forward node set is organized as branches rooted at v: $\{< f_i, R_i >\}$ (see Figure 6), where f_i is a 1-hop gateway used to cover some clusterheads in $C^2(v)$, and, R_i is a set of 2-hop gateways that are neighbors of f_i and are used to cover some clusterheads in $C^3(v)$. R_i may be empty if none in $C^3(v)$ is covered. Based on Figure 6, v and its forward nodes form a three-level forward node tree.

At the beginning, all the clusterheads in $C(v)$ are *uncovered*. When a forward node branch $< f, R >$ is selected, some clusterheads in $C^2(v)$ and/or $C^3(v)$ are covered. f is selected if it can cover the maximum number of uncovered clusterheads in $C(v)$ directly or indirectly via its neighbors. Notice that some clusterheads in $C^2(v)$ are directly covered by f and some clusterheads in $C^3(v)$ are indirectly covered by f via nodes in $N^2(v)$, they all count to the number of clusterheads in $C(v)$ covered by f. After f is determined, R can also be determined in a similar way: The neighbor of f which covers the maximum number of the uncovered clusterheads in $C^3(v)$ will be first selected into R until all the clusterheads in $C^3(v)$ that are indirectly covered by f are selected. When $< f, R >$ is included in $F(v)$, all the clusterheads that are covered by $< f, R >$ in $C(v)$ change their states to *covered*. The selection process repeats until all the clusterheads in $C(v)$ are covered. The detail selection process can be found in the appendix.

When a clusterhead u uses its forward node set to deliver a packet to all the clusterheads in $C(u)$, it piggybacks u and $C(u)$ in the broadcast packet so that the receiver can prune more clusterheads from its coverage area. Consider two neighbor clusterheads u and v. Suppose a broadcast packet sent by u, piggybacked with $C(u)$ and u, reaches v with the help of forward nodes relaying. Since all the clusterheads in $C(u) \cup \{u\}$ are covered by u's forward node set, they do not need to be covered again when v computes its forward node set. For the 3-hop coverage method, v will determine its forward node set $F(v)$ to cover any nodes in $U(v) = C(v) - C(u) - \{u\}$, where $U(v)$ is a subset of $C(v)$ that needs to be covered. But for the 2.5-hop coverage method, u's forward node set may cover some extra clusterheads in addition to $C(u) \cup \{u\}$. More specifically, if v is 2 hop away from u and u uses a path (u, f, r, v) to deliver the broadcast packet to v, clusterheads in $N(r)$ that are not in $C(u)$ also receive the broadcast packet. These clusterheads can also be excluded from $U(v)$. Therefore, $U(v) = C(w) - C(v) - \{v\} - N(r)$. In Figure 4 (a) with root 1, node 1 uses path (1, 6, 9, 3) to deliver the broadcast packet to node 3. Clusterhead 4 is covered by node 9 but it is not in $C(1)$ based on the 2.5-hop coverage method. Therefore, node 9 can be removed from $C(3)$ when node 3 calculates its forward node set.

3.4 Broadcast process and its termination

When a source node initiates a broadcast process, it follows the steps below:

1. If the source is not a clusterhead, it just sends the broadcast packet to its clusterhead.
2. When a clusterhead receives the broadcast packet for the first time, it chooses its forward node set to forward the packet to all its neighbor clusterheads. These neighbor clusterheads should exclude the forwarding clusterhead itself and those neighbor clusterheads of the forwarding clusterhead. The neighbor clusterheads of this forwarding clusterhead are piggybacked with the broadcast packet as well as the forward node set for the forwarding purpose. If the received packet is a duplicated one, the clusterhead does nothing.
3. When a non-clusterhead node receives the broadcast packet for the first time, and if it is in the forward node set, it relays the packet or else it does nothing.

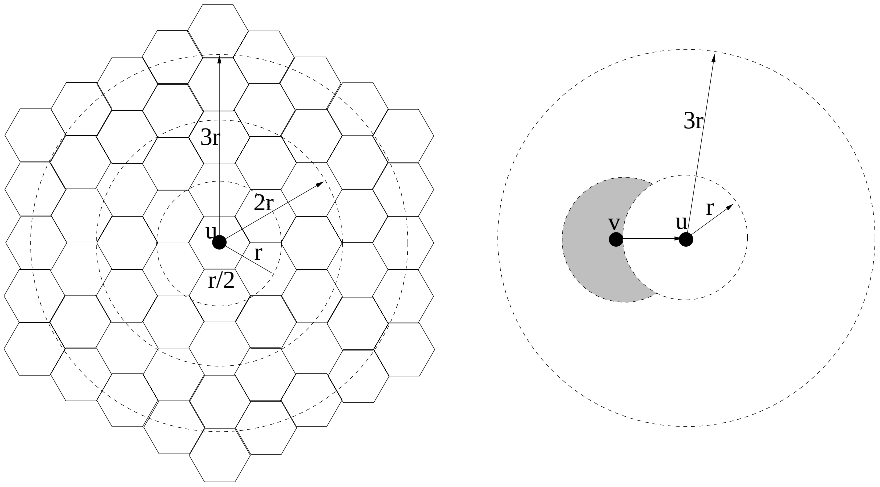
Using such a protocol, all the clusterheads in the network will eventually receive the broadcast packet provided that the network is connected. After all the clusterheads re-broadcast the packet in their clusters, all the nodes in the entire network will receive the packet.

Theorem 2 : *The proposed protocol successfully delivers a packet to all of the nodes in a given connected network and the broadcast process terminates in finite time.*

Proof: The set of clusterheads forms a DS of the network. That is, a node in the network is either a clusterhead or a neighbor of a clusterhead. Based on the protocol, each clusterhead selects a set of forward nodes to connect all its neighbor clusterheads (in a 3-hop coverage area or a 2.5-hop coverage area). In either case, all its neighbor clusters are covered (connected). Therefore, the broadcast packet reaches all the clusterheads in a finite number of steps. Since clusterheads form a DS, one extra step of forwarding covers all the nodes in the networks. □

Theorem 3 : *The number of the forward nodes for a broadcast in the proposed protocol without pruning has a constant approximation ratio to the MCDS.*

Proof: The clusterheads form the maximum independent set (MIS) of the network. Suppose the MCDS of the network is opt. It is known [1] that the size of the MIS of the network is bounded by $4 \cdot |opt| + 1$. In a clustered network, two clusterheads are 2 or 3 hops away from each other. Each clusterhead u selects some non-clusterhead nodes to connect all the clusterheads in its 3



(a) (b)

Figure 7: The 3-hop coverage area of node u: (a) covered by hexagons and (b) applied with the pruning technique.

hops vicinity area S_{3-hop}. Therefore, the area of S_{3-hop} is at most $\pi(3r)^2$. Figure 7 (a) shows three nested cycles centered at node u with radius r, $2r$ and $3r$, respectively. Notice that all the clusterheads are members of the IS, which means that any two clusterheads are not in each other's transmission range. Therefore, the distance of two clusterheads is at least r. We use the hexagon as an area unit S_{min} to cover the area S_{3-hop}. Clearly, if the side of the hexagon is $\frac{r}{2}$, there is at most one clusterhead in each hexagon S_{min}. Based on the solution shown in Figure 7 (a), the total number of hexagons that are needed is at most 61. Therefore, excluding u, the number of clusterheads in the S_{3-hop} are upper bounded by $N = 61 - 1 = 60$. Since each clusterhead needs up to 2 forward nodes, the maximum number of forward nodes for each clusterhead is $2N$. Therefore, the total number of the forward nodes (including clusterheads) for a broadcast is $(2N + 1) \cdot |MIS| = (8N + 4) \cdot |opt| + (2N + 1)$, a constant approximation ratio to the MCDS. $\square$

Note that in the above theorem, it is assumed that no pruning technique is applied. As shown in Figure 7 (b) when v is the sender and u is the receiver, the size of the coverage area of v is at least a cycle with a radius r. Clearly, the shadowed area (close to $\frac{2}{3}\pi r^2$) can be removed from the coverage area of node u to reduce the approximation ratio. The above result just shows an upper bound and the worst case rarely occurs in a real system. Therefore, our focus will be in the average case which is much lower (with an approximation ratio between 2 to 3) as shown later in simulation.

3.5 Illustration

We use the sample network in Figure 4(a) to show how the proposed protocol works and the difference between our protocol and others. Suppose that node 1 is the broadcast source, running the selection process with the 3-hop coverage method, $U(1) = C(1) = C^2(1) \cup C^3(1) = \{2\} \cup \{3, 4\} = \{2, 3, 4\}$, Node 5 (neighbor of 1) covers 3 and 4 (indirectly). Node 6 (neighbor of 1) covers 2 (directly), 3 (indirectly), and 4 (indirectly). So 6 is selected. 9 (neighbor of 6) is selected that covers both 3 and 4 directly. Therefore, the forward node set of node 1 is $F(1) = \{< 6, \{9\} >\}$. $C(1) = \{2, 3, 4\}$ and node 1 are piggybacked along the packet. Once nodes 2, 3 and 4 receive the packet, they will compute their corresponding forward node sets. $U(2) = C(2) - C(1) - \{1\} = \phi$, the forward node set of node 2 is empty. Similarly, $U(3) = U(4) = \phi$ and their forward node sets are both empty. Therefore, nodes 2, 3 and 4 just broadcast the packet among their cluster members. In this case, only nodes $1, 2, 3, 4, 6$ and 9 forward the packet.

With the 2.5-hop coverage method, $U(1) = C(1) = C^2(1) \cup C^3(1) = \{2\} \cup \{3\} = \{2, 3\}$, 5 covers 3 (indirectly) and 6 covers 2 (directly) and 3 (indirectly), so 6 is selected. 9 (neighbor of 6) is selected that covers 3 directly. The forward node set is $F(1) = \{< 6, \{9\} >\}$; $C(1) = \{2, 3\}$ and node 1 are piggybacked with the packet. For nodes 2 and 3, since $U(2) = C(2) - C(1) - \{1\} = \phi$, node 2's forward node set is empty. $U(3) = C(3) - C(1) - \{1\} - N(9) = \{1, 2, 4\} - \{2, 3\} - \{1\} - \{3, 4\} = \phi$ and $U(4) = \phi$. Therefore, $F(3)$ and $F(4)$ are both empty. In this case, only nodes $1, 2, 3, 4, 6$ and 9 forward the packet.

When using the core broadcast (CB) protocol [26] for broadcasting, the cores are first determined by the core generation algorithm provided in [25]. Since cores are determined in a distributed way, there are many possible results. One possible set of cores is $\{3, 4, 6, 7, 9\}$. The CB requires each core compute its forward node set independently, nodes 2 and 8 will be included in the forward node sets of nodes 3 and 7, respectively. When node 1 is the source, nodes $1, 2, 3, 4, 6, 7, 8$ and 9 need to forward the packet.

For the partial dominant pruning algorithm (PDP) in [16], when node 1 is the source, its forward node set is $F(1) = \{6\}$, then, $F(6) = \{2, 9\}$, $F(2) = \{7\}$, $F(9) = \{3, 4\}$, $F(7) = \{8\}$, $F(3) = \{8\}$, $F(4) = \phi$. Therefore, nodes $1, 2, 3, 4, 6, 7, 8$ and 9 forward the packet.

Based on the marking process proposed in [30], the CDS is $\{2, 3, 4, 5, 6, 7, 8, 9\}$. By using Rule 1, node 5 is extracted from the CDS. The final CDS with Rules 1 and 2 is $\{2, 3, 4, 6, 7, 8, 9\}$. Only source node and nodes in the CDS will forward a broadcast packet. Therefore, nodes $1, 2, 3, 4, 6, 7, 8$ and 9 forward the packet.

For the spanning-tree-based CDS algorithm (STCDS) proposed in [1], if the root is node 6, the nodes marked black are $3, 4, 6$ and 7 and the nodes marked gray are 2 and 9 (see Figure 4(b)). The connected dominating set for forwarding a broadcast packet (including the source 1) is $1, 2, 3, 4, 6, 7$ and 9.

For the approximation algorithm of the MCDS in [8], the MCDS of the network is $\{2, 3, 4, 6, 9\}$. Therefore, when node 1 is the source, nodes $1, 2, 3, 4, 6$ and 9 forward the packet.

4 Performance Simulation

In this section, we measure the average number of forward nodes for a packet to reach all the nodes and the average number of sequential rounds needed to establish a CDS in a randomly generated network. The simulation runs under the following simulation environment: The area of working space is 100 × 100. Nodes are randomly placed in this confined area. The links are bi-directional links. If the network is not connected, it is discarded. There are no other traffic except the one generated from the broadcast packets and the one used to maintain cluster structures. No transmission errors (such as collision and contention) are considered here. It is assumed that all these issues are taken care of at an ideal MAC layer. The movement range of nodes should be relatively small during the broadcast period compared with the node's transmission range. Otherwise, it is not cost-effective to maintain cluster structures since the clusterheads cannot set up the fresh neighbor set in a timely manner. Specifically, we assume the following environment: Mobile hosts are still allowed to roam freely in the working space. However the broadcast process (including forward node selection and the broadcast process itself) is done quickly so that the coverage area of each host (1-hop and 2-hop neighborhood) remains the same during the process for each host v. In addition, each host has updated and consistent information of its coverage area when the broadcast process starts. In each simulation, two different transmission ranges (25 and 50) are tested. For each fixed transmission range, the number of nodes in the graph ranges from 20 to 100. We generate 1000 random graphs to get the average number of the forward node set. The source node for each network is also randomly selected.

The performance of our proposed protocol (FNSB) is compared with other approaches mentioned in this paper: the partial dominant pruning algorithm (PDP) proposed in [16], the core broadcast protocol (CB) in [26], the dominating set based broadcast protocol [28] by using the marking process with Rules 1 and 2 (MPR1&2) in [30], and the spanning-tree-based CDS algorithm (STCDS) proposed in [1]. The clusters are constructed with the lowest-id (LID) and highest

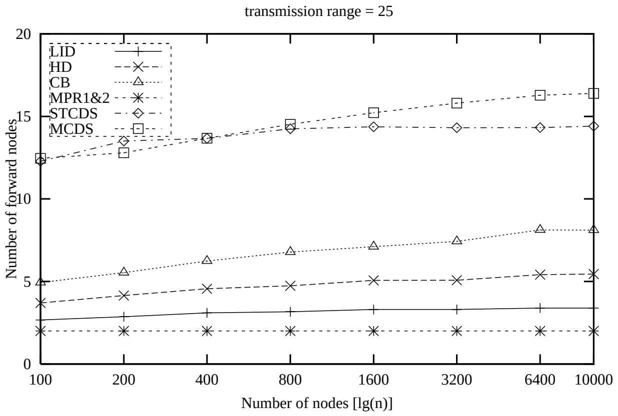


Figure 8: The average number of rounds for different protocols.

node degree (HD) cluster algorithms. Both 3-hop coverage (referred as I) and 2.5-hop coverage (referred as II) methods are applied for a clusterhead to gather its neighbor clusterhead information. We use the result of the MCDS in [8] as the approximation of the lower bound for the MCDS.

Figure 8 shows the average number of rounds for the different protocols to build the underlying infrastructure (including the CDS) of the network when the node transmission range is 25. The MPR1&2 has the minimum constant number of rounds among all. Since the MPR1&2 algorithm requires each node gather 2-hop neighbor set information to compute the CDS of the network, it actually needs just 2 rounds of neighbor set information exchanges (if coverage nodes are neighbors of covered nodes in Rules 1 and 2; otherwise, 3 rounds are needed). Both clustered algorithms (HD and LID) have small average number of rounds although the clusterhead election process will happen sequentially in some extreme cases. In the HD, node id is used to break a tie in node degree (if any). The number of rounds for the HD is a slightly larger than that for the LID. The reason is that, for the HD algorithm, a clusterhead is elected based on its node degree. That a node has higher node degree means that it has more neighbors, and its neighbors are also more likely to have higher node degrees than nodes in other parts. Therefore, the sequential election process is more likely to happen when using the HD than using the LID. The average number of rounds for the CB to compute its core nodes is also relatively small since it is indeed a quasi-local algorithm based on our taxonomy. The average number of the rounds for the MCDS algorithm is the same as the number of nodes in the MCDS because for each round only one node is selected to add into the MCDS. The STCDS needs many rounds to set up a tree-style CDS. The algorithm in [1] has three stages to form the CDS: determine each node's position in the topological sorting

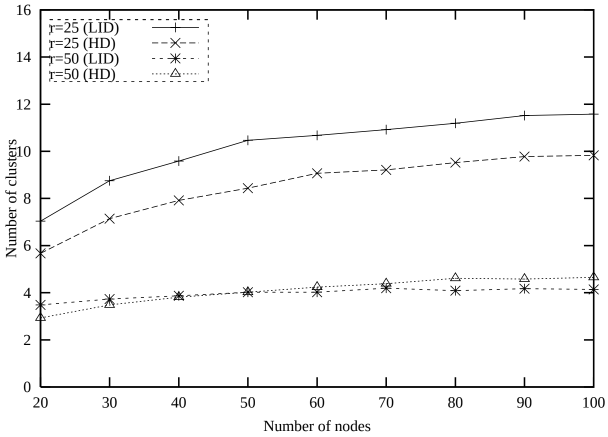


Figure 9: Average number of clusters in a network.

order based on a spanning tree, form an MIS of the network and connect these nodes in the MIS to build a tree-style CDS. In [1], the algorithm requires these three stages execute one at a time. Although these three stages can be executed simultaneously for the whole network, they still need many rounds. In Figure 8, the curve of the STCDS shows the average number of rounds when all these three stages execute simultaneously. It is much higher than others and is comparable to the MCDS, which is a global algorithm.

Figure 9 shows the average number of the clusters in a network with different transmission ranges (25 and 50). The number of clusters does not change significantly when the size of network increases. It suggests that the cluster structure is more suitable for a dense network. The number of clusters built by the LID algorithm is a slightly larger than that built by the HD algorithm.

Figure 10 (a) shows the result when the node's transmission range is 25. This range is rather small compared with the size of the working space. When the number of nodes is small (between 20 and 30), the FNSB has 10-20% more forward nodes than any other algorithms (PDP, CB, MPR1&2, and STCDS). While the number of nodes increases, the curves of the PDP and the CB rise significantly, but the slopes of the FNSB, the MPR1&2 and the STCDS stay relatively flat. When the number of nodes is 100, the number of forward nodes of the FNSB is only 60% of that of the PDP and the CB. This is because the number of clusters in the confined area is rather insensitive to the number of nodes in this area. Therefore, the size of forward nodes does not increase much when the size of network increases. The CB protocol does not work well when the network is large because it requires each node to determine its forward node set independently.

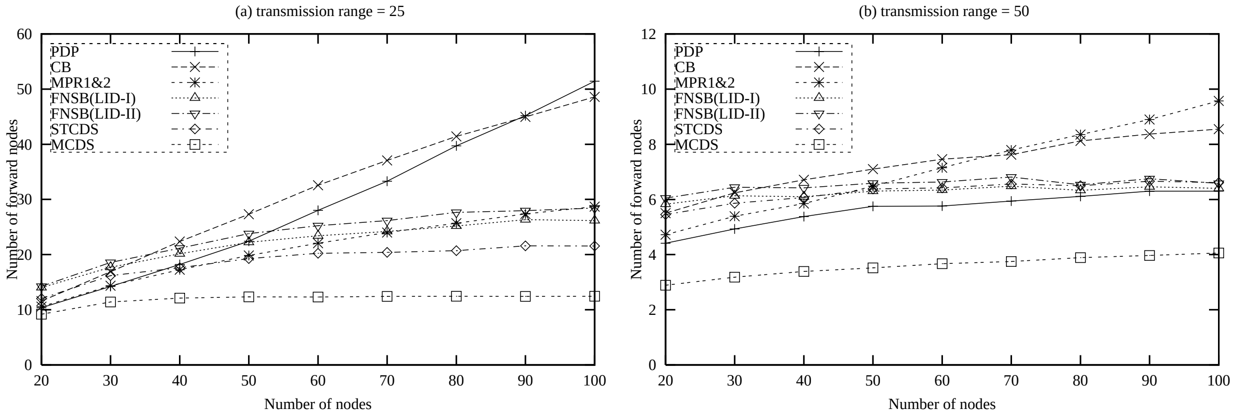


Figure 10: Average number of the forward nodes: (a) range = 25 and (b) range = 50.

More redundant nodes will be included in the forward node set. The PDP algorithm also has a large number of forward node set when the network is large, even though it piggybacks the forward node set information in the broadcast packet. The FNSB (LID-I) is slightly better than the FNSB (LID-II) because more clusters can be included in $C(v)$ by using the 3-hop coverage method so that more nodes can be pruned. The STCDS has the best performance for the relatively sparse network. The size of forward nodes in STCDS is about 75% of the FNSB.

In Figure 10 (b), the node's transmission range increases to 50. The numbers of clusterheads under all methods (except the MCDS) are all relatively small (between 4 to 10). This is because the network is close to a completely connected graph for a dense network. When the network is small, they are better than the FNSB, but when the number of nodes is large, their performances are worse than the FNSB. Since no matter what size the network is, the diameter of the network is relatively small (less than 4 in average), the number of the clusters is also quite small (i.e., the size of each cluster is large). Therefore, the total number of forward nodes is relatively stable. The figure for the MPR1&2 has a sloping curve because more nodes are selected in the CDS while few can be extracted by Rules 1 and 2 when the number of nodes increases. The number of the forward nodes for the STCDS has 10% size saving when the size of the network is 20 and has almost the same size when the size of the network is 100, compared with the FNSB. This is because when the network diameter is small, the number of forward nodes in the mesh-style CDS (FNSB) is almost the same as that in the tree-style CDS (STCDS). The PDP, in this circumstance, has the best performance for all ranges of numbers.

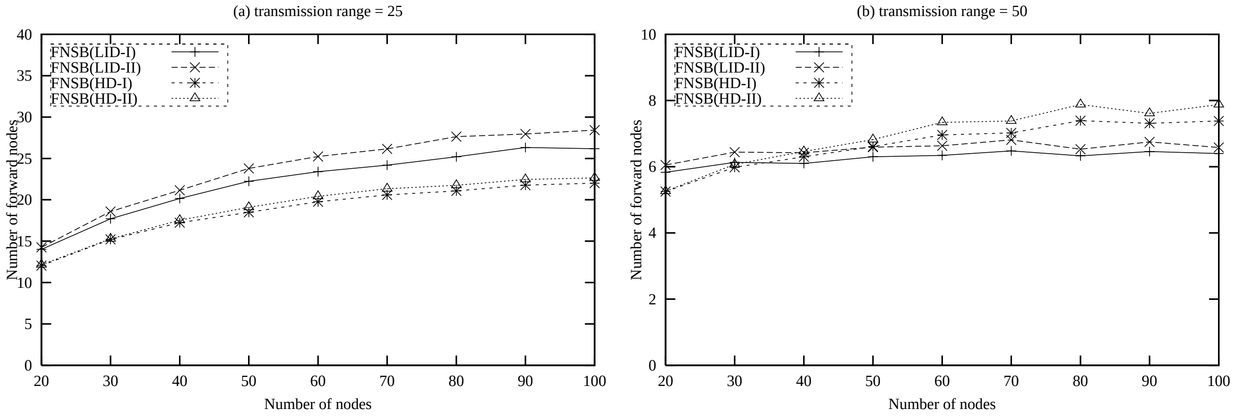


Figure 11: The difference between clusters constructed by the LID algorithm and by the HD algorithm: (a) range = 25 and (b) range = 50.

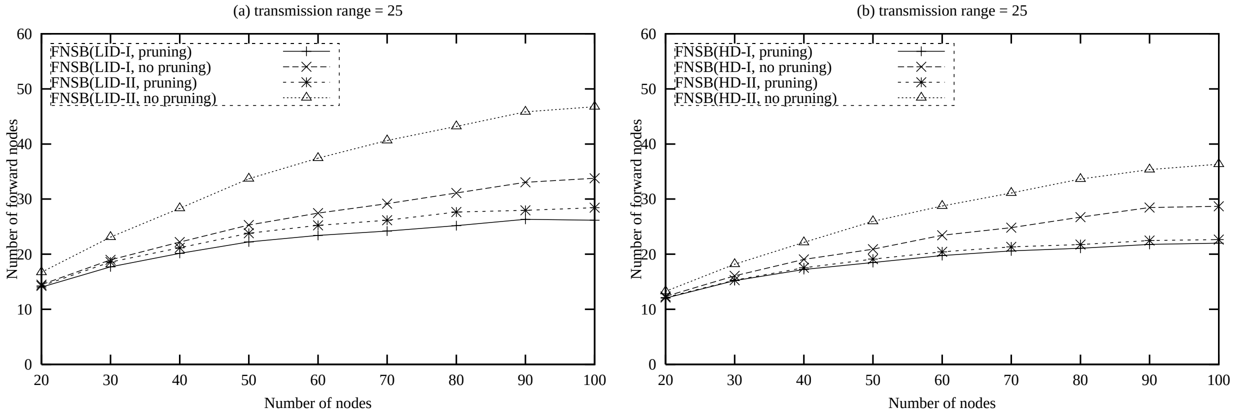


Figure 12: The number of forward nodes with using pruning technique vs. without using pruning technique when the node's transmission range is 25: (a) the clusters are formed by the LID algorithm and (b) the clusters are formed by the HD algorithm.

In Figure 11, the difference between clusters that constructed by the LID and the HD algorithms are compared. When the transmission range is 25, the HD has a better performance than the LID since each cluster can include more nodes if the node with the highest node degree is selected as a clusterhead and, hence, the total number of clusters is smaller on average. When the transmission range becomes 50, the LID shows a better performance than the HD when the number of nodes is over 40. It is because, in this circumstance, the average number of clusters constructed by the LID is less than the one from the HD. So does the total number of forward nodes.

Figure 12 shows the effect of using the pruning technique to extract more forward nodes in the FNSB protocol when the node's transmission range is 25. From the figure, we can see that the number of forward nodes, when using the pruning technique, is less than the one without using it, especially for the one based on the 2.5-hop coverage method. One interest thing is that without pruning, the performance of the FNSB based on the 3-hop coverage method is much better than that based on the 2.5-hop coverage method. But when using the pruning technique, they have almost the same number of forward nodes. Since the 3-hop coverage method is more costly for gathering the neighboring information, the 2.5-hop coverage method is a better choice.

In summary, we have the following observations from the simulations:

- Both the STCDS and the MCDS generate a large number of sequential rounds, while clustered networks (HD and LID) exhibit a small number of sequential rounds in the average case.
- The number of clusters is relatively stable while the size of the network increases.
- The way that each cluster is constructed affects the size of the forward node set. A cluster formed by the HD algorithm has a smaller forward node set than the one formed by the LID algorithm, but the difference is insignificant.
- The FNSB protocol has good performance when the node's transmission range is small and the number of the nodes is large (i.e., the diameter of the network is relatively large).
- The FNSB protocol is relatively insensitive to the size of the network.
- The pruning technique can greatly reduce the total number of forward nodes compared with the one without using it in the clustered network.
- The FNSB has a small approximation ratio to the MCDS, between 2 and 3, in the average case.

- The STCDS has the best approximation ratio when the network is relatively sparse. However, its superiority is not significant in the average case. The STCDS performs marginally when the network is relatively dense.
- The proposed 2.5-hop coverage method is almost as efficient as the 3-hop coverage method when the pruning technique is used.

5 Conclusions

In this paper, we have proposed a taxonomy of broadcast protocols in MANETs. A new quasi-local broadcast protocol, called forward-node-set-based broadcast protocol (FNSB), has been proposed. The FNSB is based on the clustered architecture where each clusterhead computes its own forward node set locally. A non-clusterhead node forwards the broadcast packet if it is a forward node or else it does nothing. Therefore, the broadcast operation can be limited in clusterheads and the nodes in the forward node set. We have also proposed a 2.5-hop coverage method where each clusterhead only needs to cover the clusterheads within 2 hops and the clusterheads which have members within 2 hops. The clusterheads utilize the incoming clusterhead's neighbor clusterhead information that attached to the packet to further reduce the forward node set. This approach makes the broadcast more efficient with a constant approximation ratio under both the worst and average cases. The empirical results of the FNSB show that the total number of the forwarding nodes is relatively stable for different sizes of the network and it outperforms several existing quasi-local and local protocols, such as the core broadcast protocol and the partial dominant pruning technique, when the diameter of the network is relatively large.

References

[1] K. M. Alzoubi, P. J. Wan, and O. Frieder. New distributed algorithm for connected dominating set in wireless ad hoc networks. *Proc. HICSS'35*, 2002. (CD-ROM).

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

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

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

[5] I. Chlamtac and A. Farago. A new approach to the design and analysis of peer-to-peer mobile networks. *Wireless Networks*, 5:149–156, 1999.

[6] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86:165–177, 1990.

[7] F. Dai and J. Wu. Distributed dominant pruning in ad hoc wireless networks. *TR-CSE-02-07*, Feb. 2002.

[8] B. Das, R. Sivakumar, and V. Bharghavan. Routing in ad-hoc networks using a virtual backbone. *Proc. ICCCN'97*, pages 1–20, Sept. 1997.

[9] 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.

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

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

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

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

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

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

[16] W. Lou and J. Wu. On reducing broadcast redundancy in ad hoc wireless networks. *TR-CSE-02-08*, Feb. 2002.

[17] 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.

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

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

[20] 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.

[21] W. Peng and X.C. Lu. On the reduction of broadcast redundancy in mobile ad hoc networks. *Proc. First Annual Workshop on Mobile and Ad Hoc Networking and Computing, MOBIHOC*, pages 129–130, Aug. 2000. Boston, USA.

[22] C. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. *Proc. 2nd IEEE WMCSA*, pages 90–100, Feb. 1999.

[23] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying for flooding broadcast message in mobile wireless networks. *Proc. HICSS-35*, Jan. 2002.

[24] E. M. Royer and C. K. Toh. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, 6(2):46–55, 1999.

[25] P. Sinha, R. Sivakumar, and V. Bharghavan. CEDAR: a core-extraction distributed ad hoc routing algorithm. *IEEE Journal on Selected Areas in Communications, Special Issue on Ad Hoc Networks*, 17(8):1454–1465, Aug. 1999.

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

[27] R. Sivakumar, B. Das, and V. Bharghavan. An improved spine-based infrastructure for routing in ad hoc networks. *IEEE Symposium on Computers and Communications (ISCC) 98*, 1998. Athens, Greece.

[28] 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.

[29] P. J. Wan, K. Alzoubi, and O. Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. *Proc. INFOCOM'2002*, June 2002.

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

Appendix

Assume $F(v) = \{< f_1, R_1 >, < f_2, R_2 >, ..., < f_m, R_m >\}$ is v's forward node set, $U(v)$ is the subset of $C(v)$ that is uncovered so far. At the i^{th} iteration, the forward node f_i is selected from $N^1(v)$. $S(f_i)$ is the subset of the clusterheads in $U(v)$ that is covered by f_i and it consists of two parts: clusterheads in $C^2(v)$ that f_i covers directly and clusterheads in $C^3(v)$ that f_i covers indirectly via nodes in R_i (the latter part is simply denoted as $V(f_i)$). The f_i with the maximum size of $S(f_i)$ will be first selected. Set $R_i = \{r_j | r_j \in N^1(f_i) \cap N^2(v)\}$ where r_j is a 2-hop gateway that can cover the clusterheads in $C^3(v)$. The selection of r_j follows the same rule as the one for f_i.

```
Selection process:

  Let $F(v) = \phi$, $U(v) = C(v)$, $K = N^1(v)$, and $S(f_i) = (N^1(f_i) \cap C^2(v)) \cup (N^2(f_i) \cap C^3(v))$ where $f_i \in N^1(v)$.

  While $(U(v) \neq \phi)$
    (a) Find $f_i$ such that $S(f_i)$ is the maximum in $N^1(v)$. A tie is broken by selecting the node with the smaller id.
    (b) Let $R_i = \phi$, $V(f_i) = N^2(f_i) \cap C^3(v)$.
        While $(V(f_i) \neq \phi)$
          i. Find $r_j$ that covers the maximum number of $V(f_i)$ where $r_j \in N^1(f_i)$. A tie is broken by selecting the node with the smaller id.
          ii. $R_i = R_i \cup \{r_j\}$, $V(f_i) = V(f_i) - N^1(r_j)$.
    (c) $F(v) = F(v) \cup \{< f_i, R_i >\}$, $U(v) = U(v) - S(f_i)$, $K = K - \{f_i\}$, and $S(f_j) = S(f_j) - S(f_i)$ for all $f_j \in K$.
```
