

# On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks

Jie Wu and Hailan Li  
Department of Computer Science and Engineering  
Florida Atlantic University  
Boca Raton, FL 33431  
{jie, hli}@cse.fau.edu

## Abstract

Efficient routing among a set of mobile hosts (also called nodes) is one of the most important functions in ad-hoc wireless networks. Routing based on a connected dominating set is a frequently used approach, where the searching space for a route is reduced to nodes in the set. A set is dominating if all the nodes in the system are either in the set or neighbors of nodes in the set. In this paper, we propose a simple and efficient distributed algorithm for calculating connected dominating set in ad-hoc wireless networks, where connections of nodes are determined by their geographical distances. Our simulation results show that the proposed approach outperforms a classical algorithm. Our approach can be potentially used in designing efficient routing algorithms based on a connected dominating set.

**Keywords:** *ad hoc wireless networks, dominating sets, mobile computing, routing, simulation*

## 1 Introduction

An *ad hoc* wireless network is a special type of wireless network in which a collection of mobile hosts with wireless network interfaces may form a temporary network, without the aid of any established infrastructure or centralized administration. If two hosts that want to communicate are outside their *wireless transmission ranges*, they could communicate only if other hosts between them in the ad hoc wireless network are willing to forward packets for them.

We can use a unweighted graph  $G = (V, E)$  to represent an ad hoc wireless network, where  $V$  represents a set of wireless mobile hosts and  $E$  represents a set of edges. To simplify our discussion, we assume all mobile hosts have same transmission ranges, in other words an edge between host pairs  $\{v, u\}$  indicates that both hosts  $v$  and  $u$  are within their wireless transmitter ranges. Thus the corresponding graph will be an undirected graph.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DIAL M 99 Seattle WA USA

Copyright ACM 1999 1-58113-174-7/99/08...\$5.00

Routing in ad hoc wireless networks requires fast convergence and low communication overhead. Routing information has to be localized to adapt quickly to network topological changes. *Connected-dominating-set-based routing* can be a solution to this kind of network environment. A subset of the vertices (also name as *gateways*) of a graph is a dominating set if every vertex not in the subset is adjacent to at least one vertex in the subset.

The main advantage of connected-dominating-set-based routing is that it centralizes the whole network into small connected dominating set subnetwork, which means only gateway hosts keep routing information, so that as long as network topological changes do not affect this subnetwork there is no need to recalculate routing tables.

Since finding a minimum connected dominating set is NP-complete for most graphs, we propose a simple distributed approximation algorithm that can quickly determine a connected dominating set in a given connected graph, which represents an ad hoc wireless network. We show that proposed approach outperforms a classical approach in terms of complexity and average size of dominating set. We also discuss ways to update and recalculate the dominating set when the underlying graph changes with the movement of mobile hosts. We also briefly describe efficient routing using connected dominating set.

This paper is organized as follows: Section 2 overviews related works. Section 3 presents the proposed approach on finding a small connected dominating set. The connected dominating set updating and recalculation are discussed in Section 4. Section 5 briefly describes routing protocol in ad hoc networks. Performance evaluation is done in Section 6, where our algorithm is compared with the one proposed by Das et al [3]. Finally, in Section 7 we conclude this paper and discuss future work.

## 2 Previous Works

There are numerous routing protocols [1, 2, 4, 3, 5, 7, 8, 9, 10, 11, 12], particularly for wireless networks, are proposed. Among them, [1, 5, 7, 10, 11] have been reviewed in detail in [9]. In the rest of this section, we will concentrate on approaches that are dominating-set-based [2, 4, 3, 8, 9, 12].

The *cluster-based algorithm* [9] divides a given unweighted graph into a number of overlapping, clusters. One (or

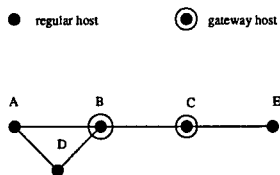


Figure 1: The example of ad hoc wireless network

more) representative node, called a *boundary node*, is selected from each cluster to form a connected dominating set subnetwork. The routing protocol is completed in two phases: *cluster formation* and *cluster maintenance*. During the cluster formation, the network is viewed as a dynamically growing system, in other words, assume that mobile hosts are inserted into the network sequentially. Therefore, each node needs information of the entire network topology. The algorithm runs  $O(\nu)$  rounds, for each round its complexities are  $O(\Delta^3)$  in terms of time and  $O(B + \Delta)$  in terms of messages, where  $\nu$  is the number of hosts in the network,  $B$  is the number of boundary nodes, and  $\Delta$  is the maximum node degree.

Das et al proposed a series of routing algorithms [2, 4, 3, 12] for the ad-hoc wireless network. Similar to the cluster-based routing [9], the idea is to identify a subnetwork that forms a *minimum connected dominating set* (MCDS). Each node in the subnetwork is called a *spine node* or *backbone node* (we call it gateway in our proposed algorithm). Das et al proposed a distributed version of Guha and Khuller's *approximation algorithm* [6] for calculating minimum connected dominating set. This distributed algorithm has been used in all the papers by Das et al [2, 4, 3, 12].

This MCDS calculation algorithm has two main advantages over to the cluster-based approach [9]. First, each node only needs 2-distance neighborhood information, unlike [9], in which each node needs information of the entire network topology. Second, the algorithm runs  $O(\gamma)$  rounds. The over all complexities are  $O(\gamma\Delta^2 + \nu)$  in terms of time and  $O(\Delta\nu\gamma + m + \nu \log \nu)$  in terms of messages, where  $\gamma$  is the number of nodes in the resultant dominating set,  $m$  is the number of edges,  $\nu$  is the number of node, and  $\Delta$  is the maximum node degree. This is an improvement compared to [9] in terms of time complexity although it has higher message complexity over [9] in the worst case. The main drawback of this algorithm is that it still needs a non-constant number of rounds to determine a connected dominating set.

Methods in [2, 4, 3, 12] differ in the way routing tables are constructed and propagated to non-MCDS nodes. The requirement for shortest paths adds one additional dimension of complexity. Because the set of MCDS nodes (dominating set) may not include all intermediate nodes of a shortest path, the routing process cannot be restricted to MCDS nodes. In other words, in order to compute a routing table, each MCDS node needs to know entire network topology. An all-pairs shortest path algorithm is actually running on  $G$ , not on the reduced subnetwork of MCDS nodes. Therefore, it may lose part of the original goal of network centralization.

Another extreme approach, as proposed by Johnson [8], uses *dynamic source routing* without constructing any routing tables. Normally, the resultant routing path is not the shortest. However, this protocol adapts quickly to routing changes when host movement is frequent, yet requires little or no overhead during periods in which hosts move less frequently. The approach consists of *route discovery* and *route maintenance*. Route discovery allows any host to dynamically discover a route to a destination host. Each host also maintains a *route cache* in which it caches source routes that it has learned. Unlike routing table approaches that have to perform periodic routing updates, route maintenance only monitors the routing process and informs the sender of any routing errors.

### 3 Proposed Approach

As mentioned early, we will focus only on formation of a dominating set. Some desirable features are: (1) The formation process should be distributed and simple. Ideally, it requires only local information and constant number of iterative rounds of message exchanges among neighboring hosts. (2) The resultant dominating set should be connected and close to minimum. (3) The resultant dominating set should include all intermediate nodes of any shortest path. In this case, an all-pair shortest paths algorithm only needs to be applied to the subnetwork containing the dominating set.

**Marking process.** We propose a marking process that marks every vertex in a given connected and unweighted graph  $G = (V, E)$ .  $m(v)$  is a marker for vertex  $v \in V$ , which is either T (marked) or F (unmarked). We will show later that marked vertices form a connected dominating set. We assume that all vertices are unmarked initially.  $N(v) = \{u | \{v, u\} \in E\}$  represents the *open neighbor set* of vertex  $v$  and  $v$  has  $N(v)$  initially. The marking process is following: (1) Initially assign marker F to every  $v$  in  $V$ . (2) Every  $v$  exchanges its open neighbor set  $N(v)$  with all its neighbors. (3) Every  $v$  assigns its marker  $m(v)$  to T if there exist two unconnected neighbors.

In the example of Figure 1,  $N(A) = \{B, D\}$ ,  $N(B) = \{A, C, D\}$ ,  $N(C) = \{B, E\}$ ,  $N(D) = \{A, B\}$ , and  $N(E) = \{C\}$ . After the the Step 2 of the marking process. Vertex  $A$  has  $N(B)$  and  $N(D)$ ,  $B$  has  $N(A)$ ,  $N(C)$ , and  $N(D)$ ,  $C$  has  $N(B)$  and  $N(E)$ ,  $D$  has  $N(A)$  and  $N(B)$ , and  $E$  has  $N(C)$ . Based on Step 3, only vertices  $B$  and  $C$  are marked T.

**Properties.** Assume  $V'$  is the set of vertices that are marked T in  $V$ , i.e.,  $V' = \{v | v \in V, m(v) = T\}$ . The reduced graph  $G'$  is the subgraph of  $G$  induced by  $V'$ , i.e.,  $G' = G[V']$ . The following two theorems show that  $G'$  is a dominating set of  $G$  and it is connected.

**THEOREM 1:** *Given a  $G = (V, E)$  that is connected, but not completely connected, the vertex subset  $V'$ , derived from the marking process, forms a dominating set of  $G$ .*

**PROOF:** Randomly select a vertex  $v$  in  $G$ . We show that  $v$  is either in  $V'$  (a set of vertices in  $V$  that are marked T) or adjacent to a vertex in  $V'$ . Assume  $v$  is marked F, if there is at least one neighbor marked T, the theorem is proved. When all its neighbors are marked F, we consider the following two cases: (1) All the other vertices in  $G$  are neighbors of

$v$ . Based on the marking process and the fact that  $m(v)=F$ , all these neighbors must be pairwise connected, i.e.,  $G$  is completely connected. This contradicts to the assumption that  $G$  is not completely connected. (2) There is at least one vertex  $u$  in  $G$  that is not adjacent to vertex  $v$ . Construct a shortest path,  $\{v, v_1, v_2, \dots, u\}$ , between vertices  $v$  and  $u$ . Such a path always exists since  $G$  is a connected graph. Note that  $v_2$  is  $u$  when  $v$  and  $u$  are 2-distance apart in  $G$ , i.e.,  $d_G(v, u) = 2$ . Also,  $v$  and  $v_2$  are disconnected; otherwise,  $\{v, v_2, \dots, u\}$  is a shorter path between  $v$  and  $u$ . Based on the marking process, vertex  $v_1$ , with both  $v$  and  $v_2$  as its neighbors, must be marked T. Again this contradicts to the assumption that  $v$ 's neighbors are all marked F.  $\square$

When the given  $G$  is completely connected, all vertices are marked F. This make sense, since if all vertices are directly connected, there is no need of gateway hosts.

**THEOREM 2:** *The reduced graph  $G' = G - V'$  is a connected graph.*

**PROOF:** We prove this theorem by contradiction. Assume  $G'$  is disconnected and  $v$  and  $u$  are two disconnected vertices in  $G'$ . Assume  $dis_G(v, u) = k+1 > 1$  and  $\{v, v_1, v_2, \dots, v_k, u\}$  is a shortest path between vertices  $v$  and  $u$  in  $G$ . Clearly, all  $v_1, v_2, \dots, v_k$  are distinct and among them there is at least one  $v_i$  such that  $m(v_i) = F$  (otherwise,  $v$  and  $u$  are connected in  $G'$ ). On the other hand, the two adjacent vertices of  $v_i$ ,  $v_{i-1}$  and  $v_{i+1}$ , are not connected in  $G'$  (otherwise,  $\{v, v_1, v_2, \dots, v_k, u\}$  is not a shortest path). Therefore,  $m(v_i) = T$  based on the marking process.  $\square$

Vertices in a dominating set are called *gateway nodes* and vertices outside a dominating set are called *non-gateway nodes*. The next theorem shows that, except for source and destination vertices, all vertices in a shortest path are contained in the dominating set derived from the marking process.

**THEOREM 3:** *The shortest path between any two nodes does not include any non-gateway node as an intermediate node.*

**PROOF:** We prove this theorem also by contradiction. Assume a shortest path between two vertices  $v$  and  $u$  includes a non-gateway node  $v_i$  as an intermediate node, in other words, this path can be represented as  $\{v, \dots, v_{i-1}, v_i, v_{i+1}, \dots, u\}$ . We label the vertex that precedes  $v_i$  on the path as  $v_{i-1}$ , similarly, the vertex that follows  $v_i$  on the path as  $v_{i+1}$ . Because vertex  $v_i$  is a non-gateway node, i.e.,  $m(v_i)=F$ , there must be a connection between  $v_{i-1}$  and  $v_{i+1}$ . Therefore, a shorter path between  $v$  and  $u$  can be found as  $\{v, \dots, v_{i-1}, v_{i+1}, \dots, u\}$ . This contradicts to the original assumption.  $\square$

Since we proposed an approximation algorithm, in some cases, the resultant dominating set is *trivial*, i.e.,  $V = V'$ . For example, any vertex-symmetric graph will generate a trivial connected dominating set using the proposed marking process. However, the marking process is efficient for the ad hoc wireless mobile network in average case. Our simulation results (to be discussed later) confirm this observation.

**Extensions.** In the following, we propose two rules to reduce the size of a connected dominating set generated from the marking process. We first assign a distinct id,  $id(v)$ , to each vertex  $v$  in  $G'$ .  $N[v] = N(v) \cup \{v\}$  is a *closed neighbor set* of  $v$ , as oppose to the open one  $N(v)$ .

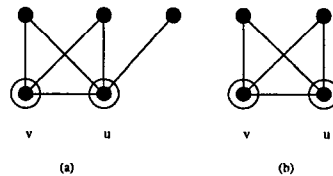


Figure 2: Two samples.

**RULE 1:** *Consider two vertices  $v$  and  $u$  in  $G'$ . If  $N[v] \subseteq N[u]$  in  $G$  and  $id(v) < id(u)$ , change the marker of  $v$  to F if node  $v$  is marked, i.e.,  $G'$  is changed to  $G' - \{v\}$ ,*

The above rule indicates when the closed neighbor set of  $v$  is covered by the one of  $u$ , vertex  $v$  can be removed from  $G'$  if  $v$ 's id is smaller than  $u$ 's. It is easy to prove that  $G' - \{v\}$  is still a connected dominating set of  $G$ . The condition  $N[v] \subseteq N[u]$  implies  $v$  and  $u$  are connected in  $G'$ .

In Figure 2 (a), since  $N[v] \subset N[u]$ , vertex  $v$  is removed from  $G'$  if  $id(v) < id(u)$  and vertex  $u$  is the only dominating node in the graph. In Figure 2 (b), since  $N[v] = N[u]$ , either  $v$  or  $u$  can be removed from  $G'$ . To sure one and only one is removed, we pick the one with a smaller id.

**RULE 2:** *Assume  $u$  and  $w$  are two marked neighbors of marked vertex  $v$  in  $G'$ . If  $N(v) \subseteq N(u) \cup N(w)$  in  $G$  and  $id(v) = \min\{id(v), id(u), id(w)\}$ , then change the marker of  $v$  to F.*

The above rule indicates when the closed neighbor set of  $v$  is covered by the neighbor sets of two of its marked neighbors,  $u$  and  $w$ , if  $v$  has the minimum id of the three, it can be removed from  $G'$ . The condition  $N(v) \subseteq N(u) \cup N(w)$  in Rule 2 implies that  $u$  and  $w$  are connected. The subtle difference between Rule 1 and Rule 2 is the use of open and close neighbor sets. Again, it is easy to prove  $G' - \{v\}$  is still a connected dominating set. Obviously, to apply Rule 2, an additional step needs to be added at the end of the marking process: If a host  $v$  is marked ( $m(v)=T$ ), send its status to all its neighbors.

Consider the example in Figure 3. Clearly,  $N(v) \subseteq N(u) \cup N(w)$ . If  $id(v) = \min\{id(v), id(u), id(w)\}$ , vertex  $v$  can be removed from  $G'$  based on Rule 2. If  $id(u) = \min\{id(v), id(u), id(w)\}$ , then vertex  $u$  can be removed based on Rule 1, since  $N[u] \subseteq N[v]$ . If  $id(w) = \min\{id(v), id(u), id(w)\}$ , no vertex can be removed. Therefore, the id assignment also decides the final outcome of the dominating set. Note that Rule 2 can be easily extended to a more general case where the open neighbor set of vertex  $v$  is covered by the union of open neighbor sets of more than two neighbors of  $v$  in  $G'$ . However, the connectivity requirement for these neighbors is more difficult to specify at vertex  $v$ .

#### 4 Update/Recalculation of Connected Dominating Set in Mobile Networks

We can summarize topological changes of an ad hoc wireless network into three different types: *mobile host's switch on*, *mobile host's switch off*, and *mobile host's movement*. In the

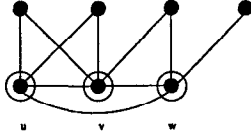


Figure 3: One additional sample.

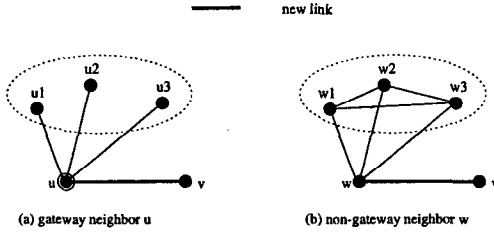


Figure 4: Mobile host  $v$  switches on

rest part of this section we will discuss update/recalculation of connected dominating set for the above cases. Without lost of generality, we assume that the underlying graph of an ad hoc wireless network is always connected.

**Mobile host  $v$ 's switch on.** When a mobile host  $v$  switches on, only its non-gateway neighbors along with host  $v$  need to update their status, because any gateway neighbor will still remain as gateway after a new vertex  $v$  is added. For example, in Figure 4 (a), when host  $v$  switches on gateway neighbor host  $u$  is not affected. Because at least two of the  $u$ 's neighbors  $u_1, u_2$ , and  $u_3$  are not connected originally, and these connections will not be affected by host  $v$ 's switch on. On the other hand, in Figure 4 (b), host  $v$ 's switch on might lead non-gateway neighbor host  $w$  to mark itself as gateway, depending on the connection between host  $v$  and  $w$ 's neighbors  $w_1, w_2$ , and  $w_3$ .

The corresponding marking process can be the following: (1) Mobile host  $v$  broadcasts to its neighbors about its switch on. (2) Each host  $w \in N[v]$  exchanges its open neighbor set  $N(w)$  with its neighbors. (3) Host  $v$  assigns its marker  $m(v)$  to T if there are two unconnected neighbors. (4) Each non-gateway host  $w \in N(v)$  assigns its marker  $m(w)$  to T if it has two unconnected neighbors. (5) Whenever there is a newly marked gateway, host  $v$  and all its gateway neighbors apply *Rule 1* and *Rule 2* to reduce the number of gateway hosts.

**Mobile host  $v$ 's switch off.** When a mobile host  $v$  switches off, only gateway neighbors of that switched off host need to update their status, because any non-gateway neighbor will still remain as non-gateway after vertex  $v$  is deleted. For example, in Figure 5 (a), when  $v$  switches off non-gateway neighbor  $w$  is not affected. The host  $w$ 's neighbors  $w_1, w_2$ , and  $w_3$  are pairwise connected originally, and these pairwise connections will not be affected by host  $v$ 's switch off. On the other hand, in Figure 5 (b), host  $v$ 's switch off might change a gateway neighbor  $u$  to a non-gateway, depending on the connection between its neighbor hosts  $u_1, u_2$ , and  $u_3$ .

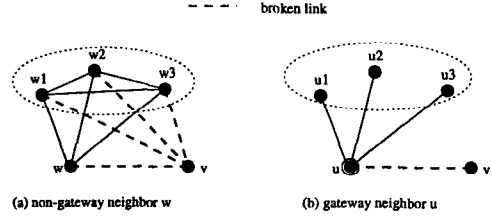


Figure 5: Mobile host  $v$  switches off

The corresponding marking process can be the following: (1) Mobile host  $v$  broadcasts to its neighbors about its switching off. (2) Each gateway neighbor  $w \in N(v)$  exchanges its open neighbor set  $N(w)$  with its neighbors. (3) Each gateway neighbor  $w$  changes its marker  $m(w)$  to F if all neighbors are pairwise connected.

Note that since the underlying graph  $G$  is connected, we can easily prove by contradiction that the resultant dominating set (using the above marking process) is still connected when a host (gateway or non-gateway) switches off.

**Mobile host  $v$ 's movement.** A mobile host  $v$ 's movement can be viewed as several simultaneous or non simultaneous link connections and disconnections. For example, when a mobile host moves, it may lead several link disconnections with its neighbor hosts, and at the same time, it may have new link connections to the hosts within its wireless transmission range, these new links may be disconnected again depending on the way host  $v$  moves.

The challenge here is when and how each vertex should update/recalculate gateway information. The gateway *update* means that only individual mobile hosts update their gateway status. The gateway *recalculation* means that the entire network recalculates gateways/non-gateway status. If many mobile hosts in the network are in movement, gateway recalculation might be a better approach, i.e., the connected dominating set is recalculated from scratch. On the other hand, if only few mobile hosts are in movement, then gateway information can be updated locally. In the remaining discussion, we will focus on the latter case. In order to synchronize mobile host's movement with gateway updates, just before mobile host  $v$  starts to move, it sends out a special signal  $\{id(v), Start\}$ , then during its movement host  $v$  continuously sends out signal  $\{id(v), Heart\_Beat\}$  at every  $\tau$  time interval, and when it stops, host  $v$  sends out signal  $\{id(v), Stop\}$ .

When a host  $u$  receives signal  $\{id(v), Start\}$ , it starts to monitor host  $v$ 's movement. If host  $u$  continuously receives signal  $\{id(v), Heart\_Beat\}$  at every  $\tau$  time interval, and at the end, it receives signal  $\{id(v), Stop\}$ , then no action is needed at host  $u$ . On the other hand, if host  $u$  does not receive a  $\{id(v), Heart\_Beat\}$  or  $\{id(v), Stop\}$  signal after  $\tau$  time interval since last time it received a  $\{id(v), Heart\_Beat\}$  or  $\{id(v), Start\}$  signal, then host  $u$  immediately concludes it has a broken link to host  $v$ , and it will perform certain actions (to be discussed later) for broken link  $\{u, v\}$ .

When a host  $u$  receives signal  $\{id(v), Heart\_Beat\}$  without receiving signal  $\{id(v), Start\}$  previously, host  $u$  can conclude that it has a new link to host  $v$ , and it imme-

diately performs certain actions (to be discussed later) for new link  $\{u, v\}$ . At the same time, host  $u$  continuously monitors host  $v$ 's movement. If host  $u$  continuously receives signal  $\{\text{id}(v), \text{Heart\_Beat}\}$  at every  $\tau$  time interval, and at the end, it receives signal  $\{\text{id}(v), \text{Stop}\}$ , then host  $u$  simply does nothing further. On the other hand, if host  $u$  does not receive a  $\{\text{id}(v), \text{Heart\_Beat}\}$  or  $\{\text{id}(v), \text{Stop}\}$  signal after  $\tau$  time period since last time it received a  $\{\text{id}(v), \text{Heart\_Beat}\}$  or  $\{\text{id}(v), \text{Start}\}$  signal, then host  $u$  immediately concludes it has a broken link to host  $v$ , and similar actions are needed as the case discussed earlier.

For mobile host  $v$  that is in movement, at every  $\tau$  time interval during its movement, it performs the following marking process: (1) Mobile host  $v$  compares its new neighbor set  $N'(v)$  with the original neighbor set  $N(v)$ . If they are the same, host  $v$  simply does nothing further; otherwise, it continues the following steps. (2) Host  $v$  exchanges the open neighbor set with its neighbors. (3) Host  $v$  marks itself as gateway if it has two unconnected neighbors; otherwise, it marks itself as non-gateway. (4) Further, if host  $v$  marked itself as gateway, host  $v$  and all its gateway neighbors apply *Rule 1* and *Rule 2* to reduce the number of gateway hosts in the network.

For neighbor  $u$  of host  $v$ , we consider two subcases: *mobile host  $u$  recognizes a new link  $\{u, v\}$*  and *mobile host  $u$  recognizes a broken link  $\{u, v\}$* .

**Mobile host  $u$  recognizes a new link  $\{u, v\}$ .** When a mobile host  $u$  recognizes a new link  $\{u, v\}$ , two types of mobile hosts need to recalculate their own gateway status. One is mobile host  $u$  itself if it is non-gateway originally; the other ones are common neighbors of mobile hosts  $u$  and  $v$ . For example, in Figure 6 (a), non-gateway host  $u$  may mark itself as gateway after a new link to  $v$  is established, depending on the connection between  $v$  and  $u$ 's neighbors  $u_1, u_2, u_3$ , and  $u_4$ . On the other hand, in Figure 6 (b), at least two of gateway host  $u$ 's neighbors are unconnected originally, new link  $\{u, v\}$  has no effect on these unconnected neighbors; therefore,  $u$  still remains as gateway. In the same figure, common gateway neighbor  $u_4$  of mobile host  $u$  and  $v$  may unmark itself as non-gateway after new link  $\{u, v\}$  is established.

The corresponding marking process can be the following: (1) Mobile host  $u$  detects a new link to  $v$ , and it exchanges the open neighbor set with its neighbors. (2) Upon receiving the open neighbor set  $N(u)$  from host  $u$ , gateway host  $w$  recalculates its status, if it is a common neighbor of hosts  $u$  and  $v$ . (3) If host  $u$  is gateway, it simply does nothing further; otherwise, host  $u$  marks itself as gateway if it has two unconnected neighbors. (4) Whenever there is a newly marked gateway, the newly marked gateway host and its gateway neighbors apply *Rule 1* and *Rule 2* to reduce the number of gateway hosts.

**Mobile host  $u$  recognizes broken link  $\{u, v\}$ .** When a mobile host  $u$  recognizes broken link  $\{u, v\}$ , two types of mobile hosts need to recalculate their gateway status. One is mobile host  $u$  itself if it is a gateway originally; the other ones are common neighbors of mobile hosts  $u$  and  $v$ . For example, in Figure 7 (a), neighbors of non-gateway host  $u$  are all pairwise connected, and they remain so after link  $\{u, v\}$  is broken. On the other hand, in Figure 6 (b),

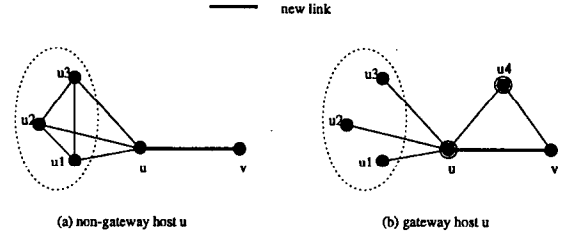


Figure 6: Mobile host  $u$  recognizes new link  $\{u, v\}$

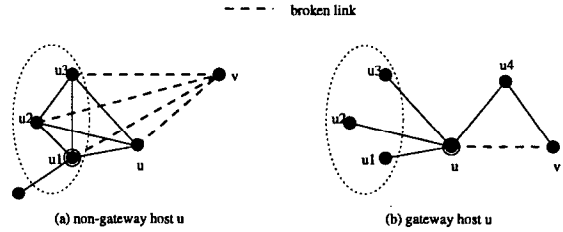


Figure 7: Mobile host  $u$  recognizes broken link  $\{u, v\}$

depending on the connections between neighbors  $u_1, u_2, u_3$ , and  $u_4$ , gateway host  $u$  may unmark itself as non-gateway after link  $\{u, v\}$  is broken. In the same figure, the mobile hosts  $u$  and  $v$ 's common non-gateway neighbor  $u_4$  may mark itself as gateway after link  $\{u, v\}$  is broken.

The corresponding marking process can be the following: (1) Mobile host  $u$  detects a broken link to  $v$ , and it exchanges the open neighbor set with its neighbors. (2) If host  $u$  is non-gateway, it simply does nothing further; otherwise, host  $u$  will assign its marker  $m(u)$  to F if its neighbors are all pairwise connected. (3) Upon receiving the open neighbor set  $N(u)$  from host  $u$ , non-gateway neighbor (excluding host  $v$ )  $w$  recalculates its status if it is a common neighbor of hosts  $u$  and  $v$ . (4) Whenever there is a newly marked gateway, the newly marked gateway host and its gateway neighbors apply *Rule 1* and *Rule 2* to reduce the number of gateway hosts.

The above marking processes are effective only if few nodes have new/broken links with host  $v$ . When there are many such nodes, a better way of updating gateways can be similar to the one for a host's switch on/off. During its movement, host  $v$  continuously sends out its open neighbor set  $N(v)$  along with signal  $\{\text{id}(v), \text{Heart\_Beat}\}$ . This triggers each host  $w \in v \cup N(v)$  to update gateway status.

The corresponding marking process can be the following: (1) Mobile host  $v$  periodically exchanges its open neighbor set with its neighbors at every  $\tau$  time interval. (2) Each host  $w \in v \cup N(v)$  assigns its marker  $m(w)$  to T if it has two unconnected neighbors. (3) Whenever there is a newly marked gateway, the newly marked gateway host and its gateway neighbors apply *Rule 1* and *Rule 2* to reduce the number of gateway hosts.

For those hosts  $u$  that have broken link to host  $v$ , their gateway status are updated as soon as they recognize link

disconnection.

The corresponding marking process can be the following:

(1) Mobile host  $u$  detects a broken link to  $v$ , and it exchanges the open neighbor set with its neighbors. (2) If host  $u$  is non-gateway, it simply does nothing further; otherwise, host  $u$  assigns its marker  $m(u)$  to  $F$  if its neighbors are all pairwise connected.

## 5 Routing in Ad hoc mobile wireless networks using connected dominating set

The routing process can be divided into three steps: (1) If the source is not a gateway host, it forwards the packets to one of its adjacent gateway hosts. (2) This gateway host acts as a new source to route the packets in the *reduced graph* consisting of gateway hosts only. (3) Eventually, the packets reach the destination gateway which is either the destination host itself or a gateway of the destination host. In the latter case, the destination gateway forwards the packets directly to the destination host.

The gateway host keeps following informations: *gateway membership of entire subnetwork, local routing table*. The way the routing tables constructed and updated on the connected dominating set subnetwork can be different. In the rest part of this section, we will briefly discuss two extreme case routing protocols: *shortest path routing* and *dynamic source routing*.

**Shortest Path Routing.** The routing can be implemented as following. When a non-gateway mobile host needs to send out packet, it first sends request to all the available gateway neighbor hosts. Each neighbor gateway hosts will find out which gateways are dominator of the destination node, and by looking at local routing table find out the destination gateway that is minimum distance away from it, then reply this minimum route length to the source. The source node will pick up the one gateway neighbor with shortest path as next hop. The routing tables in the subnetwork domain can be constructed and updated using regular *distance vector protocol* or *link state protocol*.

**Dynamic Source Routing.** We can also implement a dynamic source routing [8] in the connected dominating set subnetwork domain with following slight modification. If the source is non-gateway host, it has maximum  $k_1 \times k_2$  choice of route, on the other hand, if the source is a gateway host, it has maximum  $k_2$  choice of route, where  $k_1$  is the number of source gateway, and  $k_2$  is the number of the destination gateway. The source can pick up any route depends on its transmission criteria. Since a source may have several back up routes, whenever there is routing error occurs, instead of initiating a new route discovery procedure immediately, the source gateway will try to use back up routes to transmit packets. Only when all back up routes fails, the source will initiate rote discovery procedure.

## 6 Performance Evaluation

In this section, we compare our approaches, with Das et al's algorithm[3]. Our comparison was conducted through analytical study and simulation. Simulation for update/recalcu-

lation of connected dominating set in a dynamic network will be part of our future work.

**Algorithm complexity.** The performance can be measured by computation and communication complexity. In our approach, the time complexity of the marking process at each vertex is  $O(\Delta^2)$ , where  $\Delta(G) = \max\{|N(v)||v \in V\}$ . The total amount of message exchanges is  $O(\Delta\nu)$ , where  $\nu = |V|$  is the total number of vertices in  $G$ . More precisely, the marking process without using two rules needs one round, with applying two rules needs two rounds.

On the other hand, Das' algorithm runs  $O(\gamma)$  rounds. The over all complexities are  $O(\gamma\Delta^2 + \nu)$  in terms of time, where  $\gamma$  is the cardinality of the derived dominating set,  $O(\Delta\nu\gamma + m + \nu \log \nu)$  in terms of messages, where  $m$  is the cardinality of the edge set.

Clearly, our approach is less complex than Das's algorithm in all measurements, in particular, the number of rounds needed. Note that the number of rounds is an important metrics measuring the performance of the algorithm, because the topology of the ad hoc wireless network changes frequently with the movement of mobile hosts, the dominating set has to be updated and recalculated frequently.

Another important measurement is the size of the dominating set generated. Although we can not theoretically prove that our algorithm generates smaller connected dominating set than Das' algorithm, which has approximation ratio of  $3H(\Delta)$ , where  $H(\Delta)$  is the  $\Delta$ th harmonic number, we can show that our approach outperform Das' algorithm in average through simulation discussed in the following subsection.

**Simulation.** In this section we perform the simulation which compare the average size of the dominating set generated from our approach with the one from Das' algorithm [3] under different conditions. The smaller size of the dominating set, the better the result.

The simulation is performed using the following parameters.  $\nu$  represents the number of mobile hosts in the network, and  $\gamma$  represents the number of gateways (the size of the dominating set) in the network,  $r$  represents the radius of mobile host's transmission area, New1 is the number of gateway nodes calculated by our algorithm without applying two rules, New2 is the number of gateway nodes calculated by our algorithm with two rules, and MCDS is the number of gateway nodes calculated by Das's algorithm [3].

Random graphs are generated in a  $100 \times 100$  square units of a 2-D simulation area, by randomly throwing a certain number of mobile hosts. Any pair of nodes have link connection if their distance is less than radius  $r$ . If generated graph is disconnected, simply discard the graph. Otherwise continue the simulation.

We performed two groups of simulation. In the first group, we set the radius of the mobile host's transmission area  $r$  to four different values: 15, 25, 50, 75. In this way, we can control the density of generated graphs, since the density of generated graphs increases as  $r$  increases. For each  $r$ , we also vary the number of mobile hosts  $\nu$  from 0 to 100. For each  $\nu$ , generate a random connected graph 1000 times. Calculate New1, New2, and MCDS for each case; and at the end, we simply take the average of New1, New2, MCDS. In the second group, we set the number of mobile hosts  $\nu$

in the network to four different values: 20, 40, 60, 80. For each  $\nu$ , we vary the radius of the mobile host's transmission area  $r$  from 0 to 100. For each  $r$ , generate a random connected graph 1000 times. Calculate New1, New2, and MCDS for each case, and then, take the average of New1, New2, MCDS.

At the current stage, our simulation does not include the movement of mobile hosts, since this paper focuses only on determining a minimum connected dominating set from a given graph.

**Results.** Figures 8 (a),(b),(c),(d) show the number of gateways versus the number of nodes in the network for the increasing order of radius  $r$ . We can see that without applying two rules, the performance of our algorithm (the curve for New1) fares poorly. However, by applying two rules, the performance of our approach (the curve for New2) is much better than Das' algorithm [3] (the curve for MCDS), when mobile host's transmission area is neither too small nor too large, for example,  $r = 15, 25, 50$ . We can see the gap between New2 and MCDS increases as  $r$  increases. When  $r=75$ , our approach (with applying two rules) is suddenly outperformed by Das' algorithm. In order to fully understand this sudden change of relative performance between these two approaches, we conducted the second group of simulation.

Figures 9 (a),(b),(c),(d) show the number of gateway nodes with respect to radius  $r$  for the increasing order of number of nodes  $\nu$ . We can see that the number of gateway nodes decreases smoothly as the radius  $r$  increases using our proposed algorithm with applying two rules; however, using Das' algorithm the number of gateway nodes remains almost unchanged until the radius  $r$  reaches around 60, then it suddenly decreases to stay under the curve for New2. Although for a large radius  $r$ , Das' algorithm shows better performance, the difference between New2 and MCDS is no more than 2 gateways, it does not affect the overall comparison result.

## 7 Conclusions

In this paper, we have proposed a simple and efficient distributed algorithm for calculating connected dominating set in the ad-hoc wireless network. A simulation study has been conducted to compare our proposed algorithm with Das' algorithm [3] in terms of the size of connected dominating set generated. When the mobile host's transmission radius is not too large, the proposed algorithm generates a smaller connected dominating set. Our proposed algorithm calculates connected dominating set in  $O(\Delta^2)$  time with 2-distance neighborhood information, where  $\Delta$  is the maximum node degree in the graph. In addition, the proposed algorithm uses constant (1 or 2) rounds of message exchanges, compared with  $O(\gamma)$  rounds of message exchanges in Das' algorithm, where  $\gamma$  is the size of the dominating set.

The future works will extend the proposed algorithm to the ad-hoc wireless networks in which each mobile node has different transmission radius. We need to verify the effectiveness of our update/recalculation strategies when the topology of the underlying network changes.

## References

- [1] M. S. Corson and A. Ephremides. A distributed routing algorithm for mobile wireless networks. *ACM J. Wireless Networks*, 1(1):61 – 81, 1995.
- [2] B. Das and V. Bhargavan. Routing in ad-hoc networks using minimum connected dominating sets. *IEEE International Conference on Communications (ICC '97)*, June 1997.
- [3] B. Das, E. Sivakumar, and V. Bhargavan. Routing in ad-hoc networks using a virtual backbone. *Proceedings of the 6th International Conference on Computer Communications and Networks (IC3N '97)*, pages 1 – 20, Sept. 1997.
- [4] B. Das, R. Sivakumar, and V. Bhargavan. Routing in ad-hoc networks using a spine. *IEEE International Conference on Computers and Communications Networks '97*, 1997.
- [5] E. Gafni and D. P. Bertsekas. Distributed algorithms for generating loop-free routes with frequently changing topology. *IEEE Transactions on Communications*, COM-29(1):11 – 18, 1981.
- [6] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374 – 387, April 1998.
- [7] D. B. Johnson. Routing in ad hoc networks of mobile hosts. *Proceedings of Workshop on Mobile Computing Systems and Applications*, pages 158 – 163, December 1994.
- [8] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad-hoc wireless networks. In T. Imielinski, H. Korth, editor, *Mobile Computing*. Kluwer Academic Publishers, 1996.
- [9] P. Krishna, M. Chatterjee, N. H. Vaidya, and D. K. Pradhan. A cluster-based approach for routing in ad-hoc networks. *Proceedings of the Second USENIX Symposium on Mobile and Location-Independent Computing*, pages 1 – 10, 1995.
- [10] S. Murthy and J. J. Garcia-Luna-Avecas. A routing protocol for packet radio networks. *Proceedings of ACM International Conference on Mobile Computing and Networking*, pages 86 – 95, November 1995.
- [11] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers. *Computer Communications Review (ACM SIGCOMM 1994)*, 24(4):234 – 244, 1994.
- [12] R. Sivakumar, B. Das, and V. Bhargavan. An improved spine-based infrastructure for routing in ad hoc networks. *Proceedings of the International Symposium on Computers and Communications (ISCC'98)*, 1998.

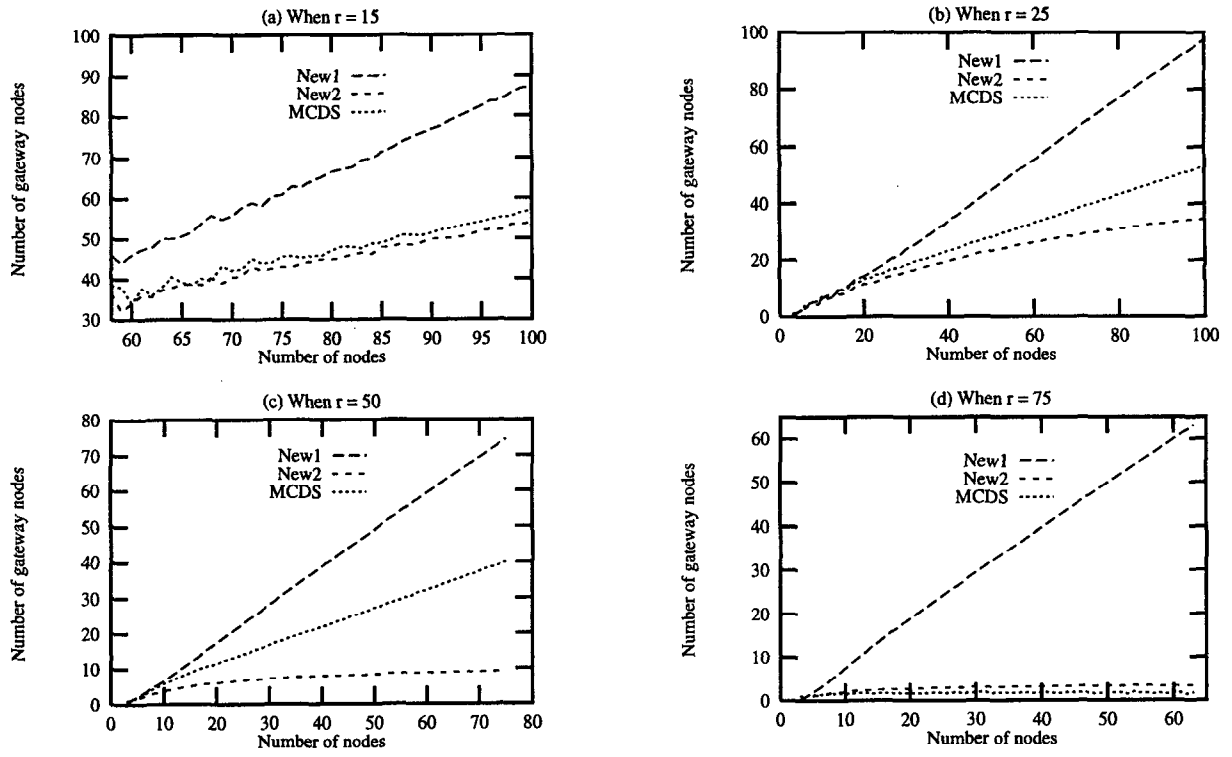


Figure 8: Average number of gateway nodes relative to the number of nodes  $\nu$ .

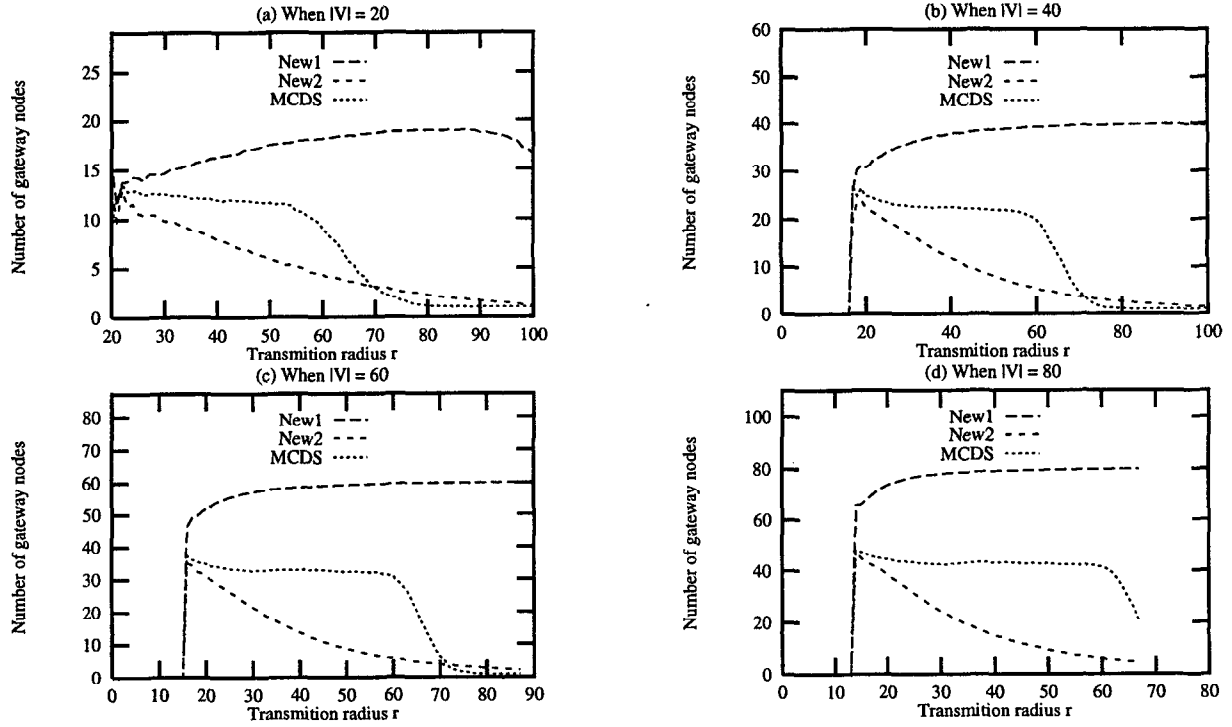


Figure 9: Average number of gateway nodes relative to the radius  $r$ .