

Extended Dominating-Set-Based Routing in Ad Hoc Wireless Networks with Unidirectional Links

Jie Wu, *Senior Member, IEEE*

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 promising approach, where the search space for a route is reduced to the 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 extend dominating-set-based routing to networks with unidirectional links. Specifically, an efficient localized algorithm for determining a dominating and absorbant set of vertices (mobile hosts) is given and this set can be easily updated when the network topology changes dynamically. A host v is called a dominating neighbor (absorbant neighbor) of another host u if there is a directed edge from v to u (from u to v). A subset of vertices is dominating and absorbant if every vertex not in the subset has one dominating neighbor and one absorbant neighbor in the subset. The derived dominating and absorbant set exhibits good locality properties; that is, the change of a node status (dominating/dominated) affects only the status of nodes in the neighborhood. The notion of dominating and absorbant set can also be applied iteratively on the dominating and absorbant set itself, forming a hierarchy of dominating and absorbant sets. The effectiveness of our approach is confirmed and the locality of node status update is verified through simulation.

Index Terms—Ad hoc wireless networks, dominating sets, hierarchical structures, mobility management, routing, simulation.

1 INTRODUCTION

An *ad hoc wireless network* is a special type of wireless mobile network in which a collection of mobile hosts with wireless network interfaces forms a temporary network, without the aid of any established infrastructure (i.e., base stations) or centralized administration (i.e., mobile switching centers). The applications of ad hoc wireless networks range from civilian use (distributed computing and sensor networks [9]) to disaster recovery (search-and-rescue) and military use (battlefield). We can use a simple 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. An edge between a host pair $\{v, u\}$ indicates that both hosts v and u are within their wireless transmitter ranges; that is, connections of hosts are based on geographic distances of hosts. Thus, the corresponding graph is an undirected graph and is called a *unit disk graph*, or simply, *unit graph*.

Routing in ad hoc wireless networks poses special challenges. In general, the main characteristics of mobile computing are low power, limited bandwidth, and unrestricted mobility. Wireless networks deliver lower bandwidth than wired networks and, hence, information collection (during the formation of a routing table) is expensive. Traditional routing protocols in wired networks, which generally use either *link state* [18], [20] or *distance vector* [14], [19], are no longer suitable for ad hoc wireless networks. In

an environment with mobile hosts as routers, convergence to new, stable routes after dynamic changes in network topology may be slow and this process could be expensive due to low bandwidth. Routing information has to be localized to adapt quickly to changes such as host movement. Mobility of hosts, which causes topological changes of the underlying network, also increases the volatility of network information. The tunnel-based triangle routing of mobile IP [23] and cellular IP [6] works well if there is a fixed infrastructure to support the notion of “home agent.” However, when all hosts move (including the home agent), such a strategy cannot be directly applied.

Routing protocols in ad hoc wireless networks fall into either *proactive* [10] or *reactive* [5] categories, although a combination of proactive and reactive is also possible [13]. In proactive routing, routes to all destinations are computed a priori and are maintained in the background via a periodic update process. Route information is maintained either as routing tables (used in distance vector protocols) or as global link state information (used in link state protocols). In reactive routing, route to a specific destination is computed “on demand,” that is, only when needed. To efficiently use resources in controlling large dynamic networks, hierarchical routing [28] is normally used. Among hierarchical routing, *dominating-set-based* routing (and the similar *cluster-based* routing) is a promising approach. A subset of the vertices of a graph is a dominating set if every vertex not in the subset is adjacent to at least one vertex in the subset. This dominating set should be connected for ease of the routing process within the induced graph consisting of dominating nodes only. Vertices in a dominating set are also called *gateway* hosts while vertices that are outside a dominating set are called *nongateway* hosts. In the example shown in Fig. 1, hosts v

• The author is with the Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431.
E-mail: jie@cse.fau.edu.

Manuscript received 16 June 2001; accepted 8 Apr. 2002.
For information on obtaining reprints of this article, please send e-mail to: tpd@computer.org, and reference IEEECS Log Number 116278.

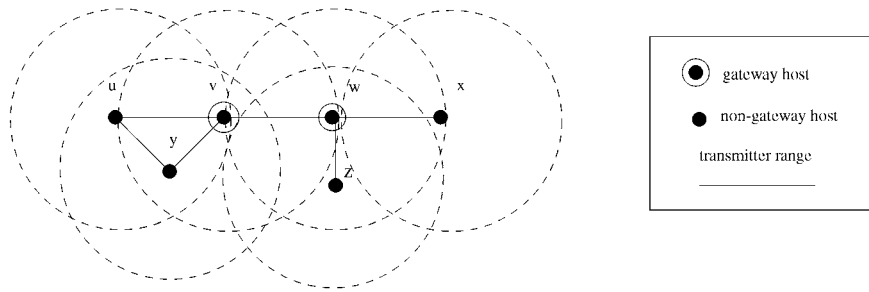


Fig. 1. A sample unit graph representing an ad hoc wireless network.

and w form a connected dominating set of the given unit graph. The main advantage of dominating-set-based routing is that it simplifies the routing process to the one in a smaller subnetwork generated from the connected dominating set. This means that only gateway hosts need to keep routing information in a proactive approach and the search space is reduced to the dominating set in a reactive approach. Note that gateway hosts are used not only to route packets but also to disseminate routing information. Clearly, the efficiency of this approach depends largely on the process of finding and maintaining a connected dominating set and the size of the corresponding subnetwork. Unfortunately, finding a minimum connected dominating set is NP-complete for most graphs.

Wu and Li [33] proposed a simple and efficient distributed algorithm that can quickly determine a connected dominating set in ad hoc wireless networks. This approach uses a *localized algorithm* called the *marking process* [9] where hosts interact with others in a restricted vicinity. Each host performs exceedingly simple tasks such as maintaining and propagating information *markers*. No information is sequentially propagated in the network. Specifically, each host is marked true if it has two unconnected neighbors. It has been shown that, collectively, these hosts achieve a desired global objective—a set of marked hosts forms a small connected dominating set. This approach also outperforms several classical approaches in terms of finding a small connected dominating set and/or does so quickly [33].

In this paper, we extend the dominating-set-based routing to ad hoc wireless networks with unidirectional links. In an ad hoc wireless network, some links may be unidirectional due to either the disparity of transmitter ranges of hosts or the *hidden terminal problem* [30]. In a hidden terminal problem, two packets sent from different hosts to the same destination collide. As a result, one or both packets are lost. For example, x in Fig. 1 can receive signals from w , but w may not be able to receive signals from x due to the interference of signals sent from v or z . In this case, the connection between w and x is unidirectional (at least temporarily). In a network with unidirectional links, the domination concept has to be redefined. Specifically, an ad hoc wireless network is represented as a directed graph, $D = (V, A)$, consisting of a finite set V of vertices and a set A of directed edges. A host v in V is called a *dominating neighbor* (*absorbant neighbor*) of another host u in V if there is a directed link (v, u) ((u, v)). A subset of vertices (mobile hosts) is dominating and absorbant if every vertex not in the subset has one dominating neighbor and one absorbant neighbor in the subset. A special localized

algorithm called the *extended marking process* is proposed. This algorithm needs only two or three rounds of information exchanges to determine a connected dominating and absorbant set. We also consider applying the extended marking process iteratively to form a hierarchy of dominating and absorbant sets. The effectiveness of the extended marking process, in terms of finding a small dominating and absorbant set, is verified through a simulation study. We show that the dominating and absorbant set can be easily updated when the network topology changes dynamically. It is shown that the derived dominating and absorbant set exhibits good locality properties.

The paper is organized as follows: Section 2 overviews the related work and summarizes our preliminary results on undirected graphs. Section 3 presents the extended marking process that calculates a small dominating and absorbant set in a directed graph. Several implementation issues are also discussed. Dominating-set-based routing in networks with unidirectional links is discussed in Section 4. A comprehensive comparison between dominating-set-based routing and cluster-based routing is also given. Methods to update the dominating and absorbant set when the network topology changes are elaborated in Section 5. Section 6 shows simulation results. The paper concludes in Section 7. Throughout the paper, we use the terms host, node, and vertex interchangeably, similarly for link and edge.

2 RELATED WORK AND PRELIMINARIES

2.1 Related Work

Various design choices are available for designing routing protocols for ad hoc wireless networks. Other than proactive and reactive approaches, other choices include: 1) flat versus hierarchical and 2) GPS-based versus non-GPS-based. Royer and Toh [25] gave other classifications of routing protocols for ad hoc wireless networks. In a flat routing scheme, all hosts are treated equally and, therefore, any host can be used to forward packets between arbitrary sources and destinations. In general, a set of homogeneous processes is applied at each host. These processes include information collection, mobility management, and routing. To permit scaling, hierarchical techniques are usually applied. The major advantage of hierarchical routing is the reduction of routing table storage and processing (including searching) overhead [29]. In non-GPS-based routing, the routing process is based solely on the connections of hosts in the network. In GPS-based routing (also location-aware routing), each host knows its physical location by geolocation techniques such as GPS. Routing is governed by the physical location of the destination.

Among hierarchical routing schemes, Das et al. [7] and Sivakumar et al. [27], proposed a series of 2-level hierarchical routing algorithms for ad hoc wireless networks. 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 (gateway host in the dominating-set-based approach) and keeps a routing table that captures the topological structure of the whole network. The formation of MCDS is based on Guha and Khuller's approximation algorithm [12]. In this approach, a connected dominating set is found by *growing* a tree T starting from a vertex with the maximum node degree. Then, a vertex v in T that has the maximum number of neighbors not in T is *selected*. Finally, a spanning tree is constructed and nonleaf nodes form a connected dominating set. The main drawback of this algorithm is that it still needs a nonconstant number of rounds to determine a connected dominating set, this is because the process of "growing a tree" is a sequential process.

Lin and Gerla [17] provided a general discussion on various clustering algorithms and discussed the following simple clustering algorithm: First, a distributed head selection process is applied. A node v is a *head* if it has the largest *id* (or maximum node degree) in its 1-hop neighborhood including v . A head and its neighbors form a cluster and these nodes are *covered*. Note that, in a cluster, normally only the head connects to all other nodes in the cluster, whereas, in a clique nodes, are pairwise connected. The above process continues on all uncovered nodes. Once the head selection process completes, gateway nodes that have two or more neighbors belonging to different clusters are selected and used to connect clusters. Head nodes form a *unconnected dominating set* (in fact no heads are connected). Head nodes and gateway nodes together form a connected dominating set. However, the process for selecting head nodes may have to be serialized in some cases, such as in a linear network with monotonically increasing or decreasing *id* distribution along the network. Variations to the above clustering algorithm exist [3]. For example, a node v can be the head even if it does not have the largest *id* in its 1-hop neighborhood, as long as there exists at least one neighbor u such that v is the largest *id* in u 's 1-hop neighborhood.

The origin of the dominating set concept can trace back to the 1850's, when the following problem was considered among chess enthusiasts in Europe: *Determine the minimum number of queens that can be placed on a chessboard so that all squares are either attacked by a queen or are occupied by a queen*. It was found that five is the minimum number of queens that can dominate all of the squares of an 8×8 chessboard. The *five queen problem* is about the placement of these five queens. Extensive work has been done in the theoretical community on finding good approximation of minimum connected dominating set (MCDS) in terms of small *approximation ratio* with respect to the domination number. The *domination number* for a given simple graph is the size of the minimum dominating set. Finding the domination number for a given graph is an NP-complete problem. Therefore, most research in the graph theory community focuses on bounds on the domination number. Heuristic rules (such as greedy algorithms) are widely used to find a small dominating set, although other methods such as dynamic programming [15], linear programming, and

linear programming relaxation [21] are also used. However, almost all results have no or little practical use in ad hoc wireless networks.

Recently, a distributed solution with a constant approximation ratio is proposed for ad hoc wireless networks [1] and it can be simplified as follows: First of all, a spanning tree rooted at v (selected through an election process) is first constructed. Nodes are labeled according to a topological sorting order of the tree. Then, nodes are marked based on their positions in the order starting from root v . All nodes are marked white initially, except the root v which is marked black. Following the order, each node is marked black unless there are black neighbors. Clearly, black nodes form a *maximal independent set*. In an independent set, nodes are not neighbors. Then, each black node (except root v) selects a neighbor with the largest label (but smaller than its own label) and marks it gray. Black and gray nodes form a connected dominating set with an approximation ratio of at most 8. Unfortunately, this scheme has to construct a global infrastructure (spanning tree) before the node selection process. Also, this scheme is serialized in both spanning tree construction and dominating node selection. In addition, this approach does not support "locality of maintenance:" A single change in network topology may destroy the spanning tree and, hence, the complete dominating set has to be reconstructed.

2.2 Preliminary Work

Wu and Li [33] conducted some preliminary work on the formation of a dominating set from an undirected graph and on a preliminary routing scheme within the induced graph from the connected dominating set. A *marking process* marks every vertex in a constant number of rounds in a given connected and simple graph $G = (V, E)$ that represents an ad hoc wireless network. $m(v)$ is a marker for vertex $v \in V$, which is either T (marked) or F (unmarked). The set of vertices that are marked T forms a connected dominating set. Assume that $N(v)$ represents the neighbor set of vertex v , and v has $N(v)$ initially. The marking process consists of three steps: 1) Initially assign marker F to every v in V . 2) Every v exchanges its neighbor set $N(v)$ with all its neighbors. 3) Every v assigns its marker $m(v)$ to T if there exist two unconnected neighbors.

Applying the marking process to the example in Fig. 1, vertices v and w are marked T and the rest are marked F . Clearly, each vertex knows 2-hop neighborhood information after Step 2 of the marking process; that is, its neighbor's neighbor information. The cost of checking the connectivity of two neighbors is upper bounded by Δ^2 set coverage checks, where Δ is the vertex degree of graph G . It is shown in [33] that the marked vertices form a dominating set of G . In addition, G' , induced from the dominating set, is connected and contains all the intermediate nodes of any shortest path between two vertices in G . Note that G is the simple graph without self-loop or multiple edges and it includes the unit graph as a special case.

The problem of determining a minimum dominating set of a given unit graph is also NP-complete [2]. The connectivity requirement adds another dimension of difficulty. Therefore, the connected dominating set derived from the marking process is not minimum or bounded. In [33], two extensions are given to enhance the marking process to reduce the size of the connected dominating set

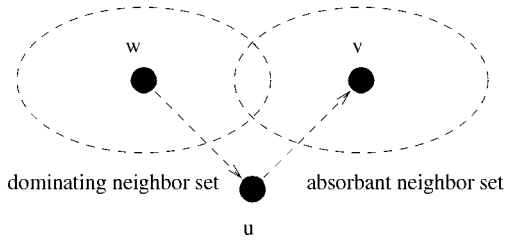


Fig. 2. Dominating (absorbant) neighbor set of vertex u .

generated from the marking process. A distinct label, $id(v)$, is assigned to each vertex v . Vertices are removed from the dominating set derived from the marking process by comparing the neighbor sets and vertex labels of adjacent vertices in the set.

With the notion of dominating set, the routing process in an ad hoc wireless network can be simplified. If a proactive approach is used, nodes and only nodes in the dominating set need to keep routing information (typically presented in a routing table associated with each node). If a reactive approach is applied, the dynamic search process for a route is restricted to the dominating set only. The dominating set can also be used in GPS-based routing. In [8], Datta et al. showed that the performance of a GPS-based greedy routing approach with guaranteed delivery can be greatly improved when the search space is limited to the dominating set.

3 DOMINATION IN DIRECTED GRAPHS

A *directed graph* $D = (V, A)$ consists of a finite set V of vertices and a set A of directed edges, where $A \subset V \times V$. D is a simple graph without self-loop or multiple edges. A directed edge (also called a unidirectional link) from u to v is denoted by an ordered pair (u, v) . If (u, v) is an edge in D , we say that u dominates v and v is an absorbant of u . A set $V' \subset V$ is a *dominating set* of D if every vertex $v \in V - V'$ is dominated by at least one vertex $u \in V'$. Also, a set $V' \subset V$ is called an *absorbant set* if for every vertex $u \in V - V'$, there exists a vertex $v \in V'$ which is an absorbant of u . The *dominating neighbor set* of vertex u is defined as $\{w : (w, u) \in A\}$. The *absorbant neighbor set* of vertex u is defined as $\{v : (u, v) \in A\}$. Fig. 2 illustrates the dominating (absorbant) neighbor set of vertex u . These two neighbor sets may overlap with each other. A directed graph D is strongly connected if for any two vertices u and v , there exists a (u, v) -path (i.e., a path connecting u to v). Throughout the paper, it is assumed that D is strongly connected. If it is not strongly connected, the network management subsystem will partition the network into a set of independent subnetworks, each of which is strongly

connected. The objective here is to quickly find a small set that is both dominating and absorbant in a given directed graph. The absorbant subset may overlap with the dominating subset. In an undirected graph, these two concepts are the same and, hence, a dominating set is also an absorbant set.

3.1 Extended Marking Process

To determine a set that is both dominating and absorbant, we propose an *extended marking process* (see Fig. 3, where $m(u)$ is a marker for vertex $u \in V$, which is either T (marked) or F (unmarked). Basically, a node is marked whenever it is on the shortest path from one neighbor to another. We will show later that the marked set is both dominating and absorbant.

Fig. 4a shows four gateway hosts (4, 7, 8, and 9) derived from the extended marking process. Arrow dashed lines correspond to unidirectional links and solid lines represent bidirectional links. A bidirectional link vu can be considered as two unidirectional links (v, u) and (u, v) . Fig. 5 shows three assignments of u , with one dominating neighbor w and one absorbant neighbor v . The only case in Fig. 5 with $m(u) = F$ is when $(w, v) \in A$, for every dominating neighbor w and every absorbant neighbor v of u . The fourth case, where v and w are bidirectionally connected (a combination of Figs. 5a and 5b), is not shown. Assume that $V' = \{u : u \in V, m(u) = T\}$. The induced graph D' is the subgraph of D induced by V' (i.e., $D' = D[V']$).

Theorem 1. V' includes all the intermediate vertices of any shortest path.

Proof. We prove this theorem by contradiction. Arbitrarily select two vertices v and u ; the strongly connected condition ensures a shortest path from v to u . Assume such a path includes an unmarked vertex v_i as an intermediate vertex: $(v, \dots, v_{i-1}, v_i, v_{i+1}, \dots, u)$. Because $m(v_i) = F$, $(v_{i-1}, v_{i+1}) \in A$ based on the extended marking process. Therefore, a shorter path $(v, \dots, v_{i-1}, v_{i+1}, \dots, u)$ can be found. This contradicts the original assumption. \square

Theorem 2. The induced graph $D' = D[V']$ is a strongly connected graph.

Proof. Arbitrarily select two vertices v and u ; assume that (v, v_1, \dots, v_k, u) is a shortest path from v to u in D . This path is also a path in D' based on Theorem 1. Therefore, Theorem 2 holds. \square

Theorem 3. V' is empty if and only if D is completely connected.

Proof. If D is completely connected, all vertices will remain unmarked based on the extended marking process; that is, V' is empty. On the other hand, if V' is empty, assume D is not completely connected. Without loss of

Extended marking process:

1. Initially assign F to each $u \in V$.
2. u changes its marker $m(u)$ to T if there exist vertices v and w such that $(w, u) \in A$ and $(u, v) \in A$, but $(w, v) \notin A$.

Fig. 3. Extended marking process.

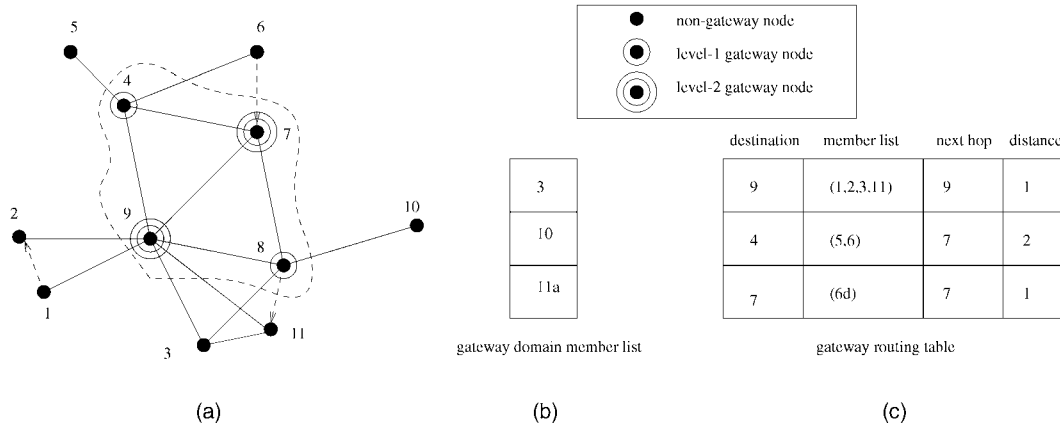


Fig. 4. (a) A sample directed graph. (b) Gateway domain membership list at host 8. (c) Gateway routing table at host 8.

generality, assume $(v, u) \notin A$. Construct a shortest path from v to u : $(v, v_1 \dots, v_k, u)$. $v_k \in V'$ based on Theorem 1. This brings a contradiction. \square

When D is completely connected, all vertices are marked F. This makes sense because, if all vertices are directly connected, there is no need of using dominating and absorbant set to reduce D .

Theorem 4. *If D is not completely connected, V' forms a dominating and absorbant set.*

Proof. We only need to prove that V' forms a dominating set, because the same procedure can be applied to prove that it is also an absorbant set. Arbitrarily select an unmarked vertex u in V if there is at least one vertex v in D that is not a dominating neighbor of vertex u . Construct a shortest path: $(v, v_1 \dots, v_k, u)$. $v_k \in V'$ (based on Theorem 1) is a marked dominating neighbor of u . If all the other vertices are dominating neighbors of vertex u and one of them is marked T, the theorem is proven; otherwise, all vertices (including u) are marked F. Based on Theorem 3, D is a completely connected graph. This brings a contradiction. \square

The selection of u in the above proof is arbitrarily selected from V (which includes V'). Therefore, we have the following corollary.

Corollary 4. *If D is not completely connected (i.e., V' is not empty), each vertex in V' has a dominating neighbor and an absorbant neighbor in V' .*

Theorems 1, 2, 3, and 4 serve as bases of the dominating-set-based routing. The dominating and absorbant set derived from the extended marking process has the desirable properties of connectivity (Theorem 2) and

routing optimality (Theorem 1). The simplicity property, which is crucial in ad hoc wireless networks, is discussed in the section on implementation. In general, the derived dominating and absorbant set is not minimum or bounded, but it can be reduced as discussed in the next section. Simulation results (in Section 5) show that, on average, a relatively small dominating and absorbant set is derived by applying the extended marking process with its two extensions. Constant approximation ratio can be obtained by applying the extended marking process to the cluster-based or GPS-based routing as will be discussed in Section 4.

3.2 Extensions

In the following, we propose two rules to reduce the size of a connected dominating and absorbant set generated from the extended marking process. We first randomly assign a distinct label, $id(v)$, to each vertex v in V . $N_d(u)$ ($N_a(u)$) represents the dominating (absorbant) neighbor set. The neighbor set $N(u)$ is the union of the corresponding dominating neighbor and absorbant neighbor sets; that is, $N(u) = N_a(u) \cup N_d(u)$. Vertex u is called neighbor of vertex v if u is a dominating, absorbant, or dominating and absorbant neighbor of v . Again, V' is the marked set after applying the extended marking process and D' is the corresponding induced graph.

Rule 1. Assume that u is a marked vertex in V' and v is a vertex in V . Unmark u if both conditions hold in D :
 1) $N_d(u) - \{v\} \subseteq N_d(v)$ and $N_a(u) - \{v\} \subseteq N_a(v)$ in D .
 2) $id(u) < id(v)$.

The above rule indicates that, when the dominating (absorbant) neighbor set of u (excluding v) is covered by the dominating (absorbant) neighbor set of v , vertex u can be

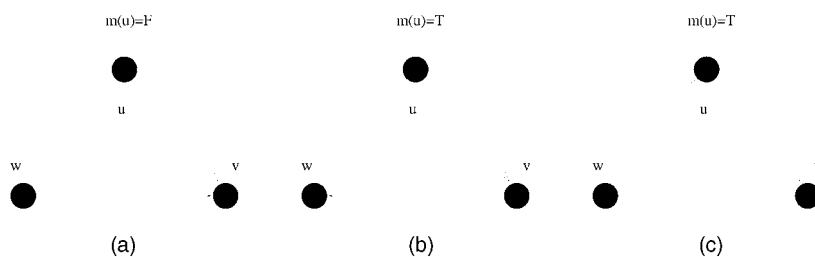


Fig. 5. Marker of u for three different situations.

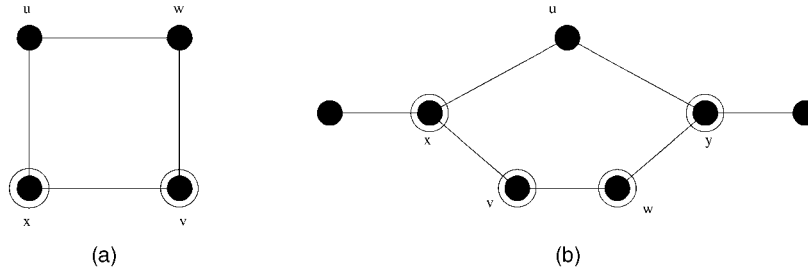


Fig. 6. Gateway status after applying (a) Rule 1 and (b) Rule 2.

removed from V' if u 's id is smaller than v 's. Note that u and v may or may not be (bidirectionally or unidirectionally) connected. In Rule 1, vertex v is not required to be in V' and, hence, there is no need of exchanging marking status among neighbors after the marking process. In fact, v is marked since there are two unconnected neighbors of u and the neighbor set of v covers the neighbor set of u . The role of id is very important to avoid *illegal simultaneous removal* of vertices in V' when Rule 1 is applied simultaneously to each vertex. In general, vertex u cannot be removed even if $N_d(u) - \{v\} \subseteq N_d(v)$ and $N_a(u) - \{v\} \subseteq N_a(v)$ in D , unless $id(u) < id(v)$. Consider a graph of four vertices, u , v , w , and x , with four undirected edges uw , vw , vx , and ux as shown in Fig. 6a. All four vertices will be marked using the extended marking process. Also, $N_d(u) = N_d(v) = N_a(u) = N_a(v) = \{w, x\}$ ($N_d(w) = N_d(x) = N_a(w) = N_a(x) = \{u, v\}$). Without using id , both u and v (also w and x) will be unmarked, leaving no marked vertex. With id , one of u and v (also one of w and x) will be unmarked (and such a vertex is called *ex-gateway*), leaving two marked vertices (v and x based on Rule 1). In Fig. 6a double-cycled nodes are gateway nodes after applying Rule 1. Note that other metrics can be used to break a tie, for example, node degree and geographical location of vertex in a particular dimension.

Rule 2. Assume that u is a marked vertex in V' and v and w are vertices in V . Unmark u if all conditions hold in D : 1) $N_d(u) - \{v, w\} \subseteq N_d(v) \cup N_d(w)$ and $N_a(u) - \{v, w\} \subseteq N_a(v) \cup N_a(w)$ in D . 2) $id(u) = \min\{id(v), id(w)\}$. 3) v and w are bidirectionally connected.

The above rule indicates that when the dominating (absorbant) neighbor set of u (excluding v and w) is covered by the union of dominating (absorbant) sets of v and w , vertex u can be removed from D' if u 's id is smaller than v 's and w 's. Note that u and v (w) may or may not be connected, but v and w must be connected. Again, u does not need to know the marking status of v and w before applying Rule 2. The coverage condition guarantees that both v and w are marked as long as Rule 2 is a nontrivial extension of Rule 1. Rule 2 is said to be a *nontrivial extension* of Rule 1 if the neighbor set of u is not covered by the neighbor set of v or w alone. In this case, there exists a neighbor x of u covered by v but not by w , making u a marked vertex. Similarly, there exists a neighbor y of u covered by w but not by v , making w a marked vertex. Fig. 6b shows an example of applying Rule 2. Gateway u can be unmarked since its neighbor set is

covered jointly by the neighbor sets of v and w . Note that, in this case, u is not a neighbor of either v or w .

Fig. 7 shows another example of using the extended marking process and its extensions to identify a set of connected dominating and absorbant nodes. Fig. 7a shows the gateway nodes (nodes with double cycles) derived by the extended marking process without applying two rules. Fig. 7b shows the remaining gateway nodes after applying two rules. In Fig. 7b, node 5 is unmarked based on Rule 1 where $u = 5$ and $v = 6$. Node 8 is unmarked based on Rule 2 where $u = 8$, $v = 10$, and $w = 13$. Node 10 is unmarked based on Rule 1 where $u = 10$ and $v = 12$.

Assume that V'_* is the resultant dominating and absorbant set after Rule 1 and Rule 2 have been applied to all vertices in V' . The following result shows that V'_* (with its induced graph D'_*) is still a connected dominating and absorbant set of V . The shortest path property of Theorem 3 still holds in D'_* for Rule 1, but not for Rule 2. In addition, each vertex only needs to apply Rule 1 and Rule 2 once and it is final; that is, the marked set cannot be further reduced by Rule 1 or Rule 2.

Theorem 5. If V' is a strongly connected dominating and absorbant set of D derived by using the extended marking process, then V'_* derived by applying Rule 1 and Rule 2 on all vertices in V' has the following properties:

1. Each vertex only needs to apply Rule 1 and Rule 2 once.
2. If V'_* is derived by applying Rule 1 alone, then V'_* still includes all intermediate vertices of at least one shortest path for any pair of vertices in V .
3. If V'_* is derived by applying Rule 2 alone, for each unmarked vertex u , the length of a shortest path (that includes u as an intermediate node before it is unmarked) increases by at most one.
4. V'_* is a dominating and absorbant set of V and D'_* is strongly connected.

Proof. We use the following process of *sequential removal* to emulate one application of Rule 1 and Rule 2 on all vertices: Vertices that are unmarked by Rule 1 or Rule 2 are first sorted in an *ascending order* of node id . Then, vertices in the sorted list are removed one by one, with one per round. At each round, the vertex with the smallest id is removed from the list. Assume that vertex u is selected at round k . V' is the marked set before round k and it is assumed to be a connected dominating and absorbant set of V . $V'_u = V' - \{u\}$ is the marked set after round k . Clearly, V'_* is the resultant marked set after all

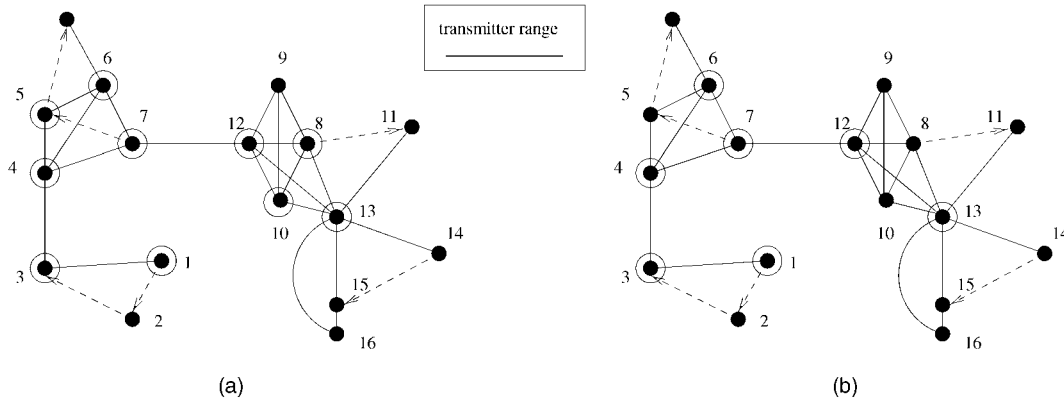


Fig. 7. Marked gateways after applying (a) extended marking process and (b) Rules 1 and 2.

rounds are completed. The above sequential removal is a correct emulation because, when u is removed, the corresponding gateway v (in Rule 1 and Rule 2) and w (in Rule 2) are guaranteed gateway status because the ids of v and w are larger than that of u .

Part 1. Assume that V'_* is the marked set after applying Rule 1 and Rule 2. If V'_* can be further reduced by applying Rule 1 or Rule 2, say u is unmarked later, then based on the definition of Rule 1 and Rule 2, u should have been unmarked in the first application of Rule 1 and Rule 2. This brings a contradiction.

Part 2. Consider a shortest path in $D' = D[V']$ connecting x to y in V'_u , if this path includes vertex u as an intermediate vertex: $(x, \dots, u_1, u, u_2, \dots, y)$, where u_1 (u_2) is a dominating (absorbant) neighbor of u , based on Rule 1, $(x, \dots, u_1, v, u_2, \dots, y)$ is also a shortest path in $D'_u = D[V'_u]$ and all its intermediate vertices are in V'_u .

Part 3. Consider a shortest path in D' connecting x to y in V'_u . Assume that this shortest path includes vertex u as an intermediate vertex: $(x, \dots, u_1, u, u_2, \dots, y)$. Based on Rule 2, u_1 has four choices: v, w, v_d (a dominating neighbor of v), and w_d (a dominating neighbor of w). Similarly, u_2 has four choices: v, w, v_a (an absorbant neighbor of v), and w_a (an absorbant neighbor of w). Also, v and w are bidirectionally connected. Hence, the shortest path can be replaced by one of the eight paths:

$$\begin{aligned} &(x, \dots, v_d, v, w, w_a, \dots, y), (x, \dots, v_d, v, v_a, \dots, y), \\ &(x, \dots, v, w, w_a, \dots, y), \quad (x, \dots, v_d, v, w, \dots, y). \end{aligned}$$

(The other four cases can be derived by interchanging the role of v and w .) Therefore, V'_u is strongly connected. However, the resultant path in D'_u may not be the shortest one in D' as in the case for $(s, \dots, v_d, v, w, w_a, \dots, d)$ with its hop count increased by one.

Part 4. The proof of the strongly connected property can be based on the proofs of Parts (2) and (3), where a path is constructed from x to y in D'_u . Since the selection of x and y is random, D'_u is still strongly connected.

Next, we show that all vertices *covered* (i.e., dominated or absorbed) by u are also covered by a vertex in V'_u . If u is removed by applying Rule 1, arbitrarily select $x \in N_d(u)$ ($x \in N_a(u)$), the condition $N_d(u) - \{v, w\} \subseteq N_d(v)$ ($N_a(u) - \{v\} \subseteq N_a(v)$) implies that $x \in N_d(v)$ ($w \in N_a(v)$) or x is v itself. If u is removed by applying Rule 2, arbitrarily select $x \in N_d(u)$ ($x \in N_a(u)$), the condition

$$N_d(u) - \{v, w\} \subseteq N_d(v) \cup N_d(w)$$

and

$$N_a(u) - \{v, w\} \subseteq N_a(v) \cup N_a(w)$$

implies that $x \in N_d(v)$ ($w \in N_a(v)$) or x is v or w . Therefore, x covered by u is also covered by v in V'_u (by Rule 1), or v or w in V'_u (by Rule 2). In addition, u itself is covered by a vertex in V'_u based on Corollary 4. \square

Note that the extended marking process can be applied iteratively to form a hierarchy of dominating and absorbant sets. In Fig. 4a, nodes 4, 7, 8, and 9 (double-cycled nodes) are level-1 gateway nodes and nodes 7 and 9 (triple-cycled nodes) are level-2 gateway nodes on top of level-1 gateway nodes. Other details of this hierarchical structure can be found in [33].

3.3 Implementation Issues

To provide a decentralized implementation, each host keeps a list of its neighbors and sends this list to all its neighbors. By doing so, each host has *2-hop neighborhood information*; that is, information about its neighbors and neighbors of all its neighbors. The extended marking process is then applied to each individual host to determine its status. The gateway and nongateway status (i.e., marked and unmarked status) of hosts are exchanged among the neighboring hosts. In an ad hoc wireless network, it would not be cost-effective for each host to keep distance- k neighborhood information, where k is a relatively large integer corresponding to a distance. However, 2-hop neighborhood information forces a *restricted implementation* of Rules 1 and 2. Specifically, it requires u and v to be neighbors (actually, they should be bidirectionally connected) in Rule 1 and v and w are neighbors of u in Rule 2. Since the number of neighbors of each host is upper bounded by Δ (the vertex degree of the network), in the restricted implementation, Rule 1 requires Δ (choosing one out of Δ) set coverage checks at each node and Rule 2 requires Δ^2 (choosing two out of Δ) set coverage checks at each node. Clearly, a *general implementation* of Rules 1 and 2 requires 3-hop neighborhood information. Since the number of one- and two-hop neighbors are upper bounded by Δ^2 , in the general implementation, Rule 1 requires Δ^2 set coverage checks at each node and Rule 2 requires Δ^4 (choosing two out of Δ^2) at each node.

The existence of unidirectional links brings a special challenge: If a link is directed from host u to host v , host v

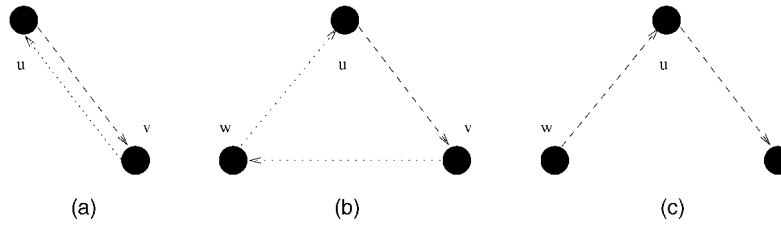


Fig. 8. (a) and (b) Two possible ways for u to recognize an absorbant neighbor v when $TTL=2$. (c) An unrecognizable triangle of three hosts.

can receive packets from host u , but host u is not aware of the existence of v . One possible solution is that each host emits a *beacon* packet at regular intervals to its neighbors. Once host u receives a beacon containing its *id* and *id* of v as its immediate forwarding host, host u knows host v as an absorbant neighbor. Note that the beacon is also used to transit neighborhood information of v to u . Certainly, pure broadcasting of beacon through flooding is not feasible because of its high consumption of bandwidth and energy. Each host has to selectively emit and forward a beacon. For example, if there is a bidirectional link uv , v only needs to send back an incoming beacon from u and there is no need for u to broadcast it to other neighbors. However, u does not know in advance the existence of link uv to avoid broadcasting. To solve this problem, we can add a Time-To-Live (TTL) field to limit the number of hops a beacon is allowed to travel. In the above case, TTL is set to 1 (also called 1-hop beacon) which is used to distribute 2-hop neighborhood information. In general, in order to recognize all unidirectional links, TTL should be set to the *maximum cycle number* of all links in D , where the cycle number of a link is defined as the size of the shortest cycle containing the link in D minus one. Clearly, in an undirected graph, the size of the shortest cycle for each link is 2 and, hence, the maximum cycle number is 1. In a directed graph, TTL is usually set to a small number, say 2 or 3, to save bandwidth. As a result, reachability information may not be able to be propagated back to some senders that have unidirectional outgoing links and the corresponding links are dropped. Therefore, an intelligent decision on TTL is needed to balance the bandwidth consumption and the drop rate of unidirectional links. Figs. 8a and 8b show two possible ways for host u to recognize an absorbant neighbor v when TTL is set to 2. Arrow dotted lines correspond to links used to pass a beacon.

The selection of $TTL=2$ (or 3) broadens the application scope of the restricted implementation (based on 2-hop neighborhood information) of Rule 1 and Rule 2. For example, it is not required that u and v in Rule 1 to be neighbors, as long as they stay within 2-hops apart and can receive the neighbor set of each other. Consider again Fig. 6a. All four vertices are marked using the extended marking process. Although u and v (similarly for w and x) are not directly connected, each can receive the beacon (containing neighbor set information) sent from the other when $TTL=2$, u (also w) can be unmarked by applying Rule 1. Fig. 6b has the same situation where u is 2-hops apart from v and w and, hence, u can be unmarked by applying Rule 2.

Based on the above analysis, a node can recognize and receive information from its dominating neighbors. The problem lies in its absorbant neighbors. For example, suppose

node u has only two connections with v and w : (w, u) and (u, v) (see Fig. 5), we consider the following four cases:

1. If there is a bidirectional link between v and w , then the beacon and neighborhood information from v can be passed on to u via w . Therefore, u can correctly assign its mark to F.
2. If there is a unidirectional link from v to w (see Fig. 5b), then the situation is similar to the first case and u can correctly assign its mark to T.
3. If there is a unidirectional link from w to v , u should be assigned F. Since the default value of $m(u)$ is assigned F, this result is correct by default even though u does not have connectivity information.
4. If there is no connection between v and w , u should be marked T (see Fig. 5c). However, u has no way of knowing the existence of node v . The default value of $m(u)$ is incorrect.

Clearly, Case 4 above causes problems. If v has other neighbors, the neighborhood information of v can be passed on to either u or w via a fourth node v_a . Note that path (v, v_a, w, u) requires TTL to be 3 while path (v, v_a, u) requires TTL to be only 2. If the beacon from v cannot reach u within TTL hops, link (u, v) is dropped.

4 DOMINATING-SET-BASED ROUTING

Assume that a connected dominated set has been determined for a given ad hoc wireless network. The routing process in dominating-set-based routing consists of three steps in Fig. 9. Potentially, many existing routing protocols: proactive, reactive, non-GPS-based, GPS-based, or their combinations can be used as underlying protocols for the induced graph. For example, destination sequenced distance vector (DSDV) [24], dynamic source routing (DSR) [5], and temporally ordered routing algorithm (TORA) [22] can be used.

We use the proactive routing approach, one that is similar to DSDV, to illustrate the dominating-set-based routing. Each gateway host keeps the following information items: *gateway domain membership list* and *gateway routing table*. The gateway domain membership list of a gateway host is a list of nongateway hosts that are adjacent to the gateway host. The gateway routing table at gateway host v includes one entry for each gateway host other than v together with its domain membership list. For example, given an ad hoc wireless network as shown in Fig. 4. Fig. 4b shows that host 8 has three members (3, 10, 11_a) in its gateway domain membership list. A neighbor with a subscript a (d) corresponds to an absorbant neighbor (dominating neighbor). Fig. 4c shows the gateway routing table at host 8. The first column of Fig. 4c is a set of entries

Dominating-set-based routing:

1. If the source is not a gateway host, it forwards routing packets to a *source gateway*, which is one of the adjacent absorbant gateway hosts.
2. This source gateway acts as a new source to route the packets in the *induced graph* generated from the connected dominating and absorbant set.
3. Eventually, the packets reach a *destination gateway*, which is either the destination host itself or a dominating gateway of the destination host. In the latter case, the destination gateway forwards the packet directly to the destination host.

Fig. 9. Dominating-set-based routing.

for each destination gateway (excluding host 8). The second column includes the membership list for each destination gateway. The third column shows next hop information of a shortest path (the length of a path is measured by hop count). The last column includes hop count for each destination gateway. Note that the way that routing tables are constructed and updated in the induced graph can be different from protocols to protocols. A gateway together with its nongateway neighbors form a cluster. A nongateway may have more than one gateway neighbor and, hence, it may belong to several clusters.

The major difference between dominating-set-based routing and cluster-based routing is the way the dominating set is constructed. In dominating-set-based routing dominating set is connected, whereas, in cluster-based routing, clusterheads form an unconnected dominating set. The cost for creating and maintaining a dominating set in both cases are similar and are done in a localized way, although the cluster-based approach may exhibit sequential propagation for certain cases. However, the property of connected dominating set gives the edge of dominating-set-based routing over cluster-based routing in both proactive routing and reactive routing. In proactive routing, routing tables in dominating-set-based routing can be simply built on connected dominating nodes directly. In cluster-based routing, since adjacent heads are separated by two or three hops, one or two gateway nodes are used to connect two adjacent heads. Unless each head is equipped with 3-hop neighborhood information in order to obtain information about all adjacent heads, routing tables have to also be built on gateway nodes. Similarly, in reactive routing, the search process is also more involved for cluster-based routing. Note that the cluster-based approach is not designed to minimize the number of gateways (also called repeaters) hence, to minimize the size of the connected dominating set (consisting of head nodes and repeaters). There are two extreme ways to select repeaters: The "normal" one includes all gateways that meet the condition (i.e., nodes with two or more neighbors in different clusters) and the "optimized" one uses a variation of Kruskal's algorithm (for constructing a minimum spanning tree) that sequentially merges two fragments (initially each cluster is a fragment). Note that the distributed Kruskal's algorithm [4] can be used that connects fragments simultaneously. However, coordination among hosts within a fragment is needed and this operation is expensive when a fragment becomes large.

Dominating-set-based routing does have a drawback in highly dense networks where communication complexity is high. The problem can be alleviated by applying the extended marking process on cluster-based routing or GPS-based (location-aware) routing [16] where the search space is partitioned into a 2D grid with one coordinator selected from each grid point. The size of each grid point is selected in such a way that coordinators form a connected graph. Like cluster-based routing, GPS-based routing also "trivializes" MCDS approximation. Each grid point has a constant number of neighbor grid points (eight as shown in Fig. 10a). In this case, the dominating and absorbant set can be constructed among coordinators (see Fig. 10a where black nodes are coordinators). If the size of each grid point is selected in such a way that each grid point can only communicate with its neighbors, any dominating and absorbant set generated from coordinators (one coordinator from each grid point) has a constant approximation ratio. Another way to remove nodes in a dense graph is by using the Yao graph [34]: For each node u , any k ($k \geq 6$) equal-separated rays originated at u define k cones. In each cone, choose the closest v (if any) within the transmitter range of u and add a directed link (u, v) . It is shown that the resultant graph is still a connected one if the original graph is connected. Fig. 10d shows a Yao graph where $k = 8$. A connected dense graph can also be made sparse by removing links only and the resultant graph is still connected. In the Gabriel graph [11], link uv is present if and only if the open disk (the shadowed area in Fig. 10b) with diameter uv contains no other nodes. In the RNG (relative neighborhood graph) [31], link uv exists if the distance between u and v is less than or equal to the distance between every other vertex w , and whichever of u and v is farther than w . In Fig. 10c, uv exists if there is no node in the shadowed area (the intersection area of two disks centered at u and v with uv as radius).

5 DOMINATING AND ABSORBANT SET UPDATE/RECALCULATION

A unit graph represents just a global snapshot of the dynamic topology for a given ad hoc wireless network. In an ad hoc wireless network, each host can move around without speed and distance limitation. Also, in order to reduce power consumption, mobile hosts may switch off at any time and switch on later. Because the topology of the network changes over time, the connected dominating set

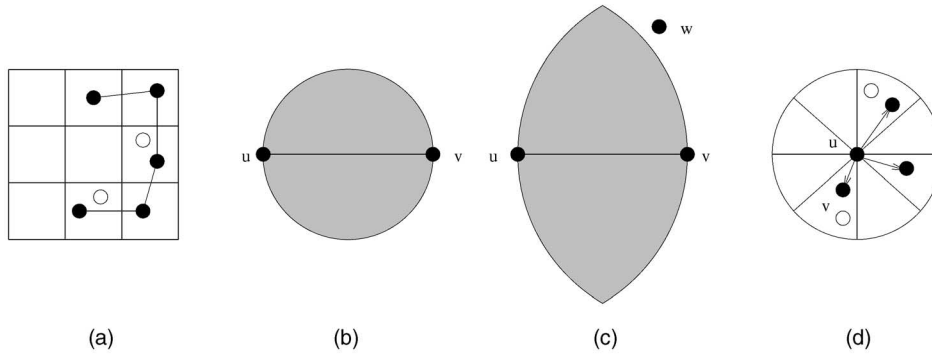


Fig. 10. (a) 2D grid. (b) Gabriel graph. (c) RNG graph. (d) Yao graph.

also needs to be updated from time to time. We can summarize topological changes of an ad hoc wireless network into three different types: mobile host switching on, mobile host switching off, and mobile host movement.

The challenge here is when and how each host 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 gateway/nongateway status. If many mobile hosts in the network move around, gateway recalculation might be a better approach; that is, the dominating and absorbant set is recalculated from scratch. On the other hand, if only few mobile hosts move around at each time interval, then gateway information can be updated locally. Fortunately, the extended marking process has the following desirable locality property:

Locality property of the extended marking process: *The status of a node (gateway/nongateway) depends only on connections of its neighbors, not the status of its neighbors.*

The implication of the locality property is that the status of a node is independent of the status of its neighbors. Therefore, when node u switches on/off, nodes and only nodes that are neighbors of u may change their status. Note that Rule 1 and Rule 2 do depend on node status. Theorem 5 shows that one application of Rule 1 and Rule 2 is sufficient to stabilize node status. In addition, such an application is limited to a 2-hop neighborhood if the restricted implementation of Rule 1 and Rule 2 is used; otherwise, it is limited to a 3-hop neighborhood in the general implementation. Therefore, the proposed scheme is effective in handling network changes in a network where a majority of changes are switching on/off operations. Such a network can be either a sensor network or a rooftop

network [26] which is not mobile, but is deployed very densely in metropolitan areas. In the following, we assume that the underlying graph of an ad hoc wireless network always remains strongly connected.

When a mobile host u switches on (see Fig. 12), only its nongateway neighbors, along with host u , need to update their status because any gateway neighbor will still remain as gateway after a new vertex u is added. For example, in Fig. 11a, when host u switches on, the status of gateway neighbor v is not affected because at least two of v 's neighbors w_1, w_2 , and w_3 are not connected originally and these connections will not be affected by host u 's switching on. On the other hand, host u 's switching on might lead a nongateway neighbor host v to mark itself as gateway, depending on the connection between host u and v 's neighbors. In Fig. 11b, nongateway neighbor v switches to gateway, while in Fig. 11c, v remains nongateway. The status of the switch-on node u depends on connections of its neighbors. For example, in Fig. 11a, u is marked F while in Fig. 11c it is marked T. The *ex-gateway* neighbors unmarked by Rule 1 or Rule 2 may be marked again by applying the extended marking process (and Rule 1 and Rule 2) depending on the new connections. Note that a new link can be either unidirectional or bidirectional. Whenever there is a newly marked gateway v (which is either the switch-on host u or its neighbor), if the restricted implementation of Rule 1 and Rule 2 is used, host v and all its gateway neighbors apply Rule 1 and Rule 2 once to further reduce the number of gateway hosts; otherwise, host v and all its 2-hop gateway neighbors apply Rule 1 and Rule 2 once. In [32], Wu and Dai showed that, when node degree (instead of node id) is used to prevent "simultaneous

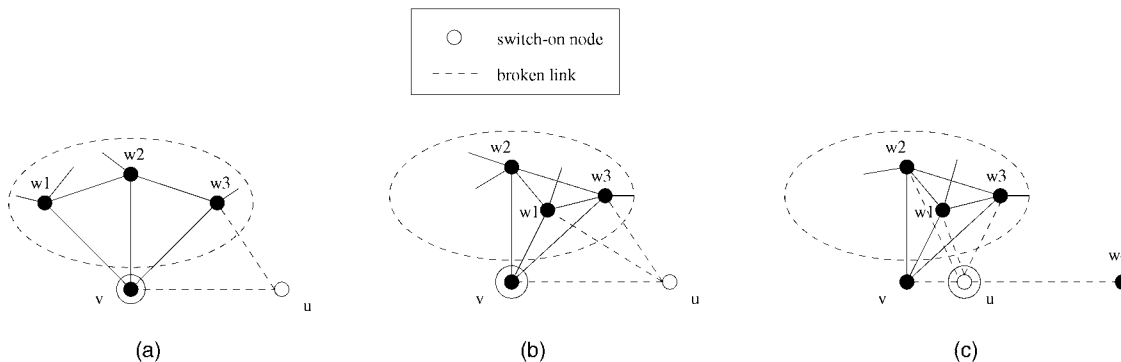


Fig. 11. Cases for mobile host u switching on.

Switching on:

1. Mobile host u broadcasts to its neighbors about its switching on. In particular, a k -hop beacon is used to broadcast this signal to its dominating and absorbant neighbors.
2. Each host $v \in u \cup N(u)$ exchanges its neighbor set $N(v)$ with its neighbors. Again, a k -hop beacon is needed for the exchange.
3. Host u assigns its marker $m(u)$ to T if there are $(w, u) \in A$ and $(u, v) \in A$, but $(w, v) \notin A$.
4. Each non-gateway neighbor $v \in N(u)$ assigns its marker $m(v)$ to T if $(w, v) \in A$ and $(v, u) \in A$, but $(w, u) \notin A$; or $(u, v) \in A$ and $(v, w) \in A$, but $(u, w) \notin A$.

Fig. 12. Status update for a switch-on node.

removal," hosts within 2 hops and 3 hops of switch-on (switch-off) hosts need to update their status in the restricted implementation and the generalized implementation, respectively. It is also shown that the rules based on node id are more stable with respect to switch-on/off operations than the ones based on node degree.

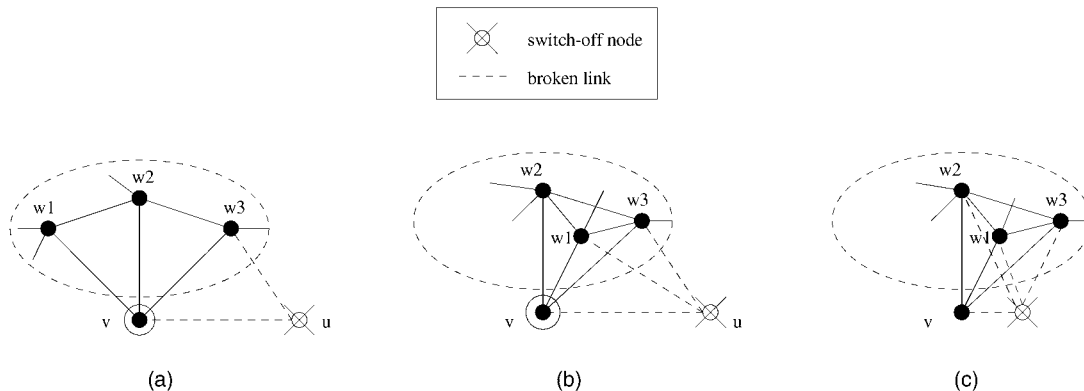
The case for a host switching off is similar to the one for a host switching on. When a mobile host v switches off (see Fig. 14), only gateway neighbors of the switch-off host need to update their status because any nongateway neighbor will still remain as nongateway after vertex v is deleted. For example, in Fig. 13a, when u switches off nongateway neighbor v is not affected. 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 u 's switching off. On the other hand, in Figs. 13b and 13c, host u 's switching off might lead a gateway neighbor u to a nongateway, depending on the connection between its neighbor hosts w_1, w_2 , and w_3 .

Note that, since the underlying graph D is connected, we can easily prove by contradiction that the resultant dominating set is still connected when a host (gateway or nongateway) switches off. Before node u switches off, it emits a gateway update signal to its neighbors. If node u is off by fault, it is assumed that its status will be detected by its neighbors. The case when the switch-off host is a gateway needs more discussion. When a gateway u switches off, the following three cases are considered:

- a. If the marked set is derived using the extended marking process only, then nothing else needs to be done.
- b. If the marked set is derived using the restricted implementation of Rule 1 and Rule 2, then the neighbors of u need to apply the extended marking process (and Rule 1 and Rule 2).
- c. If the marked set is derived using the general implementation of Rule 1 and Rule 2, then the neighbors of u and their neighbors need to apply the extended marking process (and Rule 1 and Rule 2).

The marked sets of Figs. 6a and 6b are derived using the general implementation of Rule 1 and Rule 2. In Fig. 6a, when host v switches off, ex-gateway w becomes uncovered. Therefore, each neighbor of v and its neighbors need to apply the extended marking process once followed by Rule 1 and Rule 2. In Fig. 6a host u is remarked T after v switches off. In Fig. 6b, when host w switches off, host y becomes isolated (although all hosts in the networks are covered). After the application of Rule 1 and Rule 2 to the neighbors of w and their neighbors, host u is marked T and the resultant marked set is connected.

We can also consider a link status change between unidirectional and bidirectional due to mobility: a) a unidirectional link changes to a bidirectional link and b) a bidirectional link changes to a unidirectional link. In either case, under the condition that the graph is strongly connected before and after each change, only one end node of the link might change its status. The status of all other nodes remains

Fig. 13. Cases for mobile host u switching off.

Switching off:

1. Mobile host u broadcasts to its neighbors about its switching off. In particular, a k -hop beacon is used to broadcast this signal to its dominating and absorbant neighbors.
2. Each host $v \in N(u)$ exchanges its neighbor set $N(v)$ with its neighbors. Again, a k -hop beacon is needed for the exchange.
3. Each gateway neighbor v assigns its marker $m(v)$ to F if all neighbors are pairwise connected; that is, if $(w, v) \in A$ and $(v, w') \in A$, then $(w, w') \in A$ for any two neighbors w and w' of v .

Fig. 14. Status update for a switch-off node.

unchanged. Specifically, when a bidirectional link uv is changed to (u, v) , the status of v might change from gateway to nongateway (see Fig. 15a), and when a unidirectional link (u, v) is changed to uv , the status of v might change from nongateway to gateway (see Fig. 15b).

A mobile host u 's movement can be viewed as several simultaneous or nonsimultaneous link connections and disconnections. For example, when a mobile host moves, it may lead to several link disconnections with its neighbor hosts and, at the same time, it may have new link connections to the hosts within its wireless transmitter range, these new links may be disconnected again depending on the way host u moves. In gateway updates that are caused by the movement of a mobile host, just before mobile host u starts to move, it sends out a special signal $\{id(u), start\}$, then during its movement host u continuously sends out signal $\{id(u), heart_beat\}$ at every interval of τ time units, and, when it stops moving around, host u sends out signal $\{id(u), stop\}$. The details of this approach are similar to the ones for undirected graphs [33].

6 SIMULATION

We have conducted a simulation study to measure the size of the dominating and absorbant set generated from the extended marking process. Five sets of simulation are conducted.

1. The effectiveness of Rules 1 and 2 is evaluated using node id to break a tie. Three sets of data are used corresponding to three different percentages of bidirectional links (denoted as pu): 100 percent (where all links are bidirectional), 95 percent and 90 percent.

2. Comparisons among the extended marking process with Rules 1 and 2 (based on node degree), Guha and Khuller's MCDS, and Lin and Gerla's basic clustering approach are given both in terms of the size of dominating and absorbant set and the average number of rounds.
3. The effect of host switching on/off is evaluated. There are two types of switching off: gateway and nongateway. Three ratios are used: number of changed hosts versus
 - a. the total number of gateways in the network,
 - b. the number of neighbors, and
 - c. the number of gateway neighbors.

The percentage of bidirectional links is 90 percent and $r = 50$. The marked sets derived by node degree and node degree are compared.

4. The effect of host movement is measured in terms of the number of nodes that change their status: either from gateway to nongateway or from nongateway to gateway. The percentage of bidirectional links is 90 percent and the marked set is derived by node id.
5. The drop rates of unidirectional links are collected for 2-hop and 3-hop beacons. The percentages of bidirectional links are 95 percent, 90 percent, and 80 percent.

The simulation was performed using the following parameters: ν represents the number of mobile hosts in the network, γ the number of gateways (the size of the dominating and absorbant set), r the radius of mobile host's transmitter range, pu the percentage of undirected edges (bidirectional links), pm the percentage of hosts move around within each time interval, dr_2 and dr_3 the drop rates of unidirectional links when 2-hop and 3-hop beacons are used, respectively. Random unit graphs are generated in a

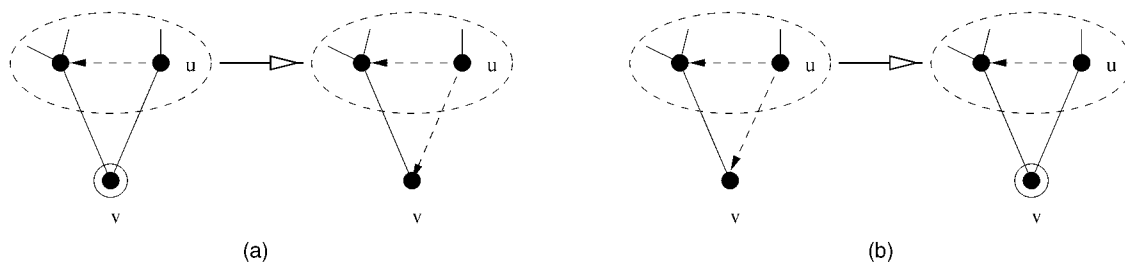


Fig. 15. Node status change due to the change of link status: (a) from uv to (u, v) and (b) from (u, v) to uv .

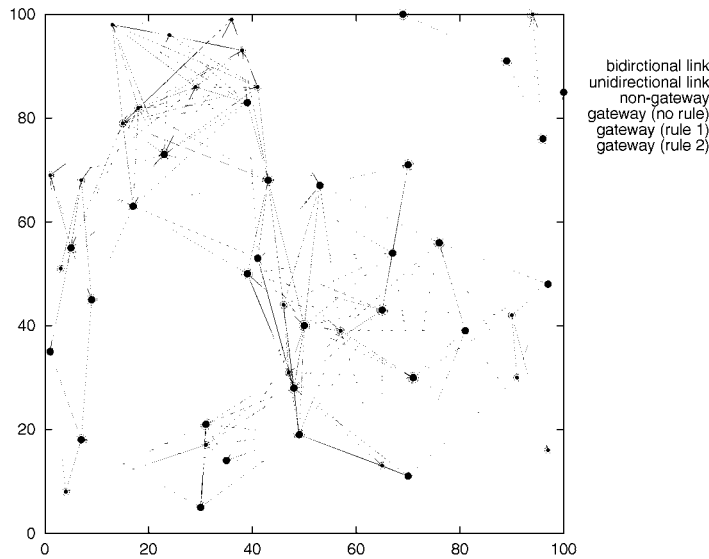


Fig. 16. A sample network generated from the simulation package.

100 × 100 square units of a 2D plane, by randomly throwing a certain number of mobile hosts. A 2D plane resembles more an actual ad hoc wireless network where mobile hosts usually stay on ground. Each mobile host has the same transmitter radius r . If the distance between any two hosts is less than radius r , then there is a link connection between these two hosts. A random number, between 0 and 100 inclusive, is associated with each link. If the random number generated from [0..1] is less than or equal to pu , then the corresponding link is undirected; otherwise, it is directed. In each simulation, the radius of the mobile host's transmitter radius r is set to two different values: 25 and 50. In this way, we can control the density of random graphs, since the density of random graphs increases as r increases. Each random graph must be a strongly connected graph; otherwise, it is discarded. For each combination of r and pu , we also vary the number of mobile hosts ν from 0 to 100. For each ν , the random graph is generated and simulated until a predefined confidence interval for the population mean is reached and, then, simulation results are measured by simply taking the sample mean (i.e., the average of all cases). Specifically, the 90 percent confidence interval of dominating set is within ± 1 percent and the 90 percent confidence interval of changed hosts caused by host switch-on/off or movement is within ± 10 percent. The general implementation of Rule 1 and Rule 2 is used, although the restricted implementation of Rule 1 and Rule 2 has the similar results [32]. Fig. 16 shows a sample network generated from the simulation package where nongateway and gateway nodes after applying the extended marking process, Rule 1, and Rule 2 (both based on node id) are shown. In this sample network, the average node degree is set to 10 with $\nu = 50$ and $pu = 95\%$.

Fig. 17 shows the number of gateways versus the number of hosts in the network for the increasing number of hosts. We can see that, without applying two rules, the performance of the extended marking process (the curve for "no rule") fares poorly. This is because the graph under simulation is not sufficiently dense and, hence, the chance that any two pairs of neighbors of a host are connected is low. However, by applying two rules, the performance of

our approach (the curve for "rules 1 and 2") improves dramatically. Rules 1 and 2 are more effective when the graph is dense (i.e., each host has a long transmitter range). Rule 2 covers Rule 1 for most of the cases. Rule 1 can be considered as a special case of Rule 2 where host w is used without contributing in host coverage. However, the following situation exists when Rule 2 cannot cover Rule 1: The neighbor set of u is covered by v and $id(u) < id(v)$, but $id(w) < id(u)$ for any neighbor w of v (i.e., $id(w)$ is no longer the minimum of $id(u)$, $id(v)$, and $id(w)$). In Fig. 16, only one gateway host that is unmarked by Rule 1 that cannot be unmarked by Rule 2. Because Rule 1 is easy to implement, it is still useful and can be applied first (to unmark some hosts) before Rule 2 is used. Also, as the percentage of bidirectional links drops, the average number of gateway hosts increases, especially for the extended marking process with two rules.

Figs. 18a and 18b show comparisons of various methods in terms of average size of dominating (and absorbant) set generated, including the proposed extended marking process with two rules, using node degree to break a tie. Guha and Khuller's approximation approach ("MCDS"), and Lin and Gerla's clustering approach with node degree are used in the selection process. In this simulation, the "cluster (normal)" one includes all gateways that meet the connection requirement and the "cluster (optimized)" one uses a variation of Kruskal's algorithm that sequentially merges two fragments (initially each cluster is a fragment). All methods are based on node degree. Clearly, "rules 1 and 2," "MCDS," "cluster (optimized)" stay close, especially when $r = 50$. "MCDS" is the best and is close to the "optimal" case in both dense and sparse graphs. "Rules 1 and 2" beats "cluster (optimized)" when $r = 50$ (dense graphs) and the situation reverses when $r = 25$ (sparse graphs). In all cases, "cluster (normal)" is the worst. The small number of gateway hosts generated from "MCDS" or "cluster (optimized)" comes with a cost as shown in Fig. 18c, where the average numbers of rounds needed to generate a dominating (and absorbant) set under various methods are shown. "MCDS" and "cluster (optimized)" both require nonconstant number of rounds. This is expected, since both of them require some sort of global operations that

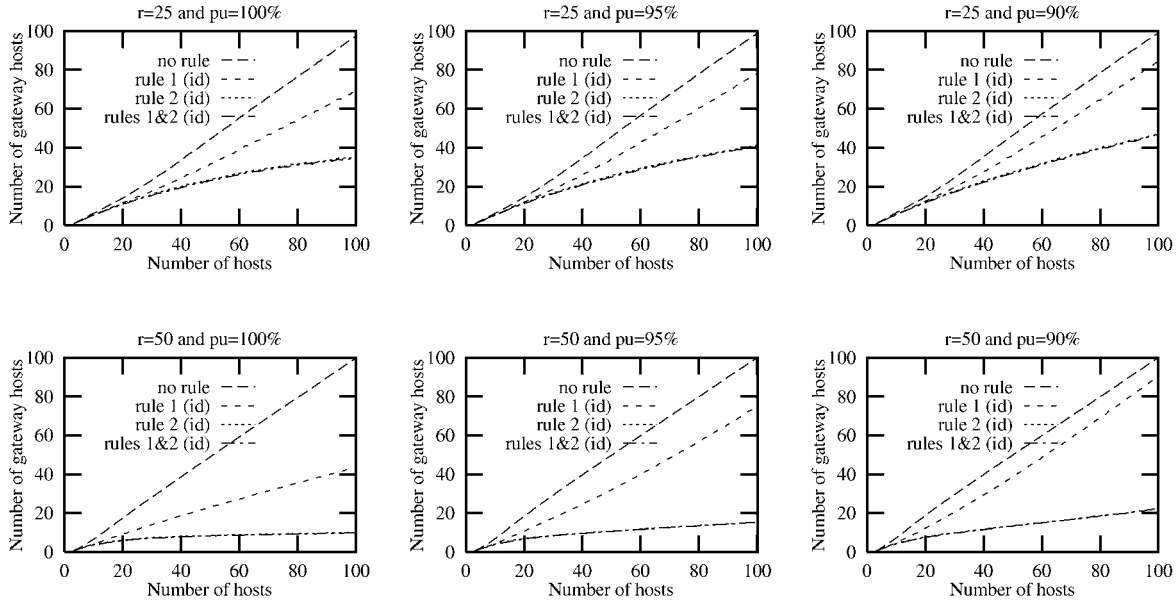


Fig. 17. Average number of gateway hosts relative to the number of hosts ν .

have to be serialized. “Rules 1 & 2” needs a constant number of rounds. “cluster (normal)” also generates a small number of rounds in average and the number of rounds slightly increases as the number of hosts increases.

Fig. 19 shows the average number of changed hosts by a switch-on/off operation with $r = 50$ and $pu = 90\%$. A changed host is the one that changes its status: either from gateway to nongateway or from nongateway to gateway. In general, a switch-on operation (the first column of Fig. 19) has more effect compared with a switch-off operation (the second and third columns of Fig. 19). There is little difference between a switch off operation that is gateway and a switch-off operation that is nongateway. Also, a switch-on/off operation affects status of more neighbors if the marked set is derived using node degree (the first line of Fig. 19), rather than node id (the second line of Fig. 19), to break a tie. This is expected, since a marked node based on Rules 1 and 2 (degree) tends to have more neighbors than the one based on Rules 1 and 2 (id).

Fig. 20 shows the average percentage of changed hosts during host movement with $pu = 90\%$ and the marked set being selected based on node id. Host movement is discretized into time intervals (each interval has τ time units). Assume that the percentage of hosts that move

around within each time interval is given. The distance of each movement is a random value in $[0..l]$, where l is a function of transmitter range r . In addition, the direction of movement is also random. In our simulation, four values are selected for l : $(1/4)r$, $(1/2)r$, r , and $2r$. Two transmitter ranges $r = 25$ and $r = 50$ are considered. Three different percentages of hosts that move around within each time interval (pm) are considered: 5 percent, 10 percent, and 25 percent. The percentage of changed hosts increases when either pm or l increases. However, the percentage of changed hosts decreases significantly as the density of the graph increases (the second row of Fig. 20, where $r = 50$).

Fig. 21 shows the average drop rates of unidirectional links. The results show that the drop rates are small in all the cases, especially for 3-hop beacons. 2-hop and 3-hop beacons are sufficient to detect unidirectional links as long as pu (the percentage of undirected links) stays relatively high. Note that all other simulations are conducted without considering the drop rate which has no significant impact as results in Fig. 21 show.

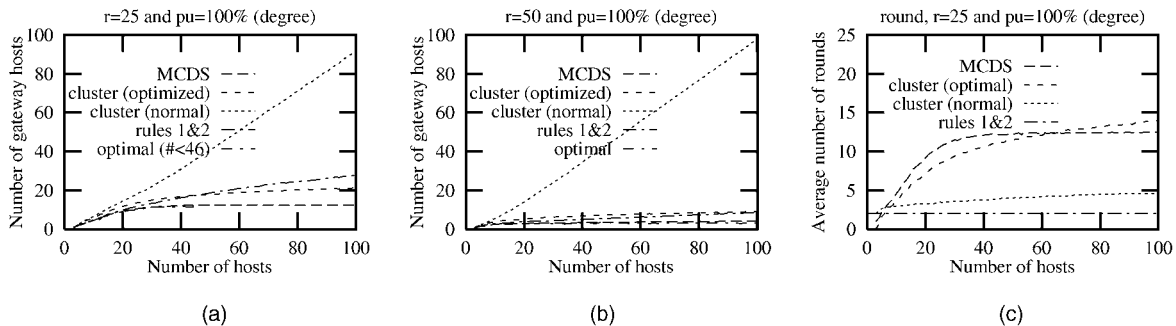


Fig. 18. (a) and (b): Average numbers of gateway hosts generated from different methods. (c) Average numbers of rounds needed for different methods.

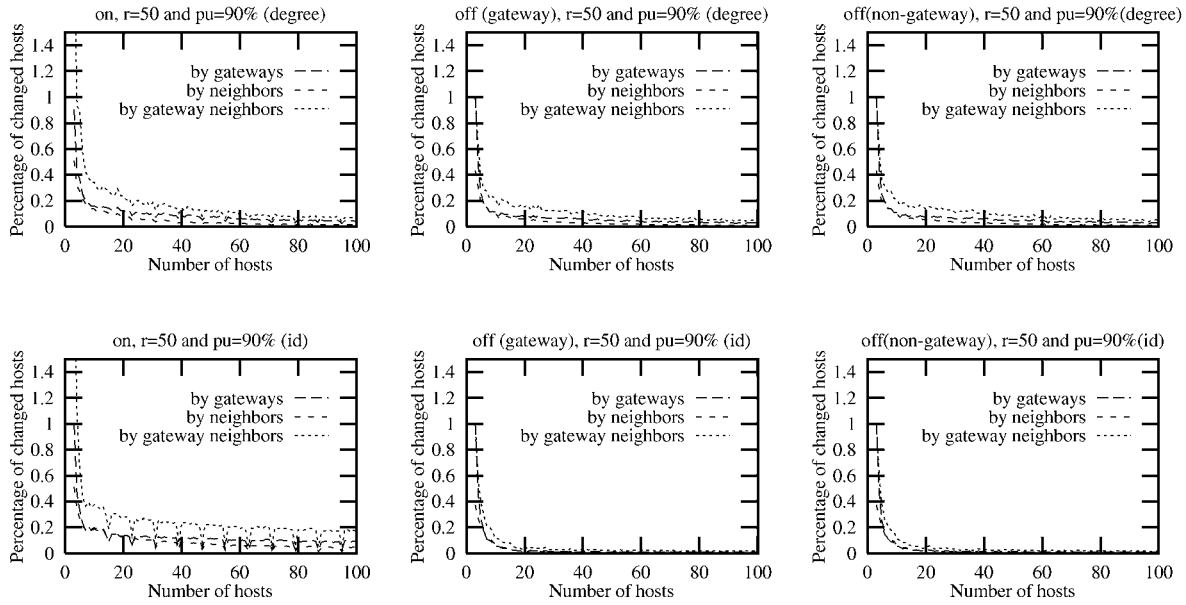


Fig. 19. Effect of an update operation: switching on and switching off.

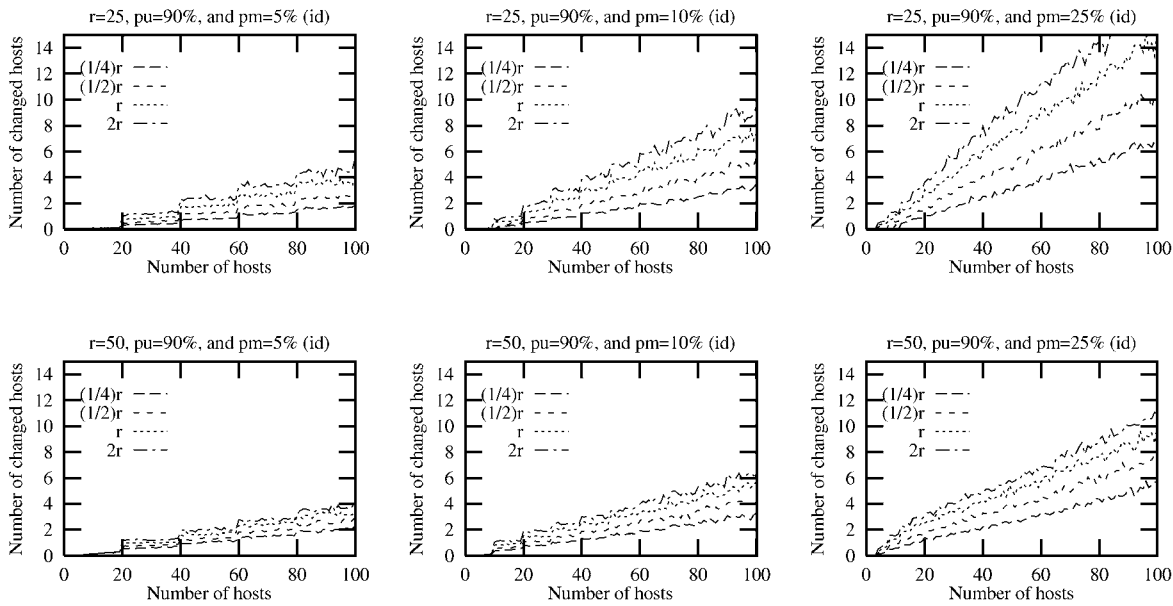


Fig. 20. Effect of host movement.

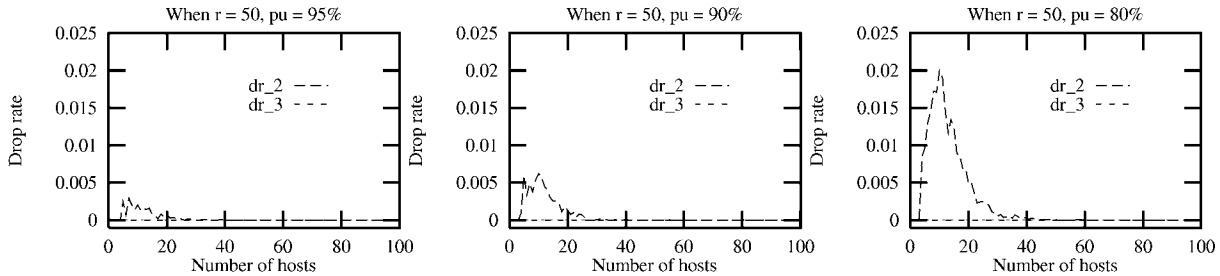


Fig. 21. Average drop rates of unidirectional links.

7 CONCLUSIONS

In this paper, we have extended dominating-set-based routing in ad hoc wireless networks with unidirectional links. This approach is based on finding a core which is a

dominating and absorbant set in a directed graph representing the topology of the ad hoc wireless network. An extended marking process has been presented that can identify such a core quickly and can update it easily in a dynamic environment. Although it is unlikely that the

dominating-set-based routing can solve all the critical routing issues in the ad hoc wireless network, we believe that it offers a very promising and unique combination of several existing approaches in conjunction with the novel use of the dominating and absorbant set. Our future work will focus on the design details of dominating-set-based routing and to verify the effectiveness of the design through an extensive simulation study. Another possible extension is to study the notion of k -dominating set. In this model, a node in the graph is either in the k -dominating set or within k hops from a node in the k -dominating set [2].

ACKNOWLEDGMENTS

The author would like to thank Fei Dai and Hailan Li who have participated at the various stages of this project. This work was supported in part by US National Science Foundation grants CCR 9900646 and ANI 0073736.

REFERENCES

- [1] K.M. Alzoubi, P.J. Wan, and O. Frieder, "New Distributed Algorithm for Connected Dominating Set in Wireless Ad Hoc Networks." *Proc. 35th Hawaii Int'l Conf. System Sciences*, pp. 1-7, Jan. 2002.
- [2] A.D. Amis, R. Prakash, T.H.P. Vuong, and D.T. Huynh, "Max-Min D-Cluster Formation in Wireless Ad Hoc Networks," *Proc. INFOCOM*, pp. 32-41, 2001.
- [3] D.J. Baker and A. Ephremides, "The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm," *IEEE Trans. Comm.*, vol. 29, no. 11, pp. 1694-1701, Nov. 1981.
- [4] D. Bertsekas and R. Gallager, *Data Networks*, second ed. Prentice Hall, 1992.
- [5] J. Broch, D. Johnson, and D. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," *IETF, Internet Draft*, draft-ietf-manet-dsr-00.txt, Mar. 1998.
- [6] A.T. Campbell, J. Gomez, and A.G. Valko, "An Overview of Cellular IP," *Proc. Wireless Comm. and Networking Conf.*, vol. 2, pp. 606-610, 1999.
- [7] B. Das, E. Sivakumar, and V. Bhargavan, "Routing in Ad Hoc Networks Using a Virtual Backbone," *Proc. Sixth Int'l Conf. Computer Comm. and Networks (IC3N '97)*, pp. 1-20, Sept. 1997.
- [8] S. Datta, I. Stojmenovic, and J. Wu, "Internal Node and Shortcut Based Routing with Guaranteed Delivery in Wireless Networks," *Cluster Computing*, Special issue on, Mobile Ad Hoc Networks, vol. 5, no. 2, pp. 169-178, Apr. 2002.
- [9] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," *Proc. ACM Int'l Conf. Mobile Computing and Networking (MOBI-COM '99)*, pp. 263-270, 1999.
- [10] R. Jacquet et al. "Optimized Link State Routing Protocol (OLSR)," *IETF Internet Draft*, org/internet-drafts/draft-ietf-olsr-01.txt, 2000.
- [11] K.R. Gabriel and R.R. Sokal, "A New Statistical Approach to Geographic Variation Analysis," *Systematic Zoology*, vol. 18, pp. 259-278, 1969.
- [12] S. Guha and S. Khuller, "Approximation Algorithms for Connected Dominating Sets," *Algorithmica*, vol. 20, no. 4, pp. 374-387, Apr. 1998.
- [13] Z.J. Haas and M.R. Peralman, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks," *IETF Internet Draft*, <http://www.ietf.org/internet-drafts/draft-ietf-zone-zrp-00.txt>, 1997.
- [14] C. Hedrick, "Routing Information Protocol," *Internet Request for Comments (RFC) 1058*, June 1988.
- [15] Y.D. Liang, "Dominations in Trapezoid Graphs," *Information Processing Letters*, vol. 52, pp. 309-315, 1994.
- [16] W.K. Liao, Y.C. Tseng, and J.P. Sheu, "GRID: A Fully Location-Aware Routing Protocol for Mobile Ad Hoc Networks," *Telecommunication Systems*, A Special Issue on Wireless Networks, vol. 18, no. 1-3, pp. 37-60, 2001.
- [17] C.R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," *IEEE J. Selected Areas in Comm.*, vol. 15, no. 7, pp. 1265-1275, 1996.
- [18] J.M. McQuillan, I. Richer, and E.C. Rosen, "The New Routing Algorithm for ARPANET," *IEEE Trans. Comm.*, vol. 28, no. 5 pp. 711-719, 1980.
- [19] J.M. McQuillan and D.C. Walden, "The ARPA Network Design Decisions," *Computer Networks*, vol. 1, no. 5, pp. 243-289, Aug. 1977.
- [20] J. Moy, "OSPF," version 2, *Internet Request For Comments RFC 1247*, July 1991.
- [21] C.H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Prentice Hall, 1988.
- [22] V.D. Park and M.S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," *Proc. IEEE Int'l Conf. Mobile Computing (INFOCOM '97)*, pp. 1405-1413, 1997.
- [23] C.E. Perkins, *Mobile IP: Design Principles and Practices*. Addison Wesley, 1997.
- [24] C.E. Perkins and E.M. Royer, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *Proc. ACM Special Interest Group on Comm. (SIGCOMM '94)*, pp. 234-244, 1994.
- [25] E.M. Royer and C.K. Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks," *IEEE Personal Comm.*, pp. 46-55, Apr. 1999.
- [26] T. Shepard, "A Channel Access Scheme for Large Dense Packet Radio Networks," *Proc. ACM Special Interest Group on Comm. (SIGCOMM) Conf. Comm., Architectures, Protocols, and Applications*, 1996.
- [27] R. Sivakumar, B. Das, and V. Bhargavan, "An Improved Spine-Based Infrastructure for Routing in Ad Hoc Networks," *Proc. Int'l Symp. Computers and Comm. (ISCC '98)*, 1998.
- [28] M. Steenstrup, *Cluster-Based Networks in Ad Hoc Networks*, C.E. Perkins, ed. Addison Wesley, 2001.
- [29] M. Steenstrup, *Routing in Comm. Networks*. Prentice Hall, 1995.
- [30] A. Tanenbaum, *Computer Networks*. Prentice Hall, 1996.
- [31] G. Toussaint, "The Relative Neighbourhood Graph of a Finite Planar Set," *Pattern Recognition*, pp. 261-268, vol. 12, no. 4, 1980.
- [32] J. Wu and F. Dai, "On Locality of Dominating Set in Ad Hoc Networks with Switch-On/Off Operations," *Proc. Int'l Symp. Parallel Architectures, Algorithms and Networks (I-SPAN '02)*, pp. 85-90, 2002.
- [33] J. Wu and H. Li, "A Dominating-Set-Based Routing Scheme in Ad Hoc Wireless Networks," *Telecomm. Systems*, A special issue on Wireless Networks, vol. 18, no. 1-3, pp. 13-36, 2001.
- [34] A.C.C. Yao, "On Constructing Minimum Spanning Trees in k -Dimensional Spaces and Related Problems," *SIAM J. Computing*, vol. 11, pp. 721-736, 1982.



Jie Wu received the BS degree in computer engineering and the MS degree in computer science both from Shanghai University of Science and Technology (now Shanghai University) in 1982 and 1985, respectively, and the PhD degree in computer engineering from Florida Atlantic University, Boca Raton, in 1989. He is currently a professor in the Department of Computer Science and Engineering, Florida Atlantic University. He has published

over 150 papers in various journals and conference proceedings. His research interests are in the area of mobile computing, routing protocols, fault-tolerant computing, and interconnection networks. Dr. Wu has served on many conference committees and editorial boards. He was a coquest editor of *IEEE Transactions on Parallel and Distributed Systems* and *Journal of Parallel and Distributing Computing*. He is the author of the text *Distributed System Design* published by the CRC Press. Dr. Wu was the recipient of the 1996-1997 and 2001-2002 Researcher of the Year Award at Florida Atlantic University. He has served as an IEEE Computer Society Distinguished Visitor. Dr. Wu is a member of the ACM and a senior member of the IEEE.

► For more information on this or any computing topic, please visit our Digital Library at <http://computer.org/publications/dlib>.