GATEWAY MULTIPOINT RELAYS – AN MPR-BASED BROADCAST ALGORITHM FOR AD HOC NETWORKS

Ou Liang, Y. Ahmet Şekercioğlu, Nallasamy Mani

Centre for Telecommunication and Information Engineering
Monash University, Melbourne, VIC 3168, Australia
{Ou.Liang, Ahmet.Sekercioglu, N.Mani}@eng.monash.edu.au

ABSTRACT

Broadcast is an essential part of ad hoc network routing protocols. An efficient broadcast algorithm can greatly reduce the number of retransmissions in a network, thus decreasing the number of packet collisions and overall power consumption. In this paper, we propose the Gateway Multipoint Relays (GMPR) broadcast algorithm, which combines the Multipoint Relay (MPR) and the maximal independent set (MIS) concepts to calculate a connected dominating set (CDS) in the network. Our algorithm, which is fully localized and distributed, can significantly reduce the redundant broadcasting in the network while keeping the cost of computation low. The GMPR constructs a CDS in two phases. In the first phase, an MIS is established where nodes in the MIS are referred to as dominators, and they form the gateways in the network. In the second phase, each gateway generates some connectors to connect other gateways based on the original MPR algorithm. Then, a self-pruning procedure is applied to each gateway to eliminate redundant gateways after the CDS construction. In this paper, we show that our algorithm has $O(\Delta^2)$ time complexity and $O(n)$ message complexity, where Δ is the maximum node degree and n is the total number of nodes in the network. Simulation is conducted to compare our new algorithm with two leading MPR-based CDS broadcast algorithms. The results show that our algorithm performs better than those algorithms in terms of the size of the generated CDS.

1. INTRODUCTION

Mobile Ad Hoc networks (MANETs) [1] have gained much attention in recent years due to their self-organizing and infrastructureless characteristics. Each node in a MANET can act as a router to receive and forward packets, allowing seamless communications between devices. Hence, MANETs have great application potential in various scenarios such as battle field communications, emergency services, disaster recovery, environmental monitoring, personal entertainment and mobile conferencing [2].

Broadcast is an important data transmission method used in MANETs to disseminate data and control messages. The goal of a broadcast algorithm is to maximize the node reachability in the network while keeping the computation and communication overheads to a minimum. However, the wireless nature of MANETs make it difficult to design an efficient broadcast algorithm. Every message sent by a node can be heard by all its adjacent nodes, and thus, only a subset of nodes in the network is needed to relay a broadcast message. Furthermore, in mobile ad hoc networks, nodes can randomly move around, leave the network or switch off, and new nodes may join the network unexpectedly. These characteristics cause the network topology to change frequently. Therefore, a broadcast algorithm that requires the global information of a network may be unstable and complex.

One approach for doing efficient broadcast is to choose a small subset of all available nodes, called *connected dominating set* (CDS) in the network based on local information such as the local topology. A *dominating set* (DS) is a subset of nodes in the network where every node in the network is either a member of the subset, or a neighbor of at least one member of the subset. A DS is called a CDS if all nodes in the DS are connected. Only nodes in a CDS can retransmit packets while other nodes can only receive them. In this kind of localized approach, a node determines the forwarding state of itself and/or its neighbors only based on its neighborhood information. A CDS constructed in such a localized manner can adapt to frequent topology changes, thus ensuring the property of stability.

The original Multipoint Relays (MPR) concept proposed in [3] is a distributed localized broadcast algorithm, which is computationally lightweight. In this algorithm, each node collects the two-hop neighborhood information (such as node ID) from its one-hop neighbors and determines the forwarding state of its one-hop neighbors based on this information. Specifically, a node knows its one-hop neighbors and neighbors of these one-hop neighbors after it collects the two-hop neighborhood information. Then it selects a subset of nodes from its one-hop neighbors as the forwarding nodes to cover all its two-hop neighbors. Nodes that have been chosen as forwarding nodes are called MPRs.

The original MPR algorithm is *source dependent*, that is, an MPR will only forward a packet that comes from its selectors (nodes that select it as an MPR). A selector and its MPRs form a local CDS to cover nodes within two hops away from the selector, and eventually, all nodes in the network are covered by a number of local CDSs. This rebroadcast process requires the last hop knowledge to indicate the source of a packet, which increases the complexity of the algorithm. To solve this problem, two MPR-based CDS algorithms have been proposed in sequence. In [4], Adjih, Jacquet, and Viennot presented a *source-independent* MPR, which generates a CDS in the network based on the original MPR algorithm. Later, Wu [5] further enhanced the *source-independent* MPR to reduce the size of the CDS. In this paper, we propose a new MPR-based broadcast algorithm which can effectively generate a CDS in a given network. We will show in Section 5 that, compared with Adjih's and Wu's algorithms, our new algorithm produces a smaller size CDS in the network.

Table 1. NOTATION

n	Total number of nodes in the network.
V	The set that contains all nodes in the network.
E	The set that contains all edges in the network.
$N^1(u)$	The one-hop neighbor set of node u, which contains all nodes that are one hop away from u.
$N^2(u)$	The two-hop neighbor set of node u, which contains all nodes that are two hops away from u.
$\mathrm{MPR}(u)$	The MPR set of node u, which contains all MPRs selected by node u.
D_u	The node degree of node u, which is the cardinality of $N^1(u)$.
Δ	The maximum node degree in a given network.

The rest of the paper is organized as follows. In Section 2, we provide preliminaries and review the original MPR algorithm. Section 3 presents the details of our MPR-based CDS broadcast algorithm. In Section 4, we prove the correctness and provide the time and message complexity of our algorithm. Section 5 presents the simulation results. Finally, we offer our concluding remarks in Section 6.

2. PRELIMINARIES

In our algorithm, we assume that each node has an unique ID (such as an IP address) and it knows IDs of all its one-hop and two-hop neighbors, which can be obtained from the "HELLO" messages transmitted periodically by each node in the network. We further assume that all nodes are fixed and have the same transmission range. The maintenance of a CDS in a mobile environment is left for future study.

Here we provide some definitions for later use in this paper. The notation used is presented in Table 1.

2.1. DS (Dominating Set) and CDS (Connected Dominating Set)

For a given connected graph $G = (V, E)$, a dominating set (DS) is a subset V' of V, where for each node u of V, u is either in V' or at least one neighbor of u is in V'. A DS is called a CDS if the subgraph induced by nodes in the DS is connected. A node in a CDS retransmits every packet it receives for the first time.

2.2. IS (Independent Set) and MIS (Maximum Independent Set)

For a given connected graph $G = (V, E)$, an independent set (IS) is a subset V' of V such that for any pair of nodes in V', there is no connection between them. A maximum independent set (MIS) is the largest IS for a given graph G, where no other nodes can be added into it.

2.3. MPR (Multipoint Relay)

The goal of the Multipoint Relay (MPR) is to reduce the flooding of broadcast packets in a network by locally minimizing retransmissions. In the MPR algorithm, each node periodically sends out a "HELLO" message to all its one-hop neighbors, which contains its one-hop neighborhood information. This message will not be retransmitted by other nodes. A node has the knowledge of its one-hop and two-hop neighbors after it receives these "HELLO" messages.

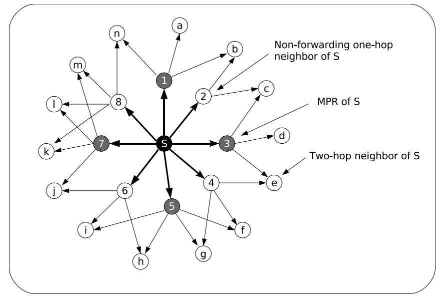


Fig. 1. An example of working of the original MPR algorithm.

Based on this neighborhood information, each node u selects some forwarding nodes from its one-hop neighbor set $N^1(u)$ to retransmit broadcast packets, so that all nodes in u's two-hop neighbor set $N^2(u)$ can receive these packets. The selected forwarding nodes are called Multipoint Relays (MPRs), and u is called the selector of these MPRs. An MPR will retransmit a message only once if the message is from one of its selectors and it is received for the first time. Node u and its MPRs form a local CDS that cover all nodes up to two hops away from u.

A simple greedy algorithm is used in [3] to calculate the MPR set for a given node u. Assume that a node v is a one-hop neighbor of u. We use the phrase "*u covers v*" if $v \in N^1(u)$. Let $P_v = |\{w \in N^1(v) | v \in N^1(u) \text{ and } w \in N^2(u)\}|$ denote the *preference* of node v, which is the number of two-hop neighbors of u that node v can cover. The greedy algorithm operates as follows:

Greedy Algorithm:

- Add nodes in $N^1(u)$ to $\mathrm{MPR}(u)$ if they are the only neighbors of some nodes in $N^2(u)$. Then update u's neighborhood information by removing MPRs and the two-hop neighbors they covered from $N^1(u)$ and $N^2(u)$ respectively.
- If there are still some uncovered two-hop neighbors, add a one-hop neighbor to $\mathrm{MPR}(u)$ if it covers the largest number of uncovered two-hop neighbors. If there is a tie, choose a node with larger P as the MPR. In case of another tie, the node with a smaller node ID is selected.

Figure 1 shows an example of the operation of the original MPR algorithm. Initially, nodes 1 and 3 are selected as MPRs since they are the only neighbors of the two-hop neighbor nodes a and d of node s. After node s updates its neighborhood information, both nodes 5 and 7, which cover two-hop neighbor set $\{f, g, h, i\}$ and $\{j, k, i, m\}$ respectively, have the largest coverage of uncovered two-hop neighbors. Because nodes 5 and 7 both have a P of 4, node 5 is chosen as the MPR based on the tie-break procedure. Among the remaining one-hop neighbors, node 7 covers the largest number of uncovered two-hop nodes, and thus it is selected as the MPR. Therefore, the MPR set of the node s is $\mathrm{MPR}(s) = \{1, 3, 5, 7\}$.

3. GMPR ALGORITHM DESCRIPTION

The process of finding a CDS with our algorithm can be described in two phases. In the first phase, an MIS is constructed in the network where nodes in the MIS are called *dominators*, and nodes covered by the dominators are referred to as *dominatees*. These dominators form the DS (dominating set) and they operate as the gateways in the network. In the second phase, each gateway (dominator node) calculates an MPR set to cover all its two-hop neighbors. However, not all MPRs need to forward packets in the network. An MPR determines its forwarding state based on the node degree of its MPR selectors. An MPR is actually a *forwarding* MPR if it is selected by a dominator whose node degree is the largest among all this MPR's one-hop neighbors. The forwarding MPRs are also in the DS and they perform as connectors of the gateways. We will prove in the next section that the gateways and the connectors generated in our algorithm can form a CDS in the network. After the CDS construction, a self-pruning procedure is applied to each gateway to remove the redundancy in the CDS.

Similar to the original MPR, our new algorithm also requires "HELLO" messages to be exchanged periodically in the network. The only extra information included in a "HELLO" message for our algorithm is the dominating state of the node that sends the "HELLO" message. A node can be in one of the four dominating states: dominator, dominatee, connector and white_node. These states can be easily represented using only two bits in a "HELLO" message. Hence, this extra cost is really marginal and can be ignored. The remains of this section describes our algorithm in detail.

3.1. Generating gateways

This phase generates an MIS in the network where nodes in the MIS form the gateways. Each node in the network initializes its dominating state as the white_node, and then they change to either the dominator state or the dominatee state subsequently. The connector state can only be entered from the dominatee state. Nodes that change to the dominator state broadcast a "HELLO" message immediately to inform its one-hop neighbors. Figure 2 illustrates the state transition process of our algorithm. Gateways are constructed based on the following steps:

- A white_node u announces itself as a dominator if it has the largest node degree D_u among it's white_node neighbors (neighbors in the white_node state) or it has no white_node neighbors and dominators around.
- A white_node u changes to the dominatee state if it receives a "HELLO" message from a dominator v, and v has a larger node degree D_v than u.
- A dominator u becomes a dominatee if a "HELLO" message is received from another dominator v, and v has a larger node degree D_v than u.
- A dominatee or a connector changes back to the white_node state if it has lost all dominators around.

Node ID is used whenever a tie happens in above steps. Nodes in the dominator state formulate an MIS in the network since no two adjacent nodes will be marked as dominators, and these dominators operate as gateways to relay packets throughout the network.

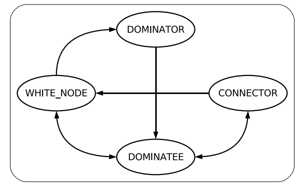


Fig. 2. The state transition diagram of a node in our algorithm.

3.2. Generating a CDS

In this phase, each gateway calculates an MPR set based on the original MPR algorithm discussed in the previous section. Due to the limited space, the MPR selection procedures are not repeated here. Similar to the original MPR algorithm, node IDs of selected MPRs are included in the "HELLO" messages sent by gateways. Upon receiving these "HELLO" messages, a dominatee determines its dominating state based on the following steps:

- A dominatee u changes its state to the connector if it is selected as an MPR by a dominator v, which has the largest D among all u's dominators.
- A connector u returns to the dominatee state if u's largest dominator (dominator with the largest D) does not choose it as the MPR.

Only nodes in the connector state retransmit packets and they ensure the connectivity of gateways. Finally, a CDS is constructed combining all gateways and connectors in the network.

3.3. Self-pruning procedure

A dominator u eliminates itself from the CDS if it has a connector v that is in $N^1(u)$, and v can cover all u's one-hop neighbors. An eliminated dominator u is referred to as a *silent-dominator*, which still announces itself as the dominator and calculates the MPR set. However, u will not retransmit packets anymore.

The self-pruning procedure can effectively reduce the number of gateways in the network, thus further limiting the size of the CDS generated in our algorithm.

Figure 3 demonstrates an example of constructing a CDS using our new algorithm. The letter near each node represents the node ID, and arrows represent "HELLO" messages and their sending directions. A CDS is constructed in following steps:

1. Initially, all nodes in this network are in the white_node state as shown in Figure 3(a). After knowing its neighborhood information, node g announces itself as the dominator because it has a larger node degree D than all its white_node neighbors. Then g calculates an MPR set to cover its two-hop neighbors. Based on the original MPR algorithm, node b is selected as the MPR due to the fact that it covers g's only two-hop neighbor node a, and b has a smaller node ID than node d. After the MPR calculation, node g immediately broadcasts a "HELLO" message to inform all its one-hop neighbors about its new dominating state and its MPR decision. The above procedures are shown in Figure 3(b).
2. Upon receiving g's "HELLO" message, g's one-hop neighbors announce themselves as dominatees, and node b further changes to the connector state because its only adja-

cent dominator g has chosen it as an MPR. Then all dominatees immediately broadcast a "HELLO" message to indicate their new dominating states to their one-hop neighbors. After receiving the "HELLO" messages from both dominatee node b and d, node a declares itself as the dominator because it has no white_nodes around. Then it also calculates an MPR set to cover its two-hop neighbors, which are node f, g and h in this case. Obviously, node a will selects node d as the MPR because it covers more two-hop neighbors of a than node b. After the MPR calculation, dominator a also sends out a "HELLO" message to all its one-hop neighbors indicating its dominating state and its MPR decision. Figure 3(c) illustrates above processes.

3. Upon receiving the "HELLO" message from a, both node b and d have the knowledge of this new dominator, and node d also notices that it is chosen as an MPR by a. However, node d ignores a's MPR decision due to the fact that a is not the largest dominator around d (D_a is smaller than D_g). We present this in Figure 3(d) by putting a cross on the corresponding arrow. Therefore, only node b operates as the connector to connect dominator a and g. At the same time, both the dominators run the self-pruning procedure to evaluate themselves, and consequently, a is removed from the CDS and becomes a *silent-dominator* because all its one-hop neighbors can be covered by its connector b.

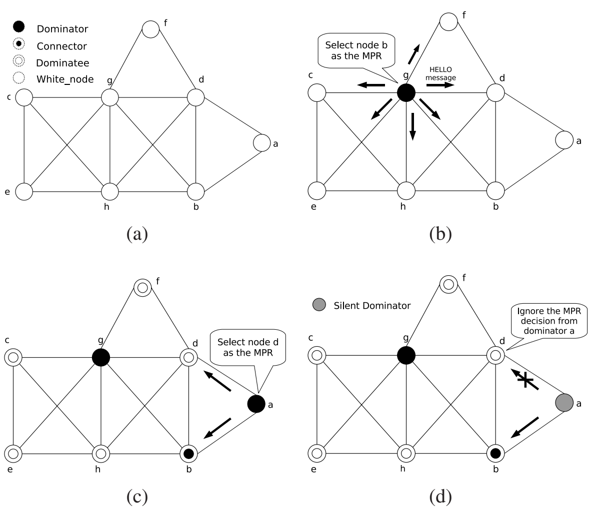


Fig. 3. An example of constructing a CDS using GMPR.

4. PROOF OF CORRECTNESS

In this section, we verify the correctness of our algorithm and we also provide its time and message complexity.

4.1. Proof of correctness

The correctness of our algorithm is proven in two parts. First, we prove that the gateways generated in our algorithm cover all nodes in the network, and thus they form a DS (dominating set). Second, we prove that the DS consists of the gateways and the connectors generated in our algorithm is indeed a CDS.

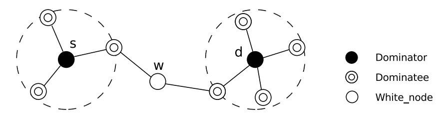


Fig. 4. Illustration of the proof of Lemma 1.

Lemma 1. *Let H be the set of gateways generated in our algorithm. For each node u in the network, u is either in H or a neighbor of at least one node in H.*

Proof. This lemma essentially indicates that H is actually a DS in the network. We prove this lemma by contradiction. Assume that there is a node w, which is not in H or has any neighbor in H. Then w must be at least two hops away from any given gateway in the network. This situation is shown in Figure 4, where node s and d are gateways in H, and the dash circle represents the node transmission range. In this network, node w will eventually announce itself as a dominator based on our algorithm, because it dos not have any dominator or white_node nearby. Therefore, node w is actually a gateway and it should be in H. This result contradicts to the assumption, and thus it proves the lemma. □

Lemma 2. *For any given gateway s and its nearest gateway d, s is at most three hops away from d.*

Proof. This can be easily demonstrated based on Lemma 1. Assume that s is four hops away from d as shown in Figure 4. In such a case, node w will elect itself as the gateway. Therefore, d is not the nearest gateway of s and the lemma is proven. □

Lemma 3. *All gateways are connected through the connectors generated in our algorithm.*

Proof. We prove this by contradiction. Let u denote the gateway that does not have any connectors to other gateways. Based on Lemma 2, u is at most three hops away from its nearest gateways. If u has some gateways that are two hops away as shown in Figure 5(a), let W denote the set that contains u and u's two-hop away gateways, there must be a node in W that has the largest node degree D (use node ID to break a tie), and it will choose at least one node in $N^1(u)$ as the MPR to cover its two-hop neighbors. Based on our algorithm, the selected MPR is actually a connector, and thus, u is connected to at least one two-hop away gateway.

If u only has three-hop away gateways as shown in Figure 5(b). u selects some nodes in $N^1(u)$ as MPRs to cover all nodes in $N^2(u)$, and based on our algorithm, these selected MPRs are actually connectors. Similarly, the three-hop away gateways also select some nodes in $N^2(u)$ as MPRs to cover their two-hop neighbors, and at least one MPR they selected becomes a connector. Therefore, connectors in $N^2(u)$ can be reached by at least one connector in $N^1(u)$, and thus u is connected to at least one three-hop away gateway through two connectors. □

Theorem 1. *Given a connected graph $G = (V, E)$, a node set V', which consists of gateways and connectors generated in our algorithm, is a CDS of G.*

Proof. The theorem is proven instantly by combining Lemma 1 and Lemma 3. □

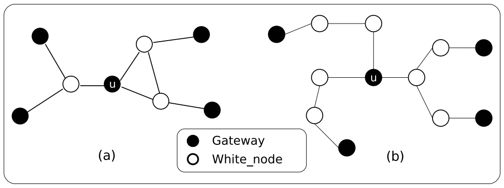


Fig. 5. Illustration of the proof of Lemma 3.

Theorem 2. *After the self-pruning procedure, the connectors and the remaining dominators generated in our algorithm still form a CDS.*

Proof. Because a *silent-dominator* still announces itself as the dominator and calculates an MPR set, its elimination from the CDS does not affect the construction of connectors. Furthermore, the connector that covers all one-hop neighbors of a *silent-dominator* can still ensure the connectivities of other connectors of that *silent-dominator*. Therefore, the property of the CDS is maintained after the self-pruning procedure. □

4.2. Time and Message complexity

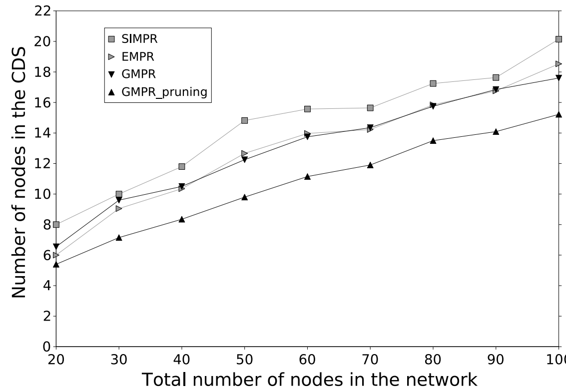
Theorem 3. *Our new distributed broadcast algorithm for constructing a CDS has an $O(\Delta^2)$ time complexity and $O(n)$ message complexity.*

Proof. The time complexity for constructing an MIS in our algorithm is $O(n)$, which can be proven using the similar method in [6]. For each gateway, it takes maximum $O(\Delta^2)$ time to calculate an MPR set, and it also takes the same time to run the self-pruning procedure. However, we exclude the self-pruning procedure from the overall time complexity calculation, because it is operated after the CDS construction, and it can be run simultaneously by all gateways. It has also been proven in [7] that at most 5 dominators can connect to a given dominatee node, and thus, a constant time is needed for a dominatee to determine whether it is a connector. Among above procedures, we estimate that the time for the MPR calculation will be the dominant part of the overall time consumed in our algorithm. Hence, the time complexity of our algorithm is $O(\Delta^2)$.

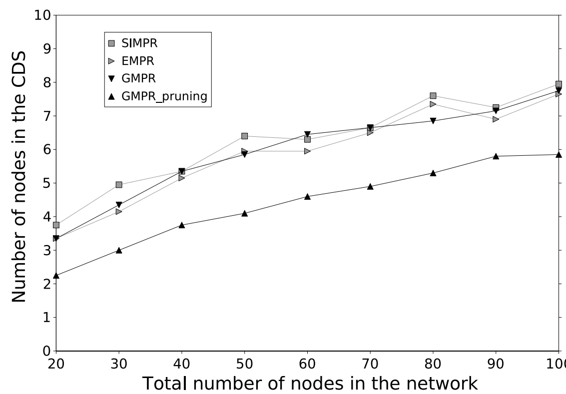
During the CDS construction of our algorithm, each gateway sends exactly one "HELLO" message to its one-hop neighbors in order to inform its dominating state and its MPR decision. Upon receiving these "HELLO" messages, each dominatee or connector also sends one "HELLO" message to its one-hop neighbors. Hence, at most five "HELLO" messages will be sent for each dominatee or connector during the CDS construction. Therefore, the total message complexity of our algorithm is $O(n)$. □

5. SIMULATION RESULTS

In this section, we present simulation results that compare the average size of the CDS generated in our new algorithm, Adjih's source-independent MPR(SIMPR) [4] and Wu's enhanced MPR (EMPR) [5]. For the sake of comparison, we also present the results of our algorithm without the self-pruning procedure and with the self-pruning procedure separately, and we denote them as the "GMPR" and "GMPR_pruning" respectively.



(a) Transmission range $R = 25$



(b) Transmission range $R = 50$

Fig. 6. The number of nodes in the CDS vs. the total number of nodes in the network.

In our simulations, each node has an equal transmission range R, and a pair of nodes can be connected if the distance between them is less than R. The topologies in our simulations are generated by randomly distributing a certain number of nodes in a 100 $\times$ 100 2-dimensional space. To ensure the connectivity of the network, each node is randomly placed within the transmission range of a previously located node, which is also chosen randomly. Two transmission ranges (R = 25 and 50) are used in our simulations to create sparse and dense networks, and for each transmission range, the total number of nodes N in the network varies from 20 to 100 with an interval of 10. A sufficient number of simulation runs are conducted for each N to achieve a 95% confidential interval within a $\pm 5\%$ margin. The averaged results are displayed in Figures 6(a) and 6(b).

From the simulation results, we can see that our algorithm without the self-pruning procedure has the similar performance as the EMPR in both transmission ranges. After applying the self-pruning procedure, our algorithm improves significantly and performs much better than the two proposed algorithms in both transmission ranges. It is also worth noting that our algorithm with the self-pruning procedure works more effectively in the dense network (when $R = 50$), where it reduces the size of the CDS generated in the EMPR by 26% on average. This is mainly due to the fact that in the dense network, the number of the gateways generated in our algorithm is reduced when the number of one-hop neighbors of a node increases. Furthermore, a longer transmission range can also increase the chance for a connector to cover all one-hop neighbors of a gateway, thus eliminating more redundant gateways in the CDS.

6. CONCLUSION

In this paper, we have proposed a new distributed broadcast algorithm for constructing a connected dominating set (CDS) in a given network. Our algorithm is based on the concepts of the Multipoint Relay (MPR) and the minimum independent set (MIS), where nodes in the MIS perform as gateways in the network and they calculate MPRs to connect each other. A self-pruning method is also introduced to further eliminate redundant gateways in the CDS. The simulation results show that our new algorithm produces a smaller size CDS than the source-independent MPR [4] and the enhanced MPR [5] in both sparse and dense networks.

The future work is to compare our algorithm with other MIS-based CDS broadcast algorithms, and we are also going to investigate the performance of our algorithm in a mobile environment.

7. REFERENCES

[1] "IETF mobile ad hoc network working group." [Online]. Available: http://www.ietf.org/html.charters/manet-charter.html

[2] L. Yang, S. Conner, X. Guo, M. Hazra, and J. Zhu. (2003, Oct.) Common wireless ad hoc network usage scenarios. Internet Draft. [Online]. Available: http://www.flarion.com/ans-research/Drafts/draft-irtf-yang-ans-scenarios-00.txt

[3] A. Laouiti, A. Qayyum, and L. Viennot, "Multipoint relaying: an efficient technique for flooding in mobile wireless networks," in *35th Annual Hawaii International Conference on System Sciences HICSS'2001*.

[4] C. Adjih, P. Jacquet, and L. Viennot, "Computing connected dominated sets with multipoint relays," Technical Report, INRIA, Oct. 2002. [Online]. Available: www.inria.fr/rrrt/rr-4597.html

[5] J. Wu, "An enhanced approach to determine a small forward node set based on multipoint relays," in *IEEE Transactions on Parallel and Distributed Systems*, Sep. 2002, pp. 866–881.

[6] K. Alzoubi, P.-J. Wan, and O. Frieder, "Message optimal connected dominating set construction for routing in mobile ad hoc networks," in *Proceedings of Third ACM int'l Symp.* Mobile Ad Hoc Networks and Computing (MobiHoc '02), 2002.

[7] K. Alzoubi, X.-Y. Li, Y. Wang, P.-J. Wan, and O. Frieder, "Geometric spanners for wireless ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 14, pp. 408–421, April 2003.