Efficient Construction of Connected Dominating Set in Wireless Ad Hoc Networks*

Bo Han, HaoHuan Fu, Lidong Lin and Weijia Jia
Department of Computer Engineering and Information Technology
City University of Hong Kong, 83 Tat Chee Avenue, Kowloon, Hong Kong
Email: hanbo00@cityu.edu.hk

Abstract

In the absence of infrastructure and facing the dynamically changing topology, efficient message routing is one of the most important issues in wireless ad hoc networks. Connected dominating set based routing is a promising approach for enhancing the routing efficiency in the wireless ad hoc networks. However, finding the minimum dominating set in an arbitrary graph is a NP-hard problem. In this paper, we propose a simple and efficient distributed algorithm for constructing a connected dominating set in wireless ad hoc networks with time complexity $O(n)$ and message complexity $O(n\log n)$ respectively. The dominating set generated from our algorithm can be more reliable and load balanced for routing as compared with some well-known algorithms. The simulation results demonstrate that our algorithm outperforms the previous work in terms of the size of resultant connected dominating set.

Key words: Ad-hoc networks, dominating set, distributed algorithm, routing

1. Introduction

A network is normally modeled as a limited graph $G = (V, E)$ where V and E are limited sets of nodes and edges in the graph. In G, define a vertex u dominates another vertex v if u=v or if u and v are adjacent. A subset of vertices of G is a dominating set if it collectively dominates all vertices in the graph. Thus, for every node v, either v is in V_0 or a direct neighbor of v is in V_0. The minimum dominating set (MDS) problem asks for a dominating set of minimum size. The size of this smallest dominating set is called the domination number of the graph. The dominating set problem and the closely related set cover problem can be used to model many important problems for networks and communications. Unfortunately, both of the two problems have been proved to be NP-hard [2]. In fact the dominating set problem is Quasi-NP-hard to approximate within a ratio of $(\log n)/48$ where n is the number of nodes [3].

A particular application of dominating set problem can be found in the fast growing field of wireless ad hoc networks. A wireless ad hoc network is a collection of mobile devices dynamically forming a temporary network without any existing infrastructure. Some examples of the possible uses of wireless ad hoc networks include participants using laptops to discuss some issues in a conference hall, business associates using notebooks to share information during a meeting, and soldiers using wearable computers to exchange situational information on the battlefield. Due to the limited transmission range of wireless network interfaces, when a message is being sent from one node to another, intermediate network nodes might be needed to serve as routers to relay this message. Although lots of interesting schemes have been proposed, finding efficient routing algorithms remains the most important problem for wireless ad hoc networks. Grouping nodes into clusters can be an effective way to improve the performance of routing algorithms. Then the routing can be done between clusters. The most basic method for clustering is to construct a dominating set. All the nodes in the dominating set act as routers, and other nodes communicate via their neighbors in the dominating set. Obviously, in wireless ad hoc network, the minimum dominating set is required to be a connected one which is called the *minimum connected dominating set* (MCDS).

* The work is supported by Research Grant Council (RGC) Hong Kong, SAR China, under grant nos.: CityU 1055/00E and CityU 1039/02E and CityU Strategic grant nos. 7001446 and 7001587.

Two main distinctions between traditional wired networks and wireless ad hoc networks are: 1) the topology of wireless ad hoc network changes very frequently since wireless devices are mobile and 2) wireless devices typically have limited power and much lower bandwidth than their wired counterparts. As a consequence, algorithms for wireless ad hoc networks should have following properties: a) distributed, which means only local information is required, b) adaptive to the dynamic change of topology, c) low communication overhead with a smaller of number of communication rounds. So the algorithm for MCDS construction must satisfy above requirements. In this paper, we present a simple and efficient distributed approximation algorithm for MCDS construction. The time complexity of our algorithm is $O(n)$ and message complexity is $O(n\log n)$. In addition, it is more suitable for load balance routing in wireless ad hoc networks than the existing algorithms (to be discussed in Sections 2 and 3).

The remainder of the paper is organized as follows: Section 2 introduces the related work. The description and analysis of our distributed algorithm are presented in Section 3. Section 4 gives the experimental results. We point out future directions and summarize our major results in Section 5.

2. Related Work

Many well-known distributed MCDS construction algorithms have been proposed for wireless ad hoc networks [4, 5, 8, 9, 10, 11, 15, 16, 18]. These works can be divided into two categories. The first kind constructs a Connected Dominating Set (CDS) initially and then further reduces the size of CDS. The most representative algorithm is given in [15]. An extreme construction is presented in [10], in which the whole connected graph is viewed as a CDS. Their algorithm removes nodes from the graph one by one. Each time one node is removed, a distributed algorithm runs to verify whether the rest of the graph is still connected. The second kind first constructs a dominating set and then makes the components connected [8, 9].

A greedy heuristic algorithm has been presented in the Laboratory for Information and Decision Systems of MIT [4] which picks out a dominating set with a size no more than $n-\sqrt{2m+1}$. Das et al. proposed a series of routing algorithms for wireless ad hoc networks [5, 6, 7]. Their algorithms identify a sub-network that forms a MCDS. Each node in the sub-network is called as a spine node or backbone node. The algorithm proposed by Das et al. is a distributed version of Guha and Khuller's approximation algorithm for calculating the minimum connected dominating set [1]. Compared with previous work, this algorithm has two major advantages: (1) each node only needs 2-distance neighborhood information; (2) the algorithm runs $O(\gamma)$ rounds. The overall complexities are $O(\gamma\Delta^2+n)$ in time and $O(\Delta n\gamma + m + n\log n)$ in number of messages where γ is the number of nodes in the resultant dominating set and Δ is the maximum node degree. A main drawback of this algorithm is that it still needs a non-constant number of rounds to determine a connected dominating set.

The algorithm proposed by Wu and Li in [15] first finds a CDS and then prunes certain redundant nodes from the CDS. The initial CDS U consists of all nodes that have at least two non-adjacent neighbors. A node u in U is considered to be locally redundant if either it has a neighbor in U that has a larger ID and dominates all other neighbors of u (Rule 1), or it has two adjacent neighbors in U that have larger IDs and together dominate all other neighbors of u (Rule 2). The message complexity and time complexity of this distributed algorithm are $O(n^2)$ and $O(n^3)$ respectively. The advantage of this algorithm is that it is very simple, and without the pruning process based on Rule 2, it can guarantee that the shortest path between any two nodes does not include any non-dominating node as an intermediate node. Unfortunately, the approximation factor of this algorithm is at most $n/2$, instead of a constant, and the size of resultant dominating set relies on the ID assignment of each node. For example, in pruning process if some node u in U has a neighbor (that is also in U) that dominates all other neighbors of u but with a smaller ID, the node u will not be viewed as the redundant node.

Construction of MCDS is a bit like cluster formation and leader election in distributed systems. In wireless ad hoc networks, some nodes need to be elected to relay the messages generated by other nodes. One way to accomplish this is to select certain "strategically located" nodes as leaders. Typically, nodes that can communicate with many others, i.e., those with high connectivity, should be selected. Every other node is assigned to one of these leaders so that the network is partitioned into clusters and this architecture was first proposed by Baker and Ephridemes [12]. Stojmenovic et al. presented a distributed construction of CDS in the context of clustering broadcast [16]. The CDS consists of two types of nodes: the cluster heads and the boarder nodes. The cluster heads form a Maximal Independent Set (MIS), i.e., a dominating set in which any pair of nodes is non-adjacent. Although no implementation details are given, it is pointed out in [8] that the time complexity and message complexity of this construction are $O(n^2)$, and the approximation factor is not a constant either.

Based on algorithm described in [16], Alzoubi and Wan proposed a more efficient distributed algorithm [8] with time and messages complexity $O(n)$ and $O(n\log n)$ respectively. Their algorithm constructs two trees, one for the level assignment of each node to construct a MIS and the other for connecting the nodes in the constructed MIS. In [9], the same authors proposed two modified algorithms: ID-Based algorithm and Level-Based algorithm. The time and message complexities of both algorithms remain the same and the approximation factors are 12 and 8 respectively. Unfortunately, in these algorithms, only theoretical analysis is given, while they are hardly used to handle a realistic situation where the mobile nodes are moving randomly. In addition, the aim of CDS construction is to conduct more efficient routing. Because the nodes in the final dominating set constructed by these algorithms form a tree, if two leaf nodes with long distance in the graph want to communicate with each other, more nodes (may be unnecessary) in the tree will be selected to relay the messages.

3. Our Distributed Algorithm

As indicated in Section 1, wireless ad hoc network can be viewed as a connected graph $G = (V,E)$, where V is the set of nodes and E is the set of edges. All edges are bidirectional. Nodes communicate by transmitting messages using common wireless channels. Each node has a fixed transmission range called its neighborhood. If two nodes are able to hear each other's transmission, they are said to be neighbors and are connected with an edge in the graph. A transmitted message can be heard only by a group of neighbors directly. In the case of wireless networks, we assume equal transmission capacity on both sides. The topology of such wireless ad hoc network can be modeled as a unit disk graph (UDG) [14], a geometric graph in which there is an edge between two nodes if and only if their distance is at most one. In the following description, we use n and m to stand for the number of nodes and the number of edges respectively.

3.1. Rank Assignment

The dominating set construction in our distributed algorithm is based on the construction of MIS. An independent set V' in G is a subset of V such that no two vertices in V' are joined by an edge in E. The maximum independent set problem asks for an independent set of maximum size. By definition, any pair of nodes in the MIS is separated by at least two hops.

However, with an elaborately chosen rank, every node in our constructed independent set can find at least one 2-hop-away neighbor that is also in the independent set. To assign the rank, we need to construct a spanning tree T rooted at some node v. This node can be designated through extra approach. For example, the root can be a session chair's notebook in a conference or a commander's wireless device in a battlefield. If it is difficult to designate any node as the root, a distributed leader election algorithm can be applied to find out a root. The algorithm given in [13] is one of the best algorithms with time complexity $O(n)$ and message complexity $O(n\log n)$. After the root is assigned, each node can identify its level with respect to T as follows: the root first announces its level as 0. Every other node, upon receiving the level announcement message from its parent in T, obtains its own level by increasing the level of its parent by one, and then announces its level. Each node also records the levels of its neighbors. In the following algorithm, we take the node's level as its rank.

3.2. Algorithm for CDS construction

We assume all nodes are unmarked initially. After each node has its rank, the CDS can be constructed in the following steps:

- Initially, the root node of the spanning tree declares itself as a dominating node by broadcasting a Declare-Dominating message and marks itself *black*. The Declare-Dominating message includes the dominating node's ID.
- Whenever a node receives a Declare-Dominating message for the first time, it declares itself as a dominated node by broadcasting an Abandon-Dominating message, marks itself *white* and records the ID in the received Declare-Dominating message. If the node receives a Declare-Dominating message from a different dominating node later, it only records the ID in the received message.
- Whenever a node has received the Abandon-Dominating messages from all lower ranks neighbors, it declares itself as a dominating node by broadcasting a Declare-Dominating message and marks itself *black*.
- When a leaf node in the spanning tree is marked, it transmits a DS-COMPLETE message to its parent. Each internal node will wait until it receives this DS-COMPLETE message from each

of its children and then forwards it up along the tree toward the root.

- After each dominated node has transmitted a DS-COMPLETE message and received more than one Declare-Dominating message, it sends a REPORT message that includes the recorded adjacent dominating nodes' ID to its neighboring dominating nodes accordingly.
- When a dominating node receives the first REPORT message about some adjacent dominating node from a neighbor *w*, it will notify *w* to join the dominating set and *w* will mark itself *gray*. The gray color is used to distinguish from the black color above, with which the dominating nodes are marked.

An example of the algorithm execution is illustrated in Figure 1* and explained below. Let node R7 be the root node (in fact, R7 is selected as the leader by algorithm in [13]).

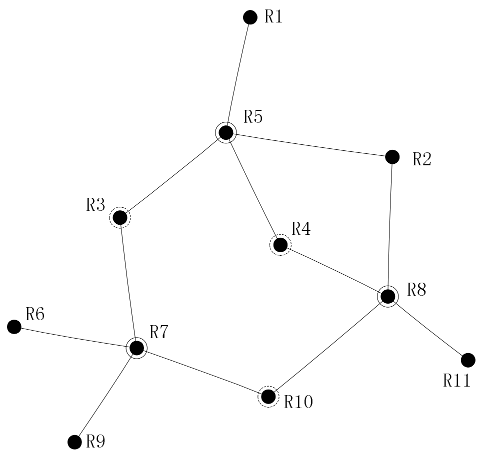

Figure 1: The example of wireless ad hoc networks

1. Node R7 marks itself black and sends a Declare-Dominating message.
2. Upon receiving a Declare-Dominating message from node R7, nodes R3, R6, R9 and R10 mark themselves white and transmit an Abandon-Dominating message.
3. Upon receiving an Abandon-Dominating message from node R3, node R5 marks itself black and sends a Declare-Dominating message, as all its low-ranked neighbors (in this example, node R3 only) have transmitted an Abandon-Dominating message. Similarly, upon receiving an Abandon-Dominating message from node R10, node R8 marks itself black and sends a Declare-Dominating message.
4. Upon receiving a Declare-Dominating message from node R5, node R1, R2 and R4 mark themselves white and send an Abandon-Dominating message. Similarly, node R11 marks itself white and sends an Abandon-Dominating message upon receiving a Declare-Dominating message from node R8.
5. When nodes R1, R2, R4, R6, R9, and R11 are marked, they send DS-COMPLETE messages and the messages are forwarded up along the spanning tree toward node R7.
6. After nodes R2, R3, R4, and R10 have sent the DS-COMPLETE messages and received more than one Declare-Dominating message, they report to their adjacent dominating nodes all the neighboring dominating nodes they know. At the last, nodes R3, R4 (or R2) and R10 are added to the final dominating set.

3.3. Analysis of our algorithm

In this subsection, we will show some properties of our algorithm.

Lemma 1*: Every node marked black has at least one 2-hop-away neighbor that is also marked black if there exists any other black dominating node.*

Proof: Except the root node, every other dominating node *v* marks itself black only when all of its neighbors with lower ranks become the dominated nodes. The condition that these nodes can become dominated nodes is that one of their neighbors, say *u*, becomes a dominating node. Obviously *u* is one of the 2-hop-away black neighbors of *v*. As to the root node, if there is any other black dominating node, one of them must be the child of some child of the root node. This completes the proof.

Theorem 1*: The nodes that are marked black or gray in our algorithm form a CDS.*

Proof: Because each white node is dominated by one of its black neighbors and we connected any pair of 2-hop-away black nodes using one of their common neighbors which will be marked as gray, this theorem can be proved trivially based on Lemma1.

Lemma 2*: Let G be a unit disk graph and opt be the size of a minimum dominating set for G, then the size of any maximum independent set for G is at most* $5 \times opt$.

Proof: Marathe et al. presented a detailed proof based on an idea that no vertex in a minimum

* In this graph, the rank of node R7 is 0, the ranks of nodes R3, R6, R9 and R10 are 1, the ranks of nodes R5 and R8 are 2, and the ranks of nodes R1, R2, R4 and R11 are 3.

dominating set can dominate more than 5 vertices in a maximum independent set [22].

Lemma 3*: Let set* $U = \{u_i\}$ *where* u_i *is the node that is marked black and H be the graph over U in which a pair of nodes is connected by an edge if and only if their graph distance in G is two, then every node in H has at most 18 neighbors.*

Proof: The proof of the lemma is equivalent to prove that one can place at most 18 circles on a circular ring, whose inner radius is 1 and outer radius is 2. We place* the circles based on the following rules: (1) on the outer boundary but not on the inner boundary; (2) the center of each circle is in the circular ring but not in any other placed circle. One of the biggest placement examples is shown in Figure 2, in which 12 circles are placed on the outer boundary and the other 6 circles can be placed alternatively on the left 12 interspaces (as the area marked A in Figure 2) that are not covered by those twelve circles. To give a clear effect, we only show 1 out of 6 circles in Figure 2. For simplicity, we omit the proof details.

Theorem 2*: The size of MCDS constructed by our algorithm is at most* $50 \times opt$.

Proof: Because the black nodes form a MIS, by Lemma 2, we know that the number of these black nodes is at most $5 \times opt$. In term of Lemma 3, we know

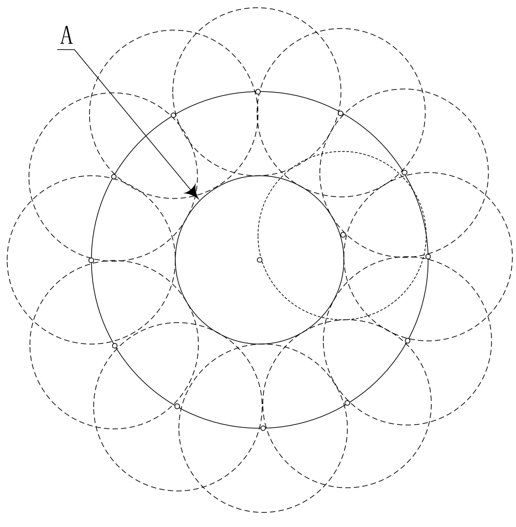

Figure 2: An example topology of wireless ad hoc networks by UDG

that for one black node, at most 18 gray nodes can be its neighbors and each connects to another black node. Thus the number of gray nodes is at most 5×18/2 $= 45 \times opt$. Add the two values, the total size of the final dominating set is thus at most $50 \times opt$.

In order to compute and update routes, we need to keep the size of CDS as small as possible. However, the mobile devices in wireless ad hoc networks typically have limited power. If the size of the CDS is small, the nodes in CDS may consume more power when the routing is based on a CDS, as the nodes in the CDS need to relay more traffic. This consumption of more power in the dominating nodes will result in a shorter lifetime for the whole network. It has been shown that, when using IEEE 802.11 network interface, the ratio of power consumption in transmit, receive, idle and sleeping modes is close to 14:10:8:1 [17].

Now we face a dilemma: to reduce the complexity of routing, we require the size of CDS to be as small as possible; but considering the power consumption of the mobile nodes, we hope that the routing is shared by more nodes and this demands the size of CDS not to be too small. To solve this conflict problem, our algorithm allows the routing be computed and maintained among the nodes marked black rather than relying on all nodes in CDS, while the traffic can be relayed on the whole CDS.

Thus we can illustrate the difference of our algorithm from the previous algorithms through the following example. For the network shown in Figure 1, the algorithm in [8] will first construct an independent set, including nodes R5, R7 and R8. After the second stage, connecting the nodes in the independent set, nodes R3 and R4 will be added to the dominating set to form a dominating tree. So at last the CDS consists of nodes R3, R4, R5, R7 and R8. Although the size seems smaller, if node R6 wants to send a message to R11, nodes R7, R3, R5, R4 and R8 would be required to relay this message. But obviously {R6, R7, R10, R8, R11} is another feasible path with fewer nodes. Thus, R10 should be added to the CDS to enable a load balanced routing. In our algorithm, to connect nodes R7 and R8, node R10, as their common neighbor, is added to the final CDS.

Although each pair of black dominating nodes only brings one of their common neighbors to the final CDS in our algorithm, they are fully aware of all their common neighbors in addition to the one in CDS. Thus, our algorithm may have alternative choice in terms of load balance and reliability for routing the messages. For example, as shown in Figure 1, nodes R5 and R8 know that they can communicate with each other through node R2 or R4. To relay traffic between R5 and R8, the load could be evenly distributed between R2 and R4. Moreover, due to the dynamic feature of wireless ad hoc networks, if R2 switches off suddenly,

* By placing a circle on somewhere, we mean to place the center of the circle at that place.

R4 can be designated to join the dominating set to keep the connectivity. This makes the reformation of CDS more efficiently.

Theorem 3*: Our algorithm is O(n) in time complexity and O(nlog n) in message complexity.*

Proof: For the rank assignment the time complexity is $O(n)$ and message complexity is $O(n\log n)$, which is dominated by the leader election or spanning tree generation. For the dominating set construction, each node sends at most two messages, Declare-Dominating message or Abandon-Dominating message and DS-COMPLETE message. This requires $O(n)$ messages in $O(n)$ time. The same is true for adding more intermediate nodes to the final connected dominating set. Therefore, the overall time and message complexities of our algorithm are $O(n)$ and $O(n\log n)$, respectively.

4. Experimental Results

In this section we perform the simulation that compares the average size of the dominating sets generated by our algorithm with Marking Process proposed by Wu and Li in [15] and the Level-Based algorithm (which has the smaller approximation factor than the other ID-Based algorithm) proposed by Alzoubi and Wan [9]. The simulation uses the following parameters.

N: the number of mobile nodes in the network;

D: the number of nodes in the connected dominating set, and

r: the radius of mobile nodes' transmission range.

Denote MCDS as the size of dominating set constructed by our algorithm, Marking-Process, the size of dominating set constructed by Marking Process with Rule 1 and 2 [15], and Level-Based, the size of dominating set constructed by Level-Based algorithm [9], respectively. Random graphs are generated in a 200×200 square area as the ad hoc networks. We randomly generate a certain number of mobile nodes and place them into this area. A pair of nodes can be connected if their distance is less than radius r. Moreover, we only use the connected graph for simulating the networks thus we do not consider the graph with islands.

We performed two groups of simulations. In the first group, we set the number of mobile nodes N = 80, 200, 300, and 500 respectively. For each N, we vary transmission radius of mobile nodes r accordingly. For each round of simulation, we randomly generate 250 connected graphs and calculate the Marking-Process, Level-Based and MCDS for these graphs and take the

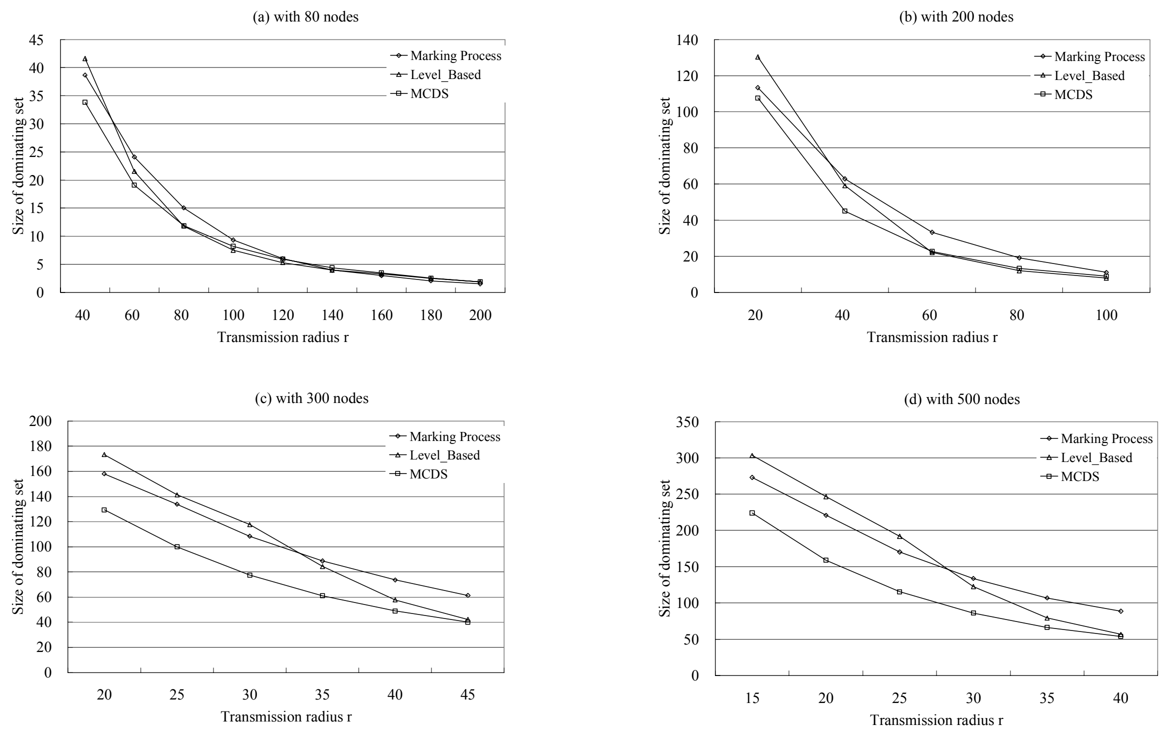

Figure 3: Average size of dominating set relative to the transmission radius

average values for each algorithm results. Figure 3(a-d) shows the size of the dominating set versus the transmission radius *r* for the increasing the number of nodes in the networks. From Figure 3(a), we can see that, when the transmission radius is large (relative to the simulation area), the three algorithms perform closely. In this situation, because the graphs are well-connected, fewer nodes are selected into the dominating set. When the transmission radius becomes small, as shown in Figure 3(b), (c) and (d) our algorithm outperforms the other two algorithms. The reason is that, in the Level-Based algorithm, a dominated node changes into a dominating node when one of its children becomes a dominating node. In some cases, this will add more redundant nodes to the dominating set. On the other hand, the Marking Process relies on the ID assignment of each node and this may also introduce some redundant nodes. Also we can see from Figure 3(b), (c) and (d) that, the gap between MCDS and Marking-Process (the smaller one of the other two) increases as N increases. When N = 200, r=15, the gap is 6 and when N = 500, r = 15, the gap is almost 50. To further demonstrate our results, in the second group of simulation, we set transmission radius of mobile nodes r to two different values: 40 and 60. For each r, we also vary the number of mobile nodes N. For the same reasons as given above, we can see that our algorithm outperforms the other two algorithms from Figure 4(a) and 4(b).

5. Conclusion and Future Work

In this paper, we have proposed a simple and efficient distributed algorithm for the construction of connected dominating set in wireless ad hoc networks. In our algorithm, we connect any pair of 2-hop-away nodes in the constructed MIS using one of their common neighbors. The time complexity of this algorithm is $O(n)$ and message complexity is $O(n\log n)$. Our algorithm has advantages of balancing the load and enhancing the reliability for routing messages. The simulation results show that compared with Marking Process algorithm in [15] and Level-Based algorithm in [9], the size of our generated connected dominating set is the smallest when the mobile nodes' transmission radius is not too large.

As mentioned above, dynamic topology is one of the most specific features of wireless ad hoc networks. The distributed algorithms in wireless ad hoc networks must handle the topology changing promptly. The topological changes can be summarized into three different types: mobile device's switch on, mobile device's switch off, and mobile device's movement [15]. Our algorithm can update the MCDS efficiently with level reassignment, cluster reformation, and cluster head re-election etc. Moreover, the MCDS construction may be extended to the design for multicast or anycast routing in wireless ad hoc networks where MCDS can be incorporated into well-known routing algorithms such as DSDV [19], DSR [20], and AODV [21].

Acknowledgements

We are grateful to Pui-On Au, Ji Shen and Man-Ching Yuen for their valuable comments on this paper.

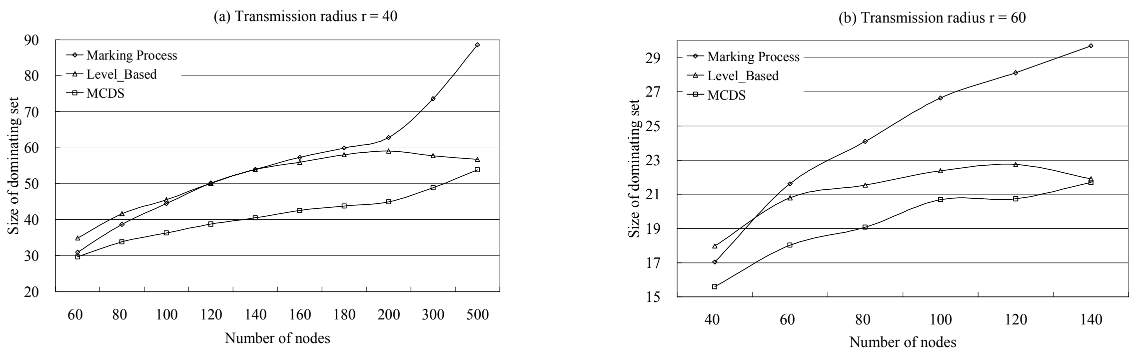

Figure 4: Average size of dominating set relative to the number of nodes

References

[1] Sudipto Guha and Samir Khuller, “Approximation Algorithms for Connected Dominating Sets”, The 4th Annual European Symposium on Algorithms (ESA), (Sep '96), Barcelona, Spain. Algorithmica Vol. 20, 1998, pp. 374-387.

[2] M. Garey and D. Johnson, “Computers and Intractability: A Guide to the Theory of NP-Completeness”, Freeman, San Francisco, 1979.

[3] Sanjeev Arora and Carsten Lund, “Hardness of Approximations”, in D. S. Hochbaum, editor, “Approximation Algorithms for NP-hard Problems”, pages 399–446, PWS, Boston, MA, 1997.

[4] Abhay K. Parekh, “Finding small dominating sets in stationless mobile packet radio networks”, Tech. Rep. LIDS-P-2050, Laboratory for Information and Decision Systems, M.I.T., Cambridge, MA, 1991.

[5] B. Das and V. Bharghavan, “Routing in ad-hoc networks using minimum connected dominating sets”, IEEE International Conference on Communications, ICC'97, Jun, 1997.

[6] B. Dss, E. Sivakumar, and V. Bhargavan, “Routing in ad-hoc networks using a virtual backbone”, Proceedings of the 6th International Conference on Computer Communications and Networks(IC3N ‘97), pages 1 - 20, Sept. 1997.

[7] B. Das, R. Sivakumar, and V. Bhargavan, “Routing in ad-hoc networks using a spine”, IEEE International Conference on Computers and Communications Networks ‘97, 1997.

[8] K. M. Alzoubi, P.-J. Wan, and O. Frieder, “New distributed algorithm for connected dominating set in wireless ad hoc networks”, HICSS35, Hawaii, Jan. 2002.

[9] K. M. Alzoubi, P.-J. Wan and O. Frieder, “Distributed Heuristics for Connected Dominating Set in Wireless Ad Hoc Networks”, IEEE ComSoc/KICS Journal on Communication Networks, Vol. 4, Issue 1, March 2002, pp. 22-29.

[10] Sergiy Butenko, Xiuzhen Cheng, Carlos A.S. Oliveira, and Panos M. Pardalos, “A New Algorithm for Connected Dominating Sets on Ad Hoc Networks”, Recent Developments in Cooperative Control and Optimization, Edited by S. Butenko, R. Murphey, and P. Pardalos, Kluwer Academic Publishers, pp. 61-73, 2003.

[11] Fabian Kuhn and Roger Wattenhofer, “Constant-Time Distributed Dominating Set Approximation”, Twenty-Second ACM Symposium on Principles of Distributed Computing (PODC 2003), July 13-16, 2003, Boston

[12] D. J. Baker and A. Ephridemes, “The architectural organization of a packet radio network via a distributed algorithm”, IEEE Transactions on Communications, COM-29 (1981), pp. 1694-1701.

[13] I. Cidon and O. Mokryn, “Propagation and Leader Election in Multihop Broadcast Environment”, 12th International Symposium on Distributed Computing (DISC98), September 1998, Greece, pp.104–119.

[14] B. N. Clark, C. J. Colbourn, and D. S. Johnson, “Unit Disk Graphs”, Discrete Mathematics, 86:165–177, 1990.

[15] J. Wu and H. Li, “On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks”, Proc. of the Third International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, , Aug. 1999, 7-14.

[16] I. Stojmenovic, M. Seddigh and J. Zunic, “Dominating sets and neighbor elimination based broadcasting algorithms in wireless networks”, Proc. IEEE Hawaii Int. Conf. on System Sciences, January 2001.

[17] B. Chen et al., “Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks”, in Proc. Mobi-Com’01, July 2001.

[18] X. Cheng, X. Huang, D. Li, W. Wu and D.-Z.Du, “Polynomial-Time Approximation Scheme for Minimum Connected Dominating Set in Ad Hoc Wireless Networks”, Networks, Vol. 42, No. 4, pp. 202-208, 2003.

[19] Charles E. Perkins and Pravin Bhagwat. “Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers”, In Proceedings of the SIGCOMM ’94 Conference on Communications Architectures, Protocols and Applications, pages 234–244, August 1994.

[20] David B. Johnson and David A. Maltz. “Dynamic source routing in ad hoc wireless networks”, In Mobile Computing, edited by Tomasz Imielinski and Hank Korth, chapter 5, pages 153–181. Kluwer Academic Publishers, 1996.

[21] Charles Perkins. “Ad Hoc On Demand Distance Vector (AODV) routing”, Internet-Draft, draft-ietf-manet-aodv-00.txt, November 1997. Work in progress.

[22] M.V. Marathe, H. Breu, H.B. Hunt III, S.S. Ravi and D.J. Rosenkrantz, “Simple heuristics for unit disk graphs”, Networks, vol. 25, pp. 59–68, 1995.