

A Dominating Set Based Clustering Algorithm for Mobile Ad hoc Networks

Deniz Cokuslu, Kayhan Erciyes and Orhan Dagdeviren

Izmir Institute of Technology
Computer Eng. Dept., Urla, Izmir 35340, Turkey
{denizcokuslu, kayhanerciyes, orhandagdeviren}@iyte.edu.tr

Abstract. We propose a new Connected Dominating Set (CDS) based algorithm for clustering in Mobile Ad hoc Networks (MANETs). Our algorithm is based on Wu and Li's [14] algorithm, however we provide significant modifications by considering the degrees of the nodes during marking process and also provide further heuristics to determine the color of a node in the initial phase. We describe, analyze and measure performance of this new algorithm by simulation and show that it performs better than Wu and Li's [14] algorithm especially in the case of dense networks.

1 Introduction

MANETs do not have any fixed infrastructure and consist of wireless mobile nodes that perform various data communication tasks. MANETs have potential applications in rescue operations, mobile conferences, battlefield communications etc. Conserving energy is an important issue for MANETs as the nodes are powered by batteries only[1].

Clustering has become an important approach to manage MANETs. In large, dynamic ad hoc networks, it is very hard to construct an efficient network topology. By clustering the entire network, one can decrease the size of the problem into small sized clusters. Clustering has many advantages in mobile networks. Clustering makes the routing process easier, also, by clustering the network, one can build a virtual backbone which makes multicasting faster. However, the overhead of cluster formation and maintenance is not trivial. In a typical clustering scheme, the MANET is firstly partitioned into a number of clusters by a suitable distributed algorithm. A Cluster Head (CH) is then allocated for each cluster which will perform various tasks on behalf of the members of the cluster. The performance metrics of a clustering algorithm are the number of clusters and the count of the *neighbor nodes* which are the adjacent nodes between clusters that are formed [1].

In this study, we search various graph theoretic algorithms for clustering in MANETs and propose a new distributed algorithm. *Dominating Set based Clustering* which is a fundamental approach and related work in this area are reviewed in Section 2. We illustrate our algorithm and sample results in Section 3 and the Conclusion Section provides the overview.

2 Background

2.1 Clustering Using Dominating Sets

A dominating set is a subset S of a graph G such that every vertex in G is either in S or adjacent to a vertex in S [2]. Dominating sets are widely used in clustering networks[8]. Dominating sets can be classified into three main classes, Independent Dominating Sets (IDS), Weakly Connected Dominating Sets (WCDS) and Connected Dominating Sets (CDS)[4].

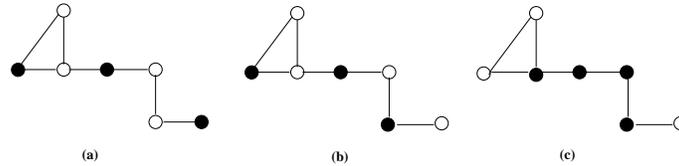


Fig. 1. (a)IDS (b)WCDS (b)CDS

- *Independent Dominating Sets*: IDS is a dominating set S of a graph G in which there are no adjacent vertices. Fig. 1.a shows a sample independent dominating set where black nodes show cluster heads.
- *Weakly Connected Dominating Sets (WCDS)*: A weakly induced subgraph $(S)_w$ is a subset S of a graph G that contains the vertices of S , their neighbors and all edges of the original graph G with at least one endpoint in S . A subset S is a weakly-connected dominating set, if S is dominating and $(S)_w$ is connected [5]. Black nodes in Fig. 1.b show a WCDS example.
- *Connected Dominating Sets*: A connected dominating set (CDS) is a subset S of a graph G such that S forms a dominating set and S is connected. Fig. 1.c shows a sample CDS.

2.2 Dominating Set Algorithms

Various algorithms exist for clustering in IDS, WCDS and CDS.

Clustering Using IDS: Baker and Ephremides [6] proposed an independent dominating set algorithm called *highest vertex ID*. A very similar algorithm to the highest id algorithm is the *lowest id algorithm* by Gerla and Tsai [7]. Gerla and Tsai developed another algorithm to find the independent dominating sets called the *highest degree algorithm*. Although these algorithms are considered as important algorithms, Chen et al. [8] proposed that these algorithms are not working correctly for some graphs. To solve this incorrect operation, Chen et al. developed the *k-distance independent dominating set* algorithm.[9].

Clustering Using WCDS: Although independent dominating sets are suitable for constructing optimum sized dominating sets, they have some deficiencies such as lack of direct communication between cluster heads. In order to obtain the connectivity between cluster heads, WCDSs can be used to construct clusters. The WCDS was first proposed for clustering in ad hoc networks by Chen and Liestman [10] called *zonal clustering*.

Clustering Using CDS: CDSs have many advantages in network applications such as ease of broadcasting and constructing virtual backbones [11], however, when we try to obtain a connected dominating set, we may have undesirable number of cluster heads. So, in constructing connected dominating sets, our primary problem is to find a minimal connected dominating set. Guha and Khuller [12] proposed two centralized greedy algorithms for finding suboptimal connected dominating sets. Das and Bharghavan [13] provided distributed implementations of Guha and Khuller's algorithms [12]. Wu and Li [14], improved Das and Bharghavan's distributed algorithm as a localized distributed algorithm for finding connected distributed sets in which each node only needs to know its distance-two neighbor [13]. Then Wu and Dai, proposed an extended version of this algorithm which uses more general rules in order to cluster graphs[18]. Xinfang Yan, Yugeng Sun, and Yanlin Wang [15] proposed a heuristic algorithm for minimum connected dominating set which uses uptime and power levels of the nodes as heuristics. Peng-Jun Wan, Khaled M. Alzoubi and Ophir Frieder [16] proposed a distributed algorithm for finding a CDS which constructs the dominating set using the Maximal Independent Sets. Hui Liu, Yi Pan and Jiannong Cao [17], improved Wu and Li's algorithm [14] by adding a third phase elimination. In the additional third phase, the algorithm searches redundant cluster heads. A cluster head is eliminated if it is dominated by two of its cluster head neighbors. Our algorithm is based on Wu and Li's CDS Algorithm [14].

Wu and Li CDS Algorithm Wu and Li CDS Algorithm [14] is a step wise operational distributed algorithm, in which every node has to wait for others in lock state in the algorithm. In this algorithm, initially each vertex marks itself *WHITE* indicating that it is not dominated yet. In the first phase, a vertex marks itself *BLACK* if any two of its neighbors are not connected to each other directly. In the second phase, a *BLACK* marked vertex v changes its color to *WHITE* if either of the following conditions is met:

1. $\exists u \in N(v)$ which is marked *BLACK* such that $N[v] \subseteq N[u]$ and $id(v) < id(u)$;
2. $\exists u, w \in N(v)$ which is marked *BLACK* such that $N(v) \subseteq N(u) \cup N(w)$ and $id(v) = \min\{id(v), id(u), id(w)\}$;

3 The Distributed Dominating Set Based Clustering Algorithm

3.1 General Idea

We propose a distributed algorithm which finds a minimal connected dominating set in a MANET. We developed our algorithm based on Wu's CDS Algorithm but we add some extra heuristics. First, we determine some situations that a node cannot change its color after the first phase. We also consider the degree of a node when marking it.

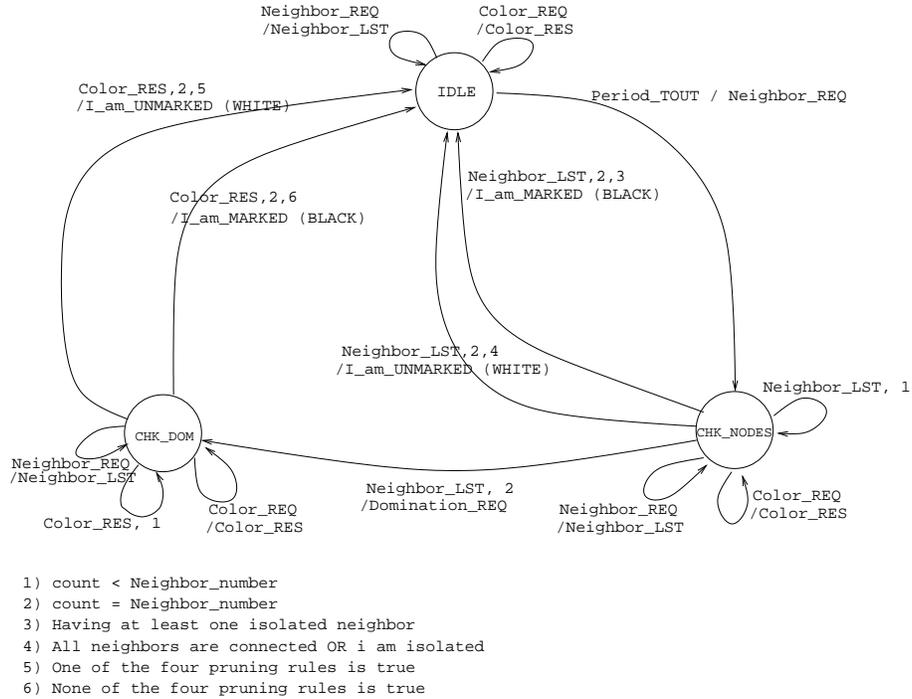


Fig. 2. Finite State Machine of the Clustering Algorithm

3.2 Algorithm

We assume that each node has a unique *node_id* and knows its adjacent neighbors. Each node has a *color* indicating whether the node is in the dominating set or not. The *color* is set to *BLACK* if the node is in the dominating set, or *WHITE* if the node is not in the dominating set. Color *GRAY* is used to indicate that the node is marked after the first phase, but it will change its color after the

second phase as either *WHITE* or *BLACK*. The finite state diagram for the algorithm can be seen in Fig. 2. *Period_TOUT* message triggers the algorithm and is sent periodically. *Neighbor_REQ* message is sent to collect a list of distance-2 neighbors. *Neighbor_LST* message includes a list of adjacent neighbors of sending node. *Color_REQ* message is used to collect a node's neighbor's colors after the first phase. *Color_RES* message includes sender's color after the first phase. Each node is in the *IDLE* state and colored as *UNDEFINED_COLOR* initially. When all *Neighbor_LST* messages are collected in the *CHK_NODES* state, the node checks the following heuristics to determine if it will be among the ones whose color will not alter after the first phase:

- If the node has at least one isolated neighbor, it changes its color to *BLACK* and its state to *IDLE*.
- If all neighbors of the node are directly connected to each other or if the node is an isolated node, it changes its color to *WHITE* and its state to *IDLE*.

If the node is not suitable for one of these two coloring heuristics, then it changes its color to *GRAY* and its state to *CHK_DOM*. When the node switches to state *CHK_DOM*, it multicasts a *Color_REQ* message to its neighbors. Then it waits until all its neighbors send their colors. When the node v collects all color information, it starts to apply the following rules:

1. $\exists u \in N(v)$ which is marked *BLACK* such that $N[v] \subseteq N[u]$;
2. $\exists u, w \in N(v)$ which is marked *BLACK* such that $N(v) \subseteq N(u) \cup N(w)$;
3. $\exists u \in N(v)$ which is marked *GRAY* such that $N[v] \subseteq N[u]$ and $degree(v) < degree(u)$ OR $(degree(v) = degree(u) \text{ AND } id(v) < id(u))$;
4. $\exists u, w \in N(v)$ which is marked *GRAY* OR *BLACK* such that $N(v) \subseteq N(u) \cup N(w)$ and $degree(v) < \min\{degree(u), degree(w)\}$ OR $degree(v) = \min\{degree(u), degree(w)\} \text{ AND } id(v) < \min\{id(u), id(w)\}$;

If one of these rules is true, then the node v changes its color to *WHITE*, else it changes its color to *BLACK*. After the node determines its permanent color, it changes its state to *IDLE*. At any state, a node can receive request messages to help other nodes run their algorithms. These messages are *Neighbor_REQ* and *Color_REQ*. In such a case, the node prepares the required information requested in the received message and continues to its current operation. No state changes are performed in these cases.

3.3 An Example Operation

We obtained the resulting connected dominating set in Fig. 3 by using our algorithm. This section explains the algorithm step by step by using the sample graph in Fig. 3. Runtime of the algorithm is explained for nodes 1 and 8, but we remark again that each node is running the algorithm concurrently.

- *Execution of algorithm at node 1:* When node 1 times out the period, it sets its color to *UNDEFINED_COLOR*, sends a *Neighbor_REQ* message to all of

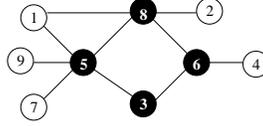


Fig. 3. Example

its neighbors and changes its state to *CHK_NODES*. It waits at that state until all response messages are collected from its neighbors. Once responses are collected, node 1 prepares a list of its directly unconnected neighbors by looking at *Neighbor_LST* messages. In this stage, it finds out that all of its neighbors are directly connected, therefore the node 1 sets its color to *WHITE* and its state to *IDLE*.

- *Execution of algorithm at node 8:* When node 8 times out the period, it sets its *color* to *UNDEFINED_COLOR*, sends a *Neighbor_REQ* message to all of its neighbors and changes its state to *CHK_NODES*. Once responses are collected, node 8 checks if one of the heuristics is suitable for it, at that stage, it finds out that the node 8 has an isolated neighbor (*node2*), so it sets its color to *BLACK* and its state to *IDLE*.

3.4 Analysis

Theorem 1. *Time complexity of the clustering algorithm is $\Theta(4)$.*

Proof. Every node executes the distributed algorithm by the exchange of 4 messages. Since all these communication occurs concurrently, at the end of this phase, the members of the CDS are determined, so the time complexity of the algorithm is $\Theta(4)$.

Theorem 2. *Message complexity of the clustering algorithm is $O(n^2)$ where n is the number of nodes in the graph.*

Proof. For every mark operation of a node, 4 messages are required (*Neighbor_REQ*, *Neighbor_LST*, *Color_REQ*, *Color_RES*). Assuming every node has $n-1$ adjacent neighbors, total number of messages sent is $4(n-1)$. Since there are n nodes, total number of messages in the system is $n(4(n-1))$. Therefore messaging complexity of our algorithm has an upperbound of $O(n^2)$.

3.5 Results

We implemented the Dominating Set Based Clustering Algorithm with POSIX threads in RedHat. Each thread is treated as a mobile node. Graph is constructed randomly with different densities. Fig. 4 displays the cluster head numbers at graphs with 40, 50, 60 and 70 per cent densities. The results are taken at ranging from 10 to 90 nodes in a random graph. It can be seen that our modified algorithm performs significantly better than Wu and Li's[14] algorithm at higher densities.

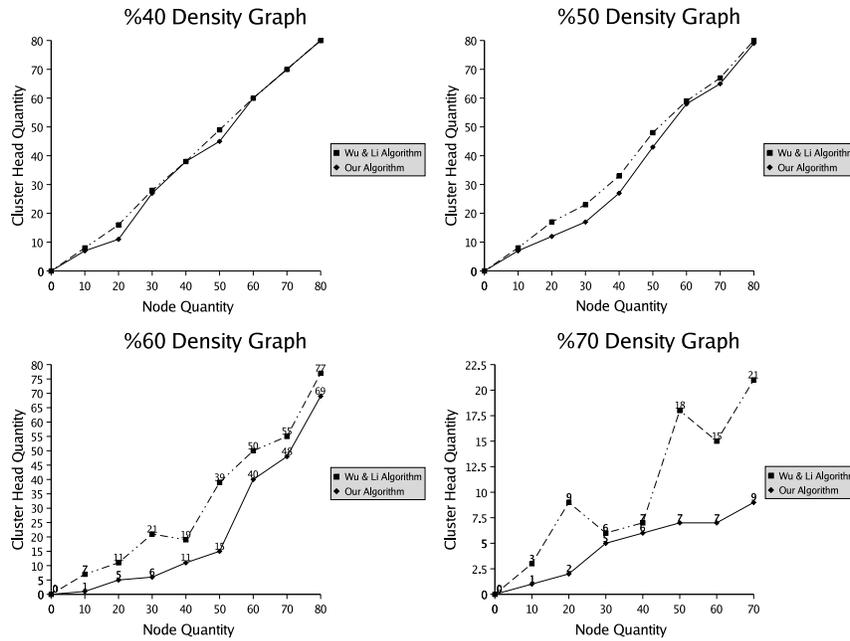


Fig. 4. Test Results

4 Conclusions

In this paper, we proposed a distributed algorithm with significant modifications on Wu and Li's algorithm for constructing a CDS. We showed the implementation of the algorithm and analyzed its time and message complexities. We showed that we can decrease the size of the CDS by adding some heuristics to Wu and Li's algorithm. We also showed that this improvement can be significant especially when the number of nodes in the MANET is large. Therefore, we can conclude that the algorithm can be preferable in dense mobile networks. Clustering described in this study can be used for various distributed tasks in MANET's such as routing and multicast communications. We are planning to experiment various *total order multicast* algorithms in such an environment where message ordering is provided by the cluster heads on behalf of the ordinary nodes of the MANET.

References

1. Nocetti, F., B., Gonzalez, J., S. and Stojmenovic, I. : Connectivity Based K-Hop Clustering in Wireless Networks, Telecommunication Systems, (2003), 205-220.
2. West, D. : Introduction to Graph Theory, Second edition, Prentice Hall, Upper Saddle River, N.J., (2001).

3. Chen, Y., Z., P. and Liestman, A., L. : Approximating Minimum Size Weakly-Connected Dominating Sets for Clustering Mobile Ad Hoc Networks, Proc. of 3'rd ACM Int l. Symp. Mobile Ad Hoc Net. and Comp., (2002), 16572.
4. Haynes, T., W., Hedetniemi, S., T. and Slater, P., J. : Domination in Graphs, Advanced Topics, Marcel Dekker, Inc., (1998).
5. Chen, Y., Z., P., Liestman, A., L. and Jiangchuan, L. : Clustering Algorithms for Ad Hoc Wireless Networks, Nova Science Publishers, (2004).
6. Baker, D. and Ephremides, A. : The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm, Communications, IEEE Transactions, (1981), 29(11), 1694-1701.
7. Gerla, M. and Jack T., C., T. : Multicluster, Mobile, Multimedia Radio Network, Wireless Networks, 1(3), (1995), 255-265.
8. Chen, G., Nocetti, F.,G., Gonzalez and J.S., Stojmenovic, I. : Connectivity Based K-Hop Clustering in Wireless Networks, System Sciences, Proc. of the 35th Annual Hawaii International Conference, (2002), 2450-2459.
9. Ohta, T., Inoue, S. and Kakuda, Y. : An Adaptive Multihop Clustering Scheme for Highly Mobile Ad Hoc Networks, Proc. of 6th ISADS'03, (2003).
10. Chen, Y., P. and Liestman, A., L. : A Zonal Algorithm for Clustering Ad Hoc Networks, International Journal of Foundations of Computer Science, (2003), 14(2), 305-322.
11. Stojmenovic, I., Seddigh M. and Zunic, J. : Dominating Sets and Neighbor Elimination-Based Broadcasting Algorithms in Wireless Networks, IEEE Transactions on Parallel and Distributed Systems, (2002), 13, 14-25.
12. Guha S. and Khuller, S. : Approximation Algorithms for Connected Dominating Sets, Springer Verlag New York, LLC, ISSN: 0178-4617, (1998).
13. Das, B. and Bharghavan, V. : Routing in Ad-Hoc Networks Using Minimum Connected Dominating Sets, Communications, ICC97 Montreal, 'Towards the Knowledge Millennium', IEEE International Conference, (1997), 1, 376-380.
14. Wu, J. and Li, H. : A Dominating-Set-Based Routing Scheme in Ad Hoc Wireless Networks, Springer Science+Business Media B.V., Formerly Kluwer Academic Publishers B.V. ISSN: 1018-4864, (2001).
15. Yan, X., Sun, Y. and Wang, Y. : A Heuristic Algorithm for Minimum Connected Dominating Set with Maximal Weight in Ad Hoc Networks, Proc. of the Grid and Cooperative Computing Second International Workshop, (2003), 719-722.
16. Wan, P., J., Alzoubi, K., M. and Frieder, O. : Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks, Springer Science+Business Media B.V., Formerly Kluwer Academic Publishers B.V., , (2002), 9(2), 141-149.
17. Liu, H., Pan, Y. and Cao, J. : An Improved Distributed Algorithm for Connected Dominating Sets in Wireless Ad Hoc Networks, Parallel and Distributed Processing and Applications, Proc. of the ISPA 2004, (2004), 340.
18. Wu, J. and Dai, F. : An Extended Localized Algorithm for Connected Dominating Set Formation in Ad-Hoc Wireless Networks, IEEE Transactions on Parallel and Distributed Systems, (2004), 15(10).