A DISTRIBUTED APPROACH FOR COMPUTING THE MINIMUM CONNECTED DOMINATING SET IN AD HOC NETWORKS

Kais Mnif, Bo Rong and Michel Kadoch

LAGRIT, Department of Electrical Engineer
École de Technologie Supérieure, Université de Québec
Montréal, Canada H3C 1K3

Abstract-

In Ad-hoc networks, an optimized way of flooding packets is to find the Minimum Connected Dominating Set (MCDS). Nodes belonging to the MCDS set are responsible for relaying messages, while other nodes are not. The problem of finding a small size MCDS is known to be NP-hard. Most of previous methods, such as CDS-based and Weakly-CDS, have followed combinatorial approach or graph coloration technique to find an approximate solution. However, these methods are centralized. In order to give a distributed solution with less computing complexity, this paper proposes a new approach of employing integer linear program. This approach offers a two-step scheme to solve the small size MCDS problem. In the first step, the size of the dominating set is minimized with the constraint that each node in the network needs to be either in this set or adjacent to at least one dominating set node. The aim of this step is to find the Minimum Dominating Set (MDS), where the elements are not connected yet. The second step finds the spanning tree for the MDS set to get all of its elements connected. To evaluate the performance of our approach, we compute the size of MCDS in a variety of graphs, Simulation results show that our approach has a very good performance in parameters such as size of MCDS and computing complexity compared with CDS-B and WCDS approaches.

Keywords: Minimum Connected dominating Set, Integer programming, Ad hoc network

1. INTRODUCTION

In the next generation of wireless communication systems, there will be a need for the rapid deployment of independent mobile users. Significant examples include establishing survivable, efficient, dynamic communication for emergency/rescue operations, disaster relief efforts, and military networks [1]. An ad hoc network is an infrastructure-less network composed of mobile nodes. Nodes communicate over relatively bandwidth constrained wireless links. Since the nodes are mobile, the network topology may change rapidly and unpredictably over time. The network is distributed, where all network activity including discovering the topology and delivering messages must be executed by the nodes themselves, i.e., routing functionality will be incorporated into mobile nodes.

The routing process can be restricted to the reduced graph formed by the minimum connected dominating set, every node from the MCDS will be considered as a gateway, and only gateway hosts need to keep routing information. However, an important aspect is how to select an optimal set of gateways based on the information of one-hope neighbours. Most protocols use some form of Hello messages, which are sent periodically by each node to inform its presence.

The connected dominating set problem can be defined as follows: find a minimum size subset *S* of nodes, such that the sub-graph induced by *S* is connected and *S* forms a dominating set [2]. This problem is shown to be NP-hard [3].

The remainder of this paper is organized as follows. Section 2 presents some background information and previous works regarding the computation of the MCDS. Section 3 overviews the proposed solution. In section 4 we present an evaluation of the proposed solution and compare it with other approaches. Last section will conclude this paper and gives a direction for the future work.

2. BACKGROUND

In ad hoc network, the MCDS is very useful especially; each node from the MCDS set will be considered as a multipoint relay. The technique of multipoint relays provides an adequate solution to reduce flooding of broadcast messages in the network. Figure 1 shows an example where a broadcast message is diffused in the network using a) pure flooding and b) the multipoint relays. It's clear that using multipoint relay's technique reduces the number of redundant re-transmissions while diffusing a broadcast message in the network and in this example four nodes are enough to retransmit messages. The smallest the multipoint relay sets are, the fewer retransmissions will occur. It is NP-hard to compute a multipoint relay set with minimum size.

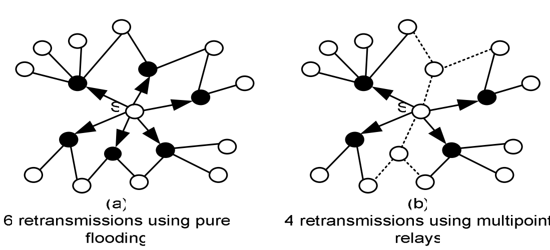

Fig. 1- Diffusion of broadcast message using a- pure flooding and b- multipoint relays in ad hoc network.

The existing heuristics are based on preferring neighbours with large degree as multipoint relays [4].

In ad hoc wireless network, some links may be unidirectional due to the hidden and exposed terminal problem. the hidden terminal problem refers to the collision of packets at a receiving node due to the simultaneous transmissions of those nodes that are not within the direct transmission range of the sender, but are within the transmission range of the receiver [5]. The exposed terminal problem refers to the inability of a node, which is blocked due to transmission by a nearly transmitting node, to transmit to another node. The hidden and exposed problem significantly reduces the throughput of a network when the traffic load is high.

2.1. Previous work

There many existing heuristics for computing a CDS [2, 6-10]. Some of them put effort to minimize the size but some did not. All of them use either degree or id as metric and they combinatorial approach or graph coloration technique to compute the CDS. In [2], Guha and Khuller proposed an approximation algorithms for connected dominating sets (CDS-based). The limitations of the last algorithm are: first it can be apply only to wireless ad hoc networks whose unit-disk graph is not a complete graph. Second, Wan *et al.* showed in [6] that the approximation factor of this algorithm is at most $n/2$ where n is the number of nodes, which is a very poorly performance.

These approaches are centralized and their efficiency decrease when the size of network grow. We propose an integer programming formulation to calculate the minimum connected dominating set MCDS. This formulation gives us the opportunity to divide the problem into two steps. The first step finds the MDS set and the second step computes the final solution of the MCDS using a distributed algorithm to find the spanning tree of the MDS set.

3. A DISTRIBUTED APPROACH TO COMPUTE THE MCDS

Our approach to compute the minimum connected dominating set can be divided into two steps. In the first step, the size of the dominating set is minimized with the constraint that each node in the network needs to be either in this set or adjacent to at least one dominating set node. The aim of this step is to find the Minimum Dominating Set (MDS). In general, we have no warranty about the connectivity of the MDS set; nodes in the MDS are not connected. The second step finds the spanning tree for the MDS set to get the final solution MCDS. To get a good approximation of the final solution, the first step can be solved using linear programming theory and get an optimal solution for the MDS set. The second step is to find the spanning tree induced by the MDS set, there were efficient algorithms that can be applied to get a good approximation (Prim or Kruskal). Figure 2 shows an example of the MCDS set. Black nodes represent the MDS set. Gray nodes represents nodes added to the MDS set to get a connected dominating set.

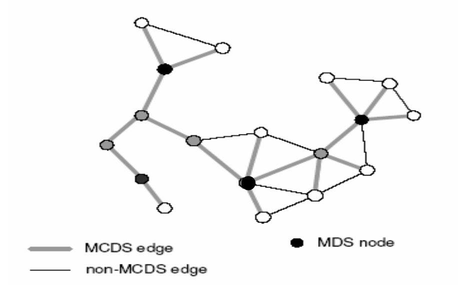

Fig. 2- A minimum connected dominating set

3.1 Dominating Set approach

In the first step, we give a formulation to find the small size of the dominating set. This formulation uses the integer linear programming theory.

We define $x(i)$ a decision variable,

$$x(i) = \begin{cases} 1 \text{ if the node } i \text{ is an element the dominating set, MDS} \\ 0 \text{ otherwhise} \end{cases}$$

The objective function minimizes the number of node of the dominating set:

$$\min \sum_{i \in V} x(i) \quad (1)$$

Domination constraint:

$$X + M \times X \geq 1 \quad (2)$$

$$\text{Avec } x(i) \in \{0,1\} \quad (3)$$

Where $X = [x(1),\ldots,x(n)]^t$ represent the decision vector, and M is $n \times n$ 0/1 adjacent matrix of G, where $m_{ij} = 1$ if node i is connected to node j.

With linear programming formulation, we are sure that we get an optimal solution of the minimum dominating set in $O(n)$ running time. According to [12], a linear programming problem with d variables and n constraints can be solved in $O(n)$ time as n tends to infinity.

3.2. Connected dominating set approach

In this section we give a general formulation of the problem of the minimum spanning tree (MST), this formulation can be found in any graph theory books. We keep a formulation given by Ahuja *et al.* [13]. Let $A(S)$ denote the set of arcs contained in the sub graph of $G = (V, A)$ induced by the node set S [i.e., $A(S)$ is the set of arcs of A with both endpoints in S]. The following integer programming formulation expresses the minimum tree problem:

$$\min \sum_{(i,j) \in A} c_{(i,j)} y_{(i,j)} \quad (4)$$

Subject to

$$\sum_{(i,j) \in A} y_{(i,j)} = n_S - 1 \quad (5)$$

$$\sum_{(i,j)\in A(S)} y_{(i,j)} \le |S|-1 \quad \text{for any set } S \text{ of nodes}, \qquad (6)$$

$$\text{with } y_{(i,j)} = \{0,1\} \qquad (7)$$

In this formulation, the 0-1 variable $y_{(i,j)}$ indicates whether we select arc (*i,j*) as part of the chosen Spanning tree, S_T. The constraint (5) is a cardinality constraint implying that we choose exactly n_S-1 arcs, where $n_S = |T_S|$, and the "*packing*" constraints (6) implies that the set of chosen arcs contain no cycles (if the chosen solution contained a cycle, and *S* were the set of nodes on a chosen cycle, the solution would violate this constraint). Note that as a function of the number of nodes in the network, this model contains an exponential number of constraints. Nevertheless, there exist efficient distributed algorithms for this kind of problem, we can apply *Prim's* algorithm for example. The running time of *Prim's* algorithm is $O(m+n.log\ n)$ [13], where *n* and *m* denote, respectively, the number of nodes and arcs in the graph. Better than that, the running time can be reduced to $O(m.log\ n)$ if *Sollin's* algorithm is used. *Sollin's* algorithm is better than *Prim's* algorithm for sparse networks, and is worse for dense network [14].

3.3. Proof of correctness

Let $G = (V, E)$, as defined in section I.2. If *S* is a dominating set of *G* such that the sub graph of *G* induced by the nodes in *S* is connected, then *S* is called a connected dominating set of *G*. The problem of finding a connected dominating set with minimum size is NP-hard.

Lemma

Let $G = (V, E)$, $S \subset V$ is connected dominating set of G if and only if there exist a spanning tree T of G such that V-S is a subset of leaves of T.

Proof: Let $S \subset V$ is connected dominating set of *G*. Let T_S a spanning tree of the sub graph of *G* induced by nodes in *S*. For each node $w \in V - S$, we choose a node $u_w \in S \cap N(w)$, where $N(v) = \{u\in V \mid \text{edge } (u,v) \in E\}$. We can see easily that $T = T_S \cup \{u_w \mid w \in V\text{-}S\}$ is a spanning tree of *G* and *U* is a subset of leaves of *T*, then every node in *U* is attached to at least one node in *V-U* and the sub graph of *G* induced by nodes in *V-U* is connected. Therefore *V-U* is a connected set of *G*. ■

4. PRFORMANCE EVALUATION

4.1. Graph generation

In this section, we perform the simulation which compares the average size of the dominating set generated from our approach with the two approaches presented in section I. Random graphs are generated in a 10x10 square units of a 2-D simulation area, a randomly number of nodes are generated and placed on this area. Two nodes are connected by a link if and only if they are sufficiently near; which represents in ad hoc environment that node is in the propagation range of the other node.

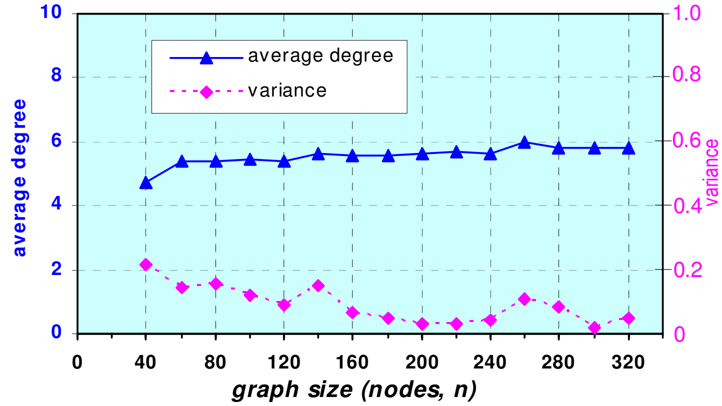

Fig. 3- Average degree and variance versus number of node in a graph

To reduce complexity, we suppose links are bidirectional. For each generated graph, we examine the degree for each node, we only take graph with non-null degree; otherwise, it is discarded. Figure 6 displays the average degree (blue) in the generated graph versus graph size and its variance (magenta). This results shows that generated graphs have almost a constant degree and a very small variance (< 0.2).

These results shows that using our approach, the randomly graph generated are sufficiently connected and dense. Figure 3 shows an example of the generated graph for *n*=60. In this example the number of edges *m*= 143 and the average degree of this graph *deg* = 4.75. The MDS set includes nodes 11, 12, 14, 15, 20, 21, 24, 25, 26, 27 and the MCDS set includes nodes from the MDS set plus nodes 3, 5, 6, 7, 23, 30, 33, 34, 40, 49, 51, 54, 58 and 60.

4.2. Simulation Results

We compare the three approaches by varying the number of node, *n,* on a graph. For each value of *n*, we compute the MCDS. Several runs of each *n* are conducted to obtain statistically confident average. For each randomly generated graph, we measure the average size of the MCDS resulting from our approach (Line. Prog.), Wi and Li approach's (WCDS) and Guha and Khuller approach's (CDS-based) Simulation shows that our approach gives an average size of the MCDS less then given by other approaches, figure 5.

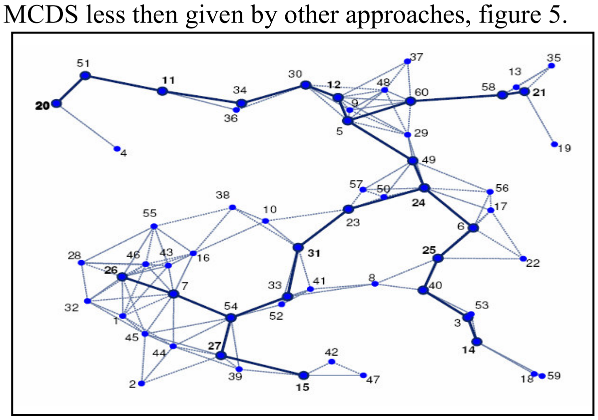
Fig. 4- Random graph generated for *n* = 60

The difference grows to 10% (for n=320) which is very significantly. These results are justified by the fact that our approach chooses the nodes having a maximum degree and this is illustrated by the figure 6. Choosing nodes with high degree reduce tree depth which improves the routing process in ad hoc network.

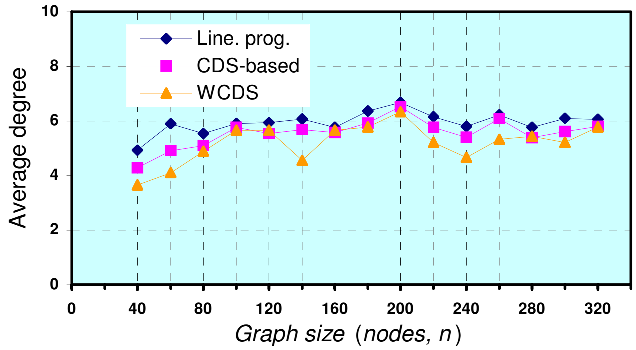

Fig. 5 - Average MCDS size versus number of node in a graph.

5. CONCLUSION

In this paper, we discussed how to find the Minimum Domination Connecting Set problem using the elementary integer programming formulation. Simulations show that this formulation gives an optimal solution compared to two heuristics approaches. We showed that the problem could be decomposed into two sub-problem which an be solved separately. The performance comparison of these three algorithms is listed in Table 1. We can conclude from this table that our approach outperforms the existing approach in time and message complexity. Also our approach needs only a single neighbour knowledge. Due to the dynamic nature of ad hoc networks, it's desirable to be able to maintain the group information at a reasonably low cost. In future, we will address the issues coming from the change of the MCDS structure due to subscriber unit's moves.

Acknowledgements

This work was supported in grants by Bell and NSERC

References

[1] NIST, "Mobile Ad Hoc Networks (MANETs) http://www.antd.nist.gov/wahn_mahn.shtml."

[2] S. Guha and S. Khuller, "Approximation Algorithms for Connected Dominating Sets" *Proceedings of the Fourth Annual European Symposium on Algorithms*, Springer-Verlag London, UK, pp. 179 -193, 1996.

[3] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint relaying for flooding broadcast messages in mobile wireless networks" *in Proc. of the 35th Annual Hawaii International*, pp. 3866-3875, 2002.

[4] C. Adjih, P. Jacquet, and L. Viennot, "Computing Connected Dominated Sets with Multipoint relays" *INRIA- RR n^o 4597*, 17 pages, 2002.

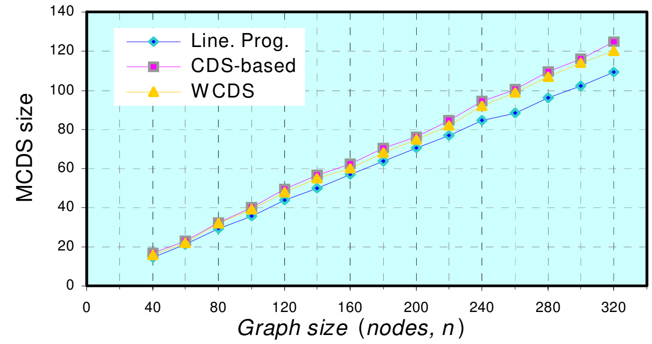

Fig. 6 - Average degree versus number of node in a graph.

Table 1: Performance comparison

	WCDS[11]	CDS-based[2]	Line. prog.
Upper bound approximation	(2+ln Δ)|*OPT*|	2(1+H(Δ)).|*OPT*|	|*OPT*|
Time complexity	(ln Δ+1)|*OPT*|	$O(n^2)$	$O(m+n\log n)$
Message complexity	$O(n^2)$	$O(n)$ - $O(n^2)$	$O(n \log n)$
Neighbour knowledge	*single*	*two*	*single*

n: number of nodes, H: a harmonic function,
m: number of edges, Δ: maximum degree in the graph,
|*OPT*|: the size of an optimal connected dominating set.

[6] P.-J. Wan, K. M. Alzoubi, and O. Frieder, "Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks" *Infocom'02*, 2002.

[7] B. Das and V. Bharghavan, "Routing in Ad-Hoc Networks Using Minimum Connected Dominating Sets" *In International*

[8] S. Guha and S. Khuller, "Approximation Algorithms for Connected Dominating Sets" *Algorithmica*, vol. 20, pp. 374-387, 1998.

[9] K. M. Alzoubi, P.-J. Wan, and O. Frieder, "New distributed algorithm for connected dominating set in wireless ad hoc networks" *in Proc. HICSS, Hawaii,* pp. 3881-3887, *2002*.

[10] J. Wu and H. Li, "A Dominating-Set-Based Routing Scheme in Ad Hoc Wireless Networks" *Telecommunication Systems*, 1999.

[11] J. Wu and H. Li, "On Calculating Connected Dominating Set for Efficient Routing in Ad hoc Wireless Networks" in *Proc. of the 3rd Inter. Workshop on Discrete Algorithms and Methods for MOBILE Computing and Communications*. Seattle, WA, USA, pp. 7-14, 1999.

[12] N. Megiddo, "On the complexity of linear programming" in *Advances in economic theory: Fifth world congress. T. Bewley*. ed. Cambridge University Press, pp. 225–268, 1987.

[13] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. New Jersey: Prentice-Hall, Inc., 1993.

[14] J. P. Jarvis and D. E. White, "Computational experience with minimum spanning tree algorithms" *Operations Research Letters*, vol. 2, pp. 36-41, 1983.