An Approximation Algorithm for Connected Dominating Set in Ad Hoc Networks

Xiuzhen Cheng, Min Ding
Department of Computer Science
The George Washington University
Washington, DC 20052, USA
{cheng,minding}@gwu.edu

Dechang Chen
Uniformed Services University
of the Health Sciences
Bethesda, MD 20817, USA
dchen@usuhs.mil

Abstract

The construction of a virtual backbone for ad hoc networks is modelled by connected dominating set (CDS) in unit-disk graphs. This paper introduces a novel idea to compute CDS effectively - our algorithm does not include the members of an existing maximal independent set (MIS) even though it does connect the MIS. Popular approaches in literature count the MIS in the resultant CDS. Therefore it is possible that our new algorithm has better performance ratio.

1 Introduction

Ad hoc wireless networks are characterized by dynamic topology, multihop communication, and limited resources. These features make routing a challenging problem. Most existing routing protocols [11][12][16][17] rely on flooding for route discovery or topology update. However, flooding is unreliable [18], and can result in excessive redundancy, contention, and collision, the notorious broadcast storm problem [15].

Recently an approach based on overlaying a virtual backbone on an ad hoc network is proposed in [18]. The idea is to route control packets along backbone nodes to decrease protocol overhead. The virtual backbone is not only good for routing, but also a good structure for multicast/broadcast in ad hoc networks [21][22]. Other applications of virtual backbone include route selection for multimedia traffic [19], mobility management [13], etc.

In this paper, we will study the construction of a virtual backbone, which is modelled by computing a minimum connected dominating set (MCDS) in unit-disk graphs. This problem formation is justified as follows:

1. All transceivers have the same communication range, thus the footprint of an ad hoc wireless network is a unit-disk graph.
2. Since the virtual backbone is used to mainly disseminate control signals, it should involve as less number of nodes as possible.
3. All non-backbone nodes should be adjacent to at least one backbone node to exchange control message.
4. The induced topology of all virtual backbone nodes should be connected.

However, computing an MCDS is NP-hard in both general [9] and unit-disk graphs [6]. No exact efficient algorithm is available. In this paper, we will propose a novel idea to effectively generate CDS for unit-disk graphs. Compared to most existing works that rely on maximal independent set [1, 2, 4, 8, 20], our algorithm excludes the MIS in the resultant CDS. Therefore it is possible that our new algorithm has better performance ratio.

This paper is organized as follows. Section 2 briefly summarizes related works. Section 3 introduces basic concepts needed in this paper. We propose our algorithm in Section 4. The conclusion is given in Section 5.

2 Related Work

Modelling virtual backbone construction in ad hoc networks with MCDS computation in unit-disk graphs is a popular approach [1, 2, 4, 8, 7, 20, 22]. We can generally divide these heuristics into two categories: *leader-election based* and *pruning*. Leader-election based approach relies on the connection of an maximal independent set (MIS), with both the connectors and the members of the MIS forming a CDS. Pruning approach removes redundant nodes from the candidate set that contains all nodes initially. The proposed work in this paper does not belong to any of these two categories. Its basic idea is to connect any MIS with a subset of nodes in a greedy fashion, and then extend this subset to cover all other nodes.

The pioneer work that introduces the first leader-election based algorithm for MCDS in unit-disk graphs is [2]. The major idea of this algorithm and its variations [8, 20] is to compute and connect an MIS. The computation and connection of an MIS can be separated explicitly as in [2] and [20], or unified together to grow a dominating tree [8]. In unit-disk graphs, the performance ratio of these algorithms are either 8 or 9, depending on how the MIS can be connected. An important feature of these algorithms is simplicity. However, their time complexities are dominated by the leader-election, which usually takes $O(n\ logn)$-time. Multi leader based algorithms [1, 4] avoid leader-election, since nodes with highest ID among their neighbors can serve as leaders. These algorithms have linear time-complexity but their performance ratio is much higher, due to the connection of multiple sub-MISs whose generations are based on different leaders. An interesting work that connects an MIS with Steiner tree is proposed in [14]. This algorithm has performance ratio 6.8, the best reported in literature so far.

The CDS construction algorithm proposed in [22] and [7] falls into the second category. Nodes whose independent degree (the maximum number of independent neighbors) is 1, or whose neighbors are covered by connected nodes with higher IDs in the CDS, are removed from the CDS. This algorithm is localized. Unfortunately its performance ratio is $O(n)$ for general graphs [20]. Ref. [3] proposes a centralized approach to prune the original CDS set. The performance for this algorithm is unavailable.

MCDS in unit-disk graphs has PTAS [5], which means that it can be approximated to any degree. However, the best published result has performance ratio 6.8 [14]. Thus there is a big gap between the theoretically achievable result and the already achieved result. The algorithm proposed in this paper is an effort to decrease the gap. We conjecture that our heuristic has performance ratio at most $ln\ 4 + 4$.

3 Preliminaries

Given a graph $G = (V, E)$, G is a unit-disk graph if it satisfies the following condition: $\forall u, v \in V, (u, v) \in E$ if and only if $distance(u, v) \leq 1$. Two nodes are *independent* if they are not neighboring to each other. $S \subset V$ is an *independent set* of G if for $\forall u, v \in S$, u and v are independent. S is maximal (MIS) if adding any node into S breaks the independence property. A *dominating set* D of G is a subset of V such that any node not in D has at least one neighbor in D. If the induced graph of D is connected, then D is called a *connected dominating set (CDS)*. Among all connected dominating sets of G, the one with minimum cardinality is the *minimum connected dominating set (MCDS)*. From these definitions, it is clear that a maximal independent set is also a dominating set. Ref. [20] has proved the following result that relates the size of any MIS of a unit-disk graph G to that of its MCDS.

Lemma 3.1 *Let S be any maximal independent set and D be any MCDS of a unit-disk graph G. Then $|S| \leq 4 \cdot |D| + 1$ for $|D| > 1$.*

4 A Greedy Approximation Algorithm

In this section, we are going to propose a heuristic algorithm to compute a CDS for unit-disk graphs. We assume each node has unique id. Our algorithm contains four phases. The first phase computes an MIS, denoted by red nodes. All other nodes not in the CDS are colored white. The second phase connects the MIS in a greedy fashion by black nodes. The third phase extends the current connected black components to cover all white nodes. And the forth phase unifies all black components into one.

Phase I: Compute any maximal independent set (MIS) S. Color all nodes in S red. Colore all other nodes white.

Refs. [4] and [20] both imply heuristics to compute maximal independent set. Here we propose a simple idea based on [4]. Assume all nodes are white initially. In the first step, color the node with smallest id black. Color all its one-hop neighbors gray; color its two-hop neighbors yellow. At each step, choose a yellow node with smallest id among all its yellow neighbors and color it black. Color its one-hop white/yellow neighbors gray; color its two-hop white neighbors yellow. Repeat this procedure until all nodes are colored either black or gray. All black nodes form an MIS.

After Phase I, a node is either white or red. Phase II is expected to connect members of S in a greedy fashion. For better elaboration, let's define a *piece* to be either a black component, or a red node. A *neighboring piece* of node v is a piece that contains at least one direct (one-hop) neighbor of v. The *weight* of a node is defined to be the number of neighboring pieces minus 1. Each step in Phase II will pick up a node to color black that gives the maximum (non-zero) reduction in the number of pieces.

Phase II: Color the node with maximum none-zero weight (break ties by degree then id) black; color all its non-black neighbors gray; Update the weight of each node accordingly. Repeat this procedure until no node with positive weight exists. If there are red nodes left, color each black and color its white neighbors gray.

Note that at each step in Phase II, the greedy choice can be a white node, a gray node, or a red node. Phase II ends when no red node left. Let G' be the induced graph of all white nodes after Phase II. G' may consist of multiple components, as shown in Fig. 1 (b). Phase III will extend black components to cover all nodes in G'.

Phase III: Extends the black components generated in Phase II to cover G'.

It is easier to find a heuristic to cover the white nodes with existing black components. The requirement of this phase is to add as less black nodes as possible. In the following, we sketch a simple procedure applied to each component in G'.

1. Let u be the node with largest degree (break ties by id) in G'. Color u black. Color all its neighbors gray.

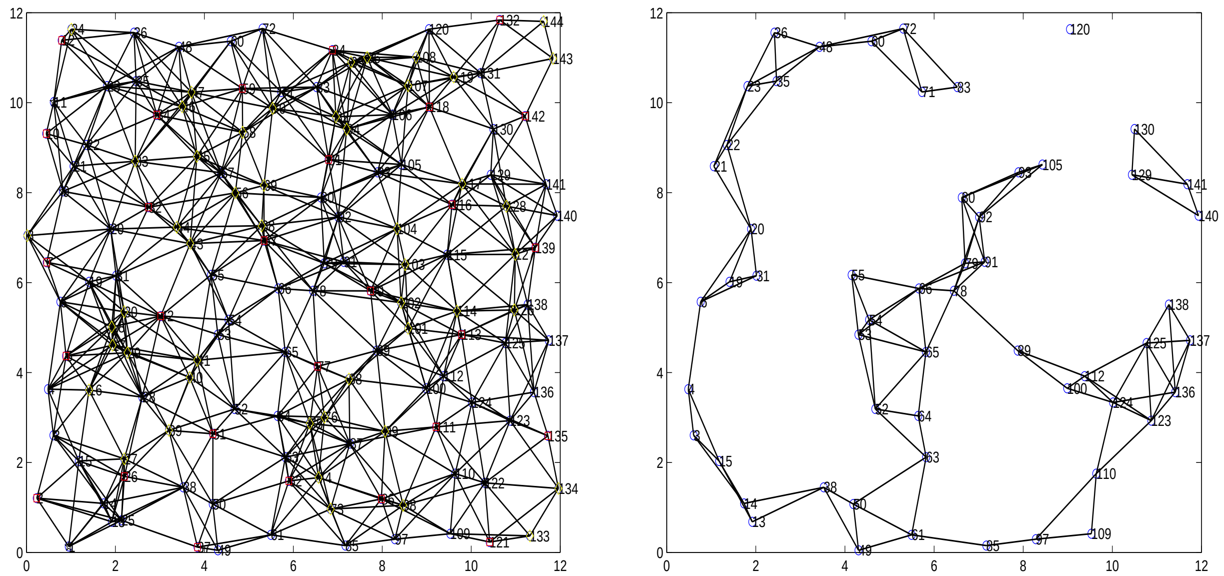
2. Define the *yield* of a gray node as the number of white neighbors. Color the gray node with highest yield (break ties by id) black and then color all its white neighbors gray. Repeat this until no white node left.

3. Consider the maximal independent set S computed in Phase I in the original graph G. Let v be the neighbor of u in S. Color v black.

Note that this procedure is applied only to the connected components in G'. Thus the degree of a nodes refers to the degree in G'. This procedure may decrease the number of black components in G but never increase it. The following phase will unify all black components into one component.

Phase IV: Choose one or two gray nodes to connect two or more black components. Repeat until only one black component left.

Not that the performance of this algorithm depends on the implementation of each phase. However, no matter how an MIS is computed in Phase I, its relationship with an optimal CDS is fixed by Lemma 3.1.



(a) (b)

Figure 1: *(a) A unit-disk graph G with 144 nodes. (b) The induced graph G' of all white nodes after Phase II.*

The number of black nodes connecting the MIS in Phase II can be analyzed with a similar way as proposed in [10]. Thus Phase II may introduce $(ln\ 4 + 1) \cdot opt$ number of black nodes, where *opt* is the size of an MCDS. On the other hand, by exploring the special geometric structure (each white node is adjacent to at least one gray node in Phase II) of G', it is possible to find out minimum number of nodes to extend the black components in Phase III to cover all white nodes. Since Phase II produces at most *opt* number of black components, Phase IV introduces at most twice of *opt* number of black nodes. Based on this analysis, we have the following conjecture:

Conjecture 1 *The proposed CDS construction algorithm has performance ratio at most* $ln\ 4 + 4$.

5 Conclusion

In this paper, we propose a heuristic to compute connected dominating set in unit-disk graphs. The idea of this algorithm is novel in that it does not include the members of a maximal independent set in the resultant CDS. We conjectured that our algorithm has performance ratio at most $ln\ 4 + 4$.

References

[1] K. M. Alzoubi, P.-J. Wan, O. Frieder, Message-Optimal Connected-Dominating-Set Construction for Routing in Mobile Ad Hoc Networks, *ACM MOBIHOC* 2002.

[2] K. M. Alzoubi, P.-J. Wan, O. Frieder, New Distributed Algorithm for Connected Dominating Set in Wireless Ad Hoc Networks, *HICSS35*, Hawaii, Janauary 2002.

[3] S. Butenko, X. Cheng, C.A.S Oliveira, and P.M. Pardalos, A new heuristic for the minimum connected dominating set problem on ad hoc wireless networks, to appear in Cooperative Control and Optimization, Kluwer Academic Publisher, pp. 61-73, 2004.

[4] X. Cheng, M. Ding, D.H. Du, and X. Jia, Virtual backbone construction in multihop ad hoc wireless networks, to appear in *Special Issue on Ad Hoc Networks* of *Wireless Communications and Mobile Computing*, 2004.

[5] 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.

[6] B. N. Clark, C. J. Colbourn and D. S. Johnson, Unit Disk Graphs, *Discrete Mathematics*, Vol. 86, pp. 165-177, 1990.

[7] F. Dai and J. Wu, Distributed dominate pruning in ad hoc wireless networks, *Proc. ICC*, p. 353, 2003.

[8] M. Ding, X. Cheng, and G. Xue, Aggregation tree construction in sensor networks, *Proc. of IEEE VTC*, 2003.

[9] M.R. Garey and D.S. Johnson, Computers and Intractability: A guide to the theory of NP-completeness, W. H. Freeman and Company, San Francisco, 1979.

[10] S. Guha and S. Khuller, Approximation Algorithms for Connected Dominating Sets, *European Symposium Algorithms*, Barcelona, Spain, September 25-27, 1996.

[11] M. Joa-Ng and I.-T. Lu, A Peer-to-Peer zone-based two-level link state routing for mobile Ad Hoc Networks, *IEEE Journal on Selected Areas in Communications*, Vol. 17(8), pp. 1415-1425, August 1999.

[12] D.B. Johnson, D. A. Maltz, Y.-C. Hu and J. G. Jetcheva, The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks, Internet Draft draft-ietf-manet-dsr-05.txt, March 2001.

[13] B. Liang and Z.J. Haas, Virtual backbone generation and maintenance in ad hoc network mobility management, *IEEE INFOCOM* 2000, pp.1293-1302.

[14] M. Min, C.X. Huang, S. C.-H. Huang, W. Wu, H. Du, and X. Jia, Improving Construction for Connected Dominating Set with Steiner Tree in Wireless Sensor Networks, *manuscript*, 2004.

[15] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen and J.-P. Sheu, The Broadcast Storm Problem in a Mobile Ad Hoc Network, *Proc. MOBICOM*, Seattle, Aug. 1999, pp. 151-162.

[16] G. Pei, M. Gerla and T.-W. Chen, Fisheye state routing: a routing scheme for ad hoc wireless networks, *ICC 2000*, Vol. 1, pp. 70-74.

[17] C. E. Perkins, E. M. Royer and S. R. Das, Ad Hoc On-demand Distance Vector (AODV) Routing, Internet Draft draft-ietf-manet-aodv-08.txt, March 2001.

[18] P. Sinha, R. Sivakumar and V. Bharghavan, Enhancing ad hoc routing with dynamic virtual infrastructures, *INFOCOM 2001*, Vol. 3, pp. 1763-1772.

[19] R. Sivakumar, P. Sinha and V. Bharghavan, CEDAR: a core-extraction distributed ad hoc routing algorithm, *IEEE Journal on Selected Areas in Communication*, Vol. 17 (8), 1999, pp. 1454-1465.

[20] P.-J. Wan, K.M. Alzoubi, and O. Frieder, Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks, *IEEE INFOCOM*, pp. 1597-1604, 2002.

[21] J. Wu and F. Dai, Broadcasting in ad hoc networks based on self-pruning, *IEEE INFOCOM*, 2003.

[22] J. Wu and H. Li, On calculating connected dominating set for efficient routing in ad hoc wireless networks, *Proc. The 3rd International workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, 1999, pp.7-14.