1-2014

A Distributed Greedy Algorithm for Constructing Connected Dominating Sets in Wireless Sensor Networks

Akshaye Dhawan
Ursinus College, adhawan@ursinus.edu

Nicholas A. Scoville
Ursinus College, nscoville@ursinus.edu

Michelle Tanco
Ursinus College, mitanco@ursinus.edu

A Distributed Greedy Algorithm for Constructing Connected Dominating Sets in Wireless Sensor Networks

Akshaye Dhawan, Michelle Tanco and Nicholas Scoville
Department of Mathematics and Computer Science, Ursinus College, 610 E Main Street, Collegeville, PA, USA
{*adhawan, mitanco, nscoville*}*@ursinus.edu*

Keywords: Wireless Sensor Networks, Dominating Sets, Distributed Algorithms

Abstract: A Connected Dominating Set (CDS) of the graph representing a Wireless Sensor Network can be used as a virtual backbone for routing in the network. Since sensor nodes are constrained by limited on-board batteries, it is desirable to have a small CDS for the network. However, constructing a minimum size CDS has been shown to be a NP-hard problem. In this paper we present a distributed greedy algorithm for constructing a CDS that we call Greedy Connect. Our algorithm operates in two phases, first constructing a dominating set and then connecting the nodes in this set. We evaluate our algorithm using simulations and compare it to the two-hop K2 algorithm in the literature. Depending on the network topology, our algorithm generally constructs a CDS that is up to 30% smaller in size than K2.

1 INTRODUCTION

Wireless Sensor Networks (WSNs) have attracted considerable research interest in the past decade (Iyengar and Brooks, 2004) (Akyildiz et al., 2002) (Chong and Kumar, 2003). They have evolved from research to deployment with many environmental, security, energy and other applications. WSNs consist of a number of low-cost sensors scattered in a geographical area of interest and connected by a wireless RF interface. Sensors gather information about the monitored area and send this information to gateway nodes known as sinks. Most common network models consist of a distributed and localized control with no central management. Each sensor serves as both a data gathering source and a router, forwarding messages from other nodes. In order to keep their cost low, the sensors are equipped with limited energy (Feeney and Nilsson, 2001) (Feeney, 2001) and computational resources. The energy supply is typically in the form of a battery and once the battery exhausted, the sensor is considered to be dead. A key approach to solve the problem of data gathering and communication involves the construction of a connected dominating set (CDS) that serves as a virtual backbone for the network.

In this paper, we use a graph $G = (V,E)$ to represent the wireless sensor network, where V is the set of sensors in the network and an edge $(u,v) \in E$ represent a link between two sensors u,v that are within communicating distance of each other. We also assume that all sensors are deployed on a 2-dimensional plane and have a uniform transmission range. The resulting graph is known as a Unit-Disk Graph (UDG) (Clark et al., 1990) since the uniform transmission range results in edges of equal (or unit) weight. We also assume that the transmission range is at least twice the sensing range since as shown in (Zhang and Hou, 2005) a covered network is also connected if this is true.

Given such a representation of a sensor network, a *dominating set* (DS) of a graph G is a subset $D \subset V$ such that for all $u \in V$ either $u \in D$ or u is adjacent to a node in D (i.e., $(u,w) \in E$ for some $w \in D$). Nodes in the dominating set D are referred to as *dominators* and the remaining nodes in $V - D$ are referred to as *dominatees*. A Connected Dominating Set (CDS) is a set that is dominating and induces a connected subgraph. In other words, it is a set of nodes $C \subset V$ such that the nodes in C are both dominating and connected.

The construction of a CDS provides the network with a virtual backbone over which routing, multicast and broadcast can be performed since every node is either in the backbone or has a neighbor in the backbone. Also, the construction of a CDS, allows the network to adapt to changes in the topology since only the nodes in the CDS need to be aware of routing information. By being connected the backbone can relay a message to either the destination directly (if the destination is in the CDS) or through the domi-

nator of the destination. Since the nodes in the CDS are actively draining their batteries by serving as relay nodes for the network, it is desirable to construct a minimum size CDS. However this has been shown to be a NP-hard problem in (Clark et al., 1990). Much attention has been given to centralized algorithms based on the use of a maximal independent set (MIS) to construct a CDS. The current best performance ratio of $4.8 + ln5$ was shown in (Li et al., 2005).

In this paper, we present a two-phase distributed and localized greedy algorithm that first constructs a dominating set (Phase 1) and then connects it (Phase 2). Our algorithm assumes that each sensor has a unique identifier. The resulting CDS has been shown to be significantly smaller in size to that of a comparable distributed algorithm in the literature that we call K2 (Dai and Wu, 2004).

The remainder of this paper is organized as follows in Section 2, we look at the literature on construction of connected dominating sets. In Section 3 we explain our two phase algorithm. We look at a simulation evaluation of our algorithm in Section 4. Finally, we conclude in Section 5.

2 Related Work

In this section we briefly summarize related work in this area. Considerable work has been done in the development of both centralized and distributed algorithms for CDS construction in the literature. Below we focus mostly on distributed algorithms.

The applications of a connected dominating set to routing in ad hoc networks were first outlined in (Ephremides et al., 1987) where they presented the idea of constructing a virtual backbone and its application to routing. This paper led to several papers that design approximation algorithms for this problem. A coloring scheme similar to the one we use is a common theme in many of these papers with all nodes being white initially, with dominator's being black and dominatee's being grey at the conclusion of these algorithms.

(Guha and Khuller, 1998) presents a centralized algorithm with a $O(H\Delta)$ approximation factor where Δ is the maximum degree and H is the harmonic function. In (Ruan et al., 2004) the authors present a 1-phase greedy algorithm that has a performance ration of $2 + ln\Delta$. (Funke et al., 2006) was one of the first distributed algorithms to show an improved analysis of the relationship between the size of a maximal independent set and a minimum CDS in a unit disk graph, which yields better bounds for many other algorithms. (Wan et al., 2002) presents a distributed algorithm for CDS construction by constructing a spanning tree first and then labeling every node in the tree as a dominator or dominatee. The Performance Ratio for this algorithm was shown to be 8. In (Alzoubi et al., 2002) the same authors noticed the difficulty of maintaining a CDS and designed a localized 2-phase algorithm that uses a Maximal Independent Set but this algorithm has a PR of 192. (Li et al., 2005) presents the best known PR of $4.8 + ln5$ in a centralized algorithm. The algorithm is known as S-MIS and uses a Steiner Tree to construct a CDS. In this algorithm they build a Maximal Independent Set in Phase 1. Then in Phase 2, they employ a greedy algorithm to construct a Steiner tree with minimal number of Steiner nodes to connect the nodes in the MIS. They mention that a distributed implementation is possible but do not elaborate on this algorithm or its PR.

(Wu and Li, 1999) presents an earlier version of a pruning algorithm that the authors refined into the K2 algorithm. Finally, we look at the K2 algorithm (Dai and Wu, 2004) that we compare ourselves against. The algorithm is a two phase algorithm which first creates a connected dominating set and then reduces the size of the set. In phase one, each sensor adds itself to the dominating set if any two of its neighbors are not neighbors. It is clear that if we start with a connected graph we will get a connected dominating set since any two non-connected sensors that share a neighbor will be connected. However, this set is likely to contain many more nodes than necessary since the marking process was very simple. In order to reduce the size of the set, the authors use a k-reduction (where k is the number of hops the algorithm is looking at) to remove unnecessary sensors from the set. For each sensor in the dominating set we consider every k-hop group of neighbors where each member of the group is in the dominating set. If one of these groups contains every neighbor of the original sensor in its neighbor set, we remove the original sensor from the dominating set. The dominating set is still connected by the group of k neighbors. We call this the K2 algorithm because we compare ourselves against the 2-hop version of this algorithm. As k increases, the size of the connected dominating set decreases. However, the message and time complexity increase since we have to check each size k group of neighbors for each sensor. For the purposes of this paper we let $k = 2$, since an algorithm cannot be localized and use a reasonable number of messages if it requires more than 2-hops of information. This algorithm has the benefit of the CDS being easy to maintain.

3 Greedy Connect Algorithm for CDS construction

In this section we present our greedy, distributed algorithm for constructing a connected dominating set. The algorithm is a two-phase algorithm. In Phase 1, we construct a dominating set and in Phase 2, we connect the dominating set to form a connected dominating set. We begin by presenting Phase 1 - a greedy approach to constructing a dominating set.

In this section, we will use a graph $G = (V,E)$ to represent the sensor network. We will also use the notation $N(u)$ to denote the one-hop neighbor set of node $u \in V$. We also assume every sensor node to have a unique identifier. The sensor nodes fields as used in our algorithm are shown in Table 1.

Table 1: Fields for a given sensor node v

Field	Meaning
v.COLOR	The current color of the sensor
v.ID	Unique identifier for the sensor
v.WhiteCount	Number of white nodes in $N(v)$

3.1 Phase 1: Greedy construction of a dominating set

Phase 1 uses node coloring to implement its greedy approach. We summarize the meaning of the colors in Table 2. Initially we start out with all nodes being colored white. The heuristic is greedy because our criteria for adding nodes to the dominating set is to pick the node that dominates the highest number of non-dominated nodes. The color white represents nodes that have not been dominated. When a node is added to the dominating set, it is colored black and its neighbors are colored grey to indicate that they have been dominated.

Table 2: Node color assignments for Phase 1.

Color	Meaning
White	Undiscovered by the Dominating Set
Grey	Dominated but has white neighbors
Black	Dominated and has no white neighbors

In the first pass, every node every node exchanges its white neighbor count with its neighbors. At this point, the node with the highest white neighbor count adds itself to the dominating set by changing its color to black and changing the color of its white neighbors to grey. Since this pass happens asynchronously, there is a possibility (as shown by the example in Section 3.3) that some nodes have not yet been dominated. The second pass looks at only those nodes that are still white and essentially repeats the process to ensure that the set is dominating.

Require: $\forall$ v $\in$ V v.COLOR $\leftarrow$ WHITE
if v.COLOR ==WHITE **then**

Initially every node will do this once
Request the white neighbor count for every neighbor $u \in N(v)$
if v.WhiteCount >= u.WhiteCount $\forall u \in N(v)$ **then**
v.COLOR $\leftarrow$ BLACK
for every neighbor $u \in N(v)$ **do**
if u.COLOR == WHITE **then**
u.COLOR $\leftarrow$GREY
end if
end for
end if
end if
if v.COLOR==WHITE **then**
Request the white neighbor count for every neighbor $u \in N(v)$
high$\leftarrow$ The node with the highest WhiteCount
high.COLOR $\leftarrow$ BLACK
for every neighbor $u \in N(high)$ **do**
if u.COLOR == WHITE **then**
u.COLOR $\leftarrow$GREY
end if
end for
end if

When the above algorithm concludes, we are left with a dominating set that is possibly fragmented into disconnected components. In Phase 2 we will connect these components. The time complexity of this phase is $O(n\Delta)$, where Δ is the maximum degree of a node in V. This is because the first pass takes $O(n\Delta)$ since every sensor exchanges information with its neighbors and the second pass takes $O(w\Delta)$ where w is the number of white nodes left after the first pass and $w \leq n$.

Lemma 1: The nodes colored Black at the end of Phase 1 represent a Dominating Set for the graph G.

Proof: Assume that at the end of Phase 1, there exists a node that is still colored white (i.e., it is not adjacent to a black node or colored black itself). In the second pass (that every white sensor goes through), each sensor either adds either itself or a neighbor to the dominating set. If a sensor is added to the dominating set, all its white neighbors are colored grey. Hence all white nodes must be either grey or black when the second pass of Phase 1 concludes. Therefore, such a node cannot exist and the set of black nodes is dominating.

3.2 Phase 2: Connecting the Dominating Set

In this subsection we will present our connection algorithm. However, before we can do so, we need to prove some properties of the dominating set constructed at the end of Phase 1 since we rely on these properties to come up with the construction that connects the disconnected dominating components formed at the end of Phase 1.

Lemma 2: Any component of a dominating set is separated by at most two vertices from another component.

Proof: Assume there is a component separated from all other components by at least three vertices at the end of the algorithm. We can visualize this scenario as $COMP1 - a - b - c - COMP2$ where a, b, c are the three vertices separating COMP1 and COMP2. Clearly the nodes a and c are dominated by the two components. Also, by Lemma 1, b must be adjacent to or in the dominating set. If b is dominated by a different component, this would create a path of length two from both COMP1 and COMP2 to the component dominating b, thereby leading to a contradiction.

Lemma 3: Connecting the dominating set created by the greedy algorithm takes adding at most $2(n-1)$ vertices to the dominating set where n is the number of components of the dominating set.

Proof: Base Case: Consider a dominating set of two components. By Lemma 2 there are at most two vertices need to connect this set. Thus the base case holds: to connect a set of two components we need at most $2 = 2(2-1)$ vertices. By the inductive hypothesis a dominating set of $n-1$ components can be connected with $2((n-1)-1) = 2(n-2) = 2n-4$ components. By Lemma 2, another component is at most 2 vertices away. Then the limit for n components is $2n-4+2 = 2n-2 = 2(n-1)$ vertices.

Based on these two lemmas we now present the connection phase of our algorithm. We make the assumption that every sensor has a unique identifier associated with it. As can be seen from the algorithm below, we initialize the component id of each grey sensor to -1 and for every connected dominating component, we initialize its component number to that of the highest id sensor in that connected component. This allows every component to have an associated id - that of the highest id sensor in that component. Now, if a grey is connected to two components with different id's, it colors itself black (in order to connect these two components). It also updates its id to that of the largest of these components. Since two components can be at most two hops away (by Lemma 2), in the next for loop, we check if any pair of sensors connects two disconnected dominating components and color this pair black.

Require: Recursively compute a component number based on the id of the largest id sensor for that component. Initialize all dominatees (grey) to an component number of -1.
for all non-dominating (grey) nodes in V **do**
  if v is connected to two dominating components with different IDs **then**
    v.COLOR ← BLACK
    v.ID ← max component ID of the components it connects
  end if
end for
for every pair (u,v) of non-dominating connected sensors **do**
  highU ← max(N(u).ID)
  highV ← max(N(v).ID)
  if (highU ≠ highV) **then**
    u.COLOR ← BLACK
    v.COLOR ← BLACK
   change v.ID and u.ID to the component ID of the component they joined
  end if
end for

3.3 An example

We will now look at an example of both phases in operation. We use the network shown in Figure 1 as our exemplar. The figure shows the sensors and the resulting graph representing the network. As mentioned in (Zhang and Hou, 2005), the transmission radius is double the sensing radius. At the start of the algorithm all vertices are colored white (shown here in yellow). The number next to each sensor indicates its white neighbor count which at initialization is just the degree of each node.

In the first round of Phase 1 each vertex checks if it has the highest (or tied highest) white neighbor count in its neighborhood. If so, it adds itself to the dominating set (denoted in black) and tells all its neighbors they have been dominated (denoted in gray). If not, the vertex waits for round two. Since each vertex is discovering its white neighbor count simultaneously, in this example only the vertex of white neighbor count five will add itself. The vertex connected to two leaves was not added since it had a neighbor with a higher white count. In the second pass, each of the white sensors checks which neighbor has the highest count (including itself) and tells that neighbor to add itself to the dominating set. The coloring at the end of each pass of Phase 1 along with the component id's of the two black nodes is shown in Figure 2. Here we as-

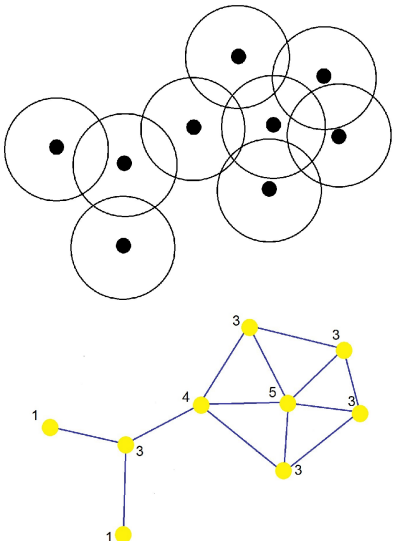

Figure 1: Example network (with sensing radius shown) and resulting graph.

sume that the two dominating (black) nodes have id's of 2 and 5 respectively. Also, all the grey nodes are initialized to component id's of -1.

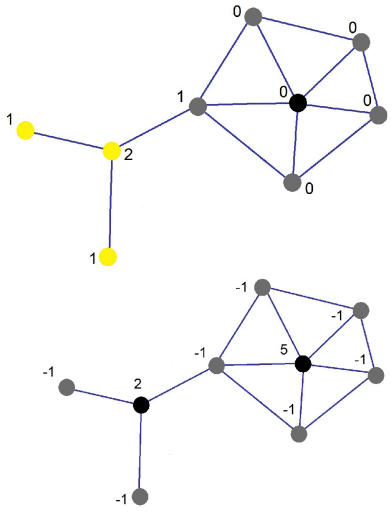

Figure 2: Coloring after two rounds of Phase 1.

At this point we have a Dominating Set with two components (each of the two black nodes being a separate component). They are dominating but not connected since they are separated by a grey node. In Phase 2, since this intermediate grey node is connected to two components with different id's it will color itself black and add itself to the dominating set, resulting in a connected dominating set as shown in Figure 3. This node will assume the component id of the higher of these two nodes (i.e., 5) and this component id will propagate across the CDS resulting in all the black nodes having a component id of 5 as shown in the second figure of Figure 3.

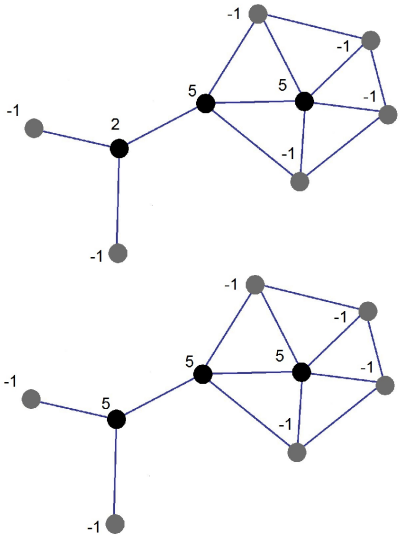

Figure 3: Coloring after Phase 2.

4 Results

To compare the performance of our algorithm we set up a simulation environment in $C++$. For our simulation setup, we create networks of sensors with 100 nodes scattered randomly in a 100x100m area. We varied the transmission range of each sensor from 10m to 30m in increments of 5m. For a given range, we generated 10 random graphs of that size. We then ran both our algorithm and the K2 algorithm (Dai and Wu, 2004) on each graph. Our results are shown in Figure 4a.

Each data point in the figure represents an average of the ten graphs generated for that size. As can be seen from the figure, our algorithm is significantly better than K2 at lower ranges. At the range 10, the size of the CDS constructed by our algorithm is about 30% smaller than that of K2. This trend is preserved at range 15 as shown in Figure 4b, but drops to a smaller number at higher ranges. In this figure, we plot the size of the CDS for both algorithms in each of the random networks we generated with this range. As can be seen from the figure, our algorithm is consistently better than K2. We believe that the at higher ranges the network is so dense that only a few nodes will suffice to construct a CDS. As a result of this, it is difficult to see any difference between the two al-

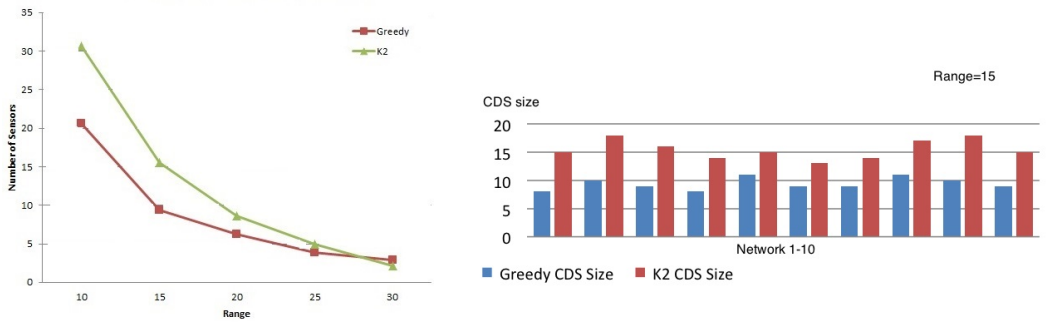

Figure 4: a. Average size of the CDS as the range is varied, b. Size of the CDS for different networks with range=15

gorithms.

One major point to note is that we have not yet conducted simulation studies on the impact of our algorithm on the lifetime of the network. This is part of our future work and will allow us to examine the message complexity of our algorithm when compared to other distributed algorithms to construct a CDS. (Moscibroda and Wattenhofer, 2005), (Cardei and Du, 2005), (Wu et al., 2001) all look at the construction of a power-aware connected dominating set by constructing a CDS, using it for a period of time and then computing a new connected dominating set so as to spread the burden of relaying across the different nodes. We are currently working on implementing our algorithm in a simulation environment that will allow us to track battery usage.

5 CONCLUSIONS

In conclusion, in this paper we present a 2-phase algorithm that starts with greedy coloring scheme to form a dominating set which we then connect using the sensor id's of the disconnected component. In simulation studies our approach has been shown to result in a smaller CDS than a popular 2-hop algorithm in the literature. Our future work includes studying the message complexity and the impact of the message passing on power using more detailed simulation studies.

REFERENCES

Akyildiz, I., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). A survey on sensor networks. *IEEE Communications*, 38(4):102–114.

Alzoubi, K. M., Wan, P.-J., and Frieder, O. (2002). Message-optimal connected dominating sets in mobile ad hoc networks. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 157–164. ACM.

Cardei, M. and Du, D.-Z. (May 2005). Improving wireless sensor network lifetime through power aware organization. *Wireless Networks*, 11:333–340(8).

Chong, C.-Y. and Kumar, S. (2003). Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8):1247–1256.

Clark, B., Colbourn, C., and Johnson, D. (1990). Unit disk graphs. *Discrete Mathematics*, 86:165–177.

Dai, F. and Wu, J. (2004). An extended localized algorithm for connected dominating set formation in ad hoc wireless networks. *IEEE Trans. Parallel Distrib. Syst.*, 15(10):908–920.

Ephremides, A., Wieselthier, J. E., and Baker, D. J. (1987). A design concept for reliable mobile radio networks with frequency hopping signaling. *Proceedings of the IEEE*, 75(1):56–73.

Feeney, L. and Nilsson, M. (2001). Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1548–1557 vol.3.

Feeney, L. M. (2001). An energy consumption model for performance analysis of routing protocols for mobile ad hoc networks. *Mob. Netw. Appl.*, 6(3):239–249.

Funke, S., Kesselman, A., Meyer, U., and Segal, M. (2006). A simple improved distributed algorithm for minimum cds in unit disk graphs. *ACM Trans. Sen. Netw.*, 2(3):444–453.

Guha, S. and Khuller, S. (1998). Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374–387.

Iyengar, S. S. and Brooks, R. (2004). Computing and communications in distributed sensor networks.

Li, Y., Thai, M. T., Wang, F., Yi, C.-W., Wan, P.-J., and Du, D.-Z. (2005). On greedy construction of connected dominating sets in wireless networks: Research articles. *Wirel. Commun. Mob. Comput.*, 5(8):927–932.

Moscibroda, T. and Wattenhofer, R. (2005). Maximizing the lifetime of dominating sets. In *IPDPS '05: Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05) - Workshop 12*, page 242.2, Washington, DC, USA. IEEE Computer Society.

Ruan, L., Du, H., Jia, X., Wu, W., Li, Y., and Ko, K.-I.

(2004). A greedy approximation for minimum connected dominating sets. *Theor. Comput. Sci.*, 329(1-3):325–330.

Wan, P.-J., Alzoubi, K. M., and Frieder, O. (2002). Distributed construction of connected dominating set in wireless ad hoc networks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1597–1604. IEEE.

Wu, J., Gao, M., and Stojmenovic, I. (2001). On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks. In *ICPP '02: Proceedings of the 2001 International Conference on Parallel Processing*, pages 346–356, Washington, DC, USA. IEEE Computer Society.

Wu, J. and Li, H. (1999). On calculating connected dominating set for efficient routing in ad hoc wireless networks. In *Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*, pages 7–14. ACM.

Zhang, H. and Hou, J. C. (2005). Maintaining sensing coverage and connectivity in large sensor networks. *Ad Hoc & Sensor Wireless Networks*, 1(1-2):89–124.