A Constant Factor Distributed Algorithm for Computing Connected Dominating Sets in Wireless Sensor Networks

Kamrul Islam, Selim G Akl, and Henk Meijer
School of Computing, Queen's University
Kingston, Ontario, Canada K7L 3N6
{islam,akl,henk}@cs.queensu.ca

Abstract

Connected dominating sets (CDSs) are probably the most common way of constructing virtual backbones for broadcasting operation in wireless sensor networks. This is because such backbones guarantee to reduce unnecessary message transmissions or flooding in the network. In this paper we propose a simple distributed algorithm to construct a small-sized CDS. Considering the sensors deployed in the plane, our main idea is based on the computation of convex hulls of sensor nodes (nodes are considered points in the plane) in a distributed manner and a simple coloring scheme, which produces a CDS in unit disk graphs whose size is at most $50.54 * |MCDS|$, *where* $|MCDS|$ *is the size of a minimum CDS. To the best of our knowledge, this is a significant improvement over the best published results in the same context [5]. We also analyze grids and trees to compute the approximation ratios for the problem. We show that our algorithm produces an optimal CDS if the graph is a tree and in the case of grids the approximation factor is* 2.

1. Introduction

A wireless sensor network is composed of a set of battery-powered sensors which communicate with one another over wireless links directly if they are within their transmission ranges, otherwise they rely on other sensors to establish a connection between them. The wireless and self-configurable capability of sensors networks make them well-suited for several scenarios, such as battle-fields, target tracking, and most scenarios that involve monitoring an environment. However, the lack of a communication infrastructure presents several challenging and interesting research issues in the design of communication protocols for sensor networks.

Establishing a virtual backbone in wireless networks, which can be seen as an analogue of the fixed communication infrastructure in wired networks, helps reduce interference, and hence the number of retransmissions for the energy-constrained sensors. This is because only a fixed and "small" number of sensors are engaged in transmissions. A virtual backbone typically consists of a subset of nodes in the network which are responsible for receiving and transmitting messages throughout the network. The connected dominating set is one of the earliest structures proposed as a candidate for a virtual backbone in wireless networks [10]. Given a graph $G = (V, E)$ where V and E are sets of nodes and edges respectively, a subset D of V is a *dominating set* if each node $u \in V$ is in D or is adjacent to some node $v \in D$. A subset K of V is a *connected dominating set* if K is a dominating set and induces a connected subgraph. A connected dominating set with minimum cardinality is called a *minimum connected dominating set* (*MCDS*). A subset of V is an *independent set* if, for any pair of nodes in the subset, there is no edge between them. An independent set is a *maximal independent set* (*MIS*) if no more nodes can be added to the independent set to generate a larger independent set. An *MIS* is a dominating set. In the context of sensor networks, a well studied problem is that of finding an *MCDS* in a unit disk graph (*UDG*). *UDG*s are a class of graphs which are often used to model connectivity in sensor networks where an edge between two nodes exists if and only if their Euclidean distance is at most one. In this paper, we consider G to be a *UDG* to model the sensor network and devise a distributed algorithm for the connected dominating set problem.

One of the popular approaches to compute a small-sized *CDS* in sensor networks modeled as a *UDG*, is to find an *MIS* (or assume that an *MIS* is given) in the network graph and then select a set of nodes called *connector*s to connect the nodes in the *MIS*. The main difficulty of achieving a small approximation factor lies in the selection of connectors because there is a constant bound of the size of *MIS* in *UDG* in terms of the size of an *MCDS*. In general, the smaller the number of connectors, the better (smaller) the approximation ratio. So clever strategies are sought to connect nodes in *MIS*. There exist several distributed and centralized algorithms to bound the number of the connectors. The readers are referred to the most recent survey in [4] to know about these techniques. Our proposed algorithm is slightly different from the usual approach. Instead of finding an *MIS* in the whole network, our approach first builds a *small* connected subgraph of the network graph based on the computation of convex-hulls of the 1-hop neighborhood of individual sensors, then finds an *MIS* in the subgraph and finally connects the nodes in the *MIS* through other nodes. We also provide an analysis of the relationship between the sizes of an *MIS* and an *MCDS*.

The rest of the paper is organized as follows. In Section 3 we provide definitions and assumptions that are used throughout the paper. The distributed algorithm is presented in Section 4. We provide an analysis of the size of an *MIS* and the size of an *MCDS* and present the main result of the approximation factor for the size of the computed *CDS* in Section 5. This section also contains specific results when the underlying unit disk graph is a tree and a grid. Time and message complexities of the algorithm are presented Section 6. We conclude in Section 7.

2. Related Work

Guha and Khuller [12] studied the *MCDS* problem and showed that this problem is NP-hard in arbitrary undirected graphs. They present a centralized approximation algorithm which guarantees a $O(\log n)$ approximation factor where n is the number of nodes in the graph. Later it was shown in [8] that computing an *MCDS* is NP-hard even for *UDG*s. The first centralized method towards solving this problem achieves a 10-approximation factor and was given by Marathe [14]. Cheng *et. al* [7] presented a polynomial-time approximation scheme that guarantees an approximation factor of $(1 + 1/s)$ with running time of $n^{O((s \log s)^2)}$. However, in reality, centralized algorithms can not be effectively applied in large sensor networks because of huge power consumption. Therefore, distributed algorithms are sought to find an approximate *MCDS* in such networks. Several distributed approximation algorithms exist in the literature, some of the earlier attempts include the algorithms of Stojmenovic and Xiang-Li [17, 19] which distributedly construct a *CDS* of size $n/2$ times that of an *MCDS*.

Alzoubi *et al.* [3] proposed the first distributed algorithm guaranteeing a constant approximation factor for *CDS* construction based on *MIS* in *UDG*s. The main features of their distributed algorithm include an approximation factor of 8, O(n) time complexity, and O$(n \log n)$ message complexity. One of the important results established in that paper, is that $\Omega(n \log n)$ a lower bound on the message complexity of any distributed algorithm for nontrivial *CDS*s.

However, the algorithm in [3] suffers from the construction of a spanning tree in wireless networks, which is expensive since the time complexity is $Diam(G)$ in which case a node “needs to” know the whole topology of the network. Building a spanning tree and maintaining a CDS in large and volatile WSNs with common node and link failures incur a significant number of retransmissions and communication overhead. Later, Alzoubi *et al.* came up with a constant factor distributed algorithm [1] that does not rely on the construction of spanning trees. However, the algorithm produces a CDS with size at most $192 * |\text{MCDS}| + 48$. Lately, Funke et al. [11] gave a distributed algorithm which provides an approximation factor of 6.91 and outperforms all other approximation factors of previous distributed algorithms. Although the factor is small, the time complexity of their algorithm is still $Diam(G)$, which is O(n) in the worst-case.

In an attempt to obtain a small constant factor, a distributed algorithm is proposed in [13] which produces a CDS with size at most $172 * |\text{MCDS}| + 33$ in UDGs. Subsequently, the next level of reduction in the constant factor was achieved by the distributed algorithm in [6], where the factor is 147 (specifically, $147 * |\text{MCDS}| + 43$). Recently D. Chen *et al.* managed to achieve even a smaller constant factor by a simple distributed algorithm. The cardinality of the CDS produced by their algorithm is at most $76 * |\text{MCDS}| + 19$. Here, we present a technique that reduces this constant factor to 50.54, that is, it produces a CDS with size at most $50.54 * |\text{MCDS}|$. We analyze special classes of graphs, for example, trees and grids, where the approximation ratios obtained by our algorithm are 1 (optimal) and 2, respectively.

3. Model and Definitions

We assume that sensor nodes are deployed in the plane and the network of nodes is modeled by an undirected graph $G = (V, E)$, where link $(u, v) \in E$ between two sensor nodes $u, v \in V$ exists if they are within their transmission range. Every sensor node $u \in V$ has the same transmission range which is normalized to one. Neighbor sets $N(u)$ and $N[u]$ of sensor node u are defined as $N(u) = \{v|(u, v) \in E, u \neq v\}$ and $N[u] = N(u) \cup \{u\}$.

We use a computational geometry idea called the *convex hull* in our algorithm. The convex hull of a set of points in the plane is defined as the smallest convex polygon that includes all the points in the set such that each point is either a corner of the convex hull or lies inside it. The convex hull of $N(u)$ is denoted by $CH(N(u))$. We say $CH(P)$ of some point set $P \subseteq V$ is *completely included* (or just *included*) in $CH(Q)$ of point set $Q \subseteq V$ if $CH(P) \subseteq CH(Q)$. Each sensor node u has a unique integer id, denoted by $id(u)$.

We assume a synchronized message passing system as in [15] in which time is divided into equal rounds. In each communication round a node of the network graph is capable of receiving messages from its neighbors, performing local computations, and broadcasting a message to its neighbors. The time complexity of an algorithm is the number of rounds it needs to produce the desired result. Although we use a synchronous system, it is well known that a synchronous message passing algorithm can be turned into an asynchronous algorithm with the same time complexity, but with a larger message complexity [15]. Sensor nodes know their geographic locations and obtain their neighbors' locations through exchanging *"Hello"* messages.

4. A Distributed Algorithm for *CDS*

From now on instead of using 'sensor nodes' we use only 'nodes'. In this section we present a simple algorithm for computing a CDS in G based on the computation of convex-hulls of the 1-hop neighborhood of individual nodes and a coloring scheme of the nodes of G. There are three colors, namely 'black', 'grey' and 'white'. Initially all nodes are white.

Here we outline the algorithm (call our algorithm *A*):

The algorithm consists of two phases (Phase I and Phase II). In Phase I, we distributedly form a primary CDS denoted by B, which is pruned in the next phase (II) to generate the final CDS denoted by CDS* (CDS* is the solution we are looking for). After the execution of Phase I, all nodes change their respective colors and become either black (if the node belongs to B) or grey (otherwise).

In Phase II, first we prune the set B of black nodes. In order to do so, we apply a distributed MIS algorithm in B. Let the set of MIS nodes returned by the algorithm be $S \subseteq B$. Although an MIS is a dominating set in a graph, the subset S is not a dominating set in G since S is obtained from the nodes in B (not from the nodes V). Then we form a dominating set in G consisting of nodes in S and a set B' of *additional* nodes ($B' \subseteq (V \setminus S)$). Thereafter, a set C of connectors is selected from $V \setminus S$ to connect the nodes in S such that $S \cup B' \cup C$ forms the desired CDS*. Following is a detailed description of the two phases of algorithm A:

Phase I (Coloring nodes):

Initially all nodes are white. Each node u locally computes the convex hull $CH(N[u])$ following an optimal convex hull algorithm [9]. Then u compares $CH(N[u])$ with $CH(N[v])$ for all $v \in N(u)$ and colors itself in the following way:
(1) **If** there is no v such that $CH(N[u]) \subset CH(N[v])$ **Then** u colors itself black.
(2) **If** there is some v such that $CH(N[u]) \subset CH(N[v])$ **Then** u colors itself grey.
(3) **If** there are some v's such that $CH(N[u]) = CH(N[v])$ **Then** node u colors itself black **If** it has the smallest id among all such v's, otherwise it colors itself grey.

This phase colors every white node either black or grey. We show that that set of black nodes B forms a CDS in G (Lemma 5.1).

In Figure 1, we give a simple example where the black nodes by Phase I of the algorithm *A* form a CDS. Node 4 colors itself black since its convex hull

$CH(N[4]) = \{1, 2, 3, 5\}$ (shown in dashed line segments) is not included in any of the convex hulls $CH(N[1])$, $CH(N[2])$, $CH(N[3])$ or $CH(N[5])$. For the same reason, nodes 5 and 6 color themselves black. However, nodes 1, 2, 3, 7, 8, and 9 are colored grey because each of their computed convex hulls is included in its corresponding neighbor's convex hull.

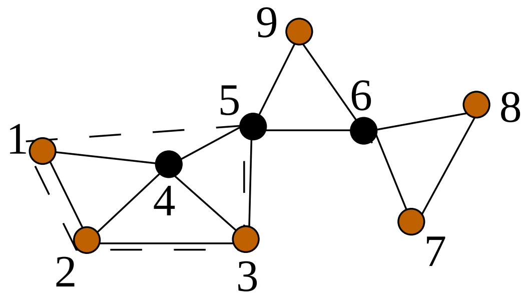


Figure 1. Node 4 is black since $CH(N[4]) = \{1, 2, 3, 5\}$ (shown in dashed line segments) is not included in any of $CH(N[1])$, $CH(N[2])$, $CH(N[3])$ or $CH(N[5])$. The same applies to nodes 5 and 6.

Phase II (Finding an MIS S and a set C of connectors):

This phase begins when Phase I is finished, that is, when there are no white nodes in G. In this phase, first, as mentioned before an MIS S is constructed from the set of black nodes B by following the distributed algorithm [16]. The authors in [16] present a very fast $O(\log^* n)$ distributed algorithm for computing such an MIS for any *growth bounded* graph, where the growth boundedness of a graph means that in a constant neighborhood of any node there is a constant number of independent nodes. (It can be mentioned that a UDG is a growth bounded graph since for any node u the number of independent nodes in its r-neighborhood, $N_r(u)$, is $O(r^2)$.)

As S is not a dominating set in G, some grey nodes in G may not be dominated (covered) by S. Note that according to the algorithm, for each grey node u there must be at least a black neighbor $v \in N(u) \cap B$ such that $CH(u) \subseteq CH(v)$, otherwise u would not be grey. Let $B(u)$ denote the set of such black neighbors of u.

Now we find an additional set B' of nodes from $V \setminus S$ in the following way to make $S \cup B'$ a dominating set in G. Call the grey nodes not dominated by S *orphan* nodes. An orphan node u selects a black node from $B(u)$ to dominate itself. If $|B(u)| > 1$, then u selects the one with the minimum id. The set of black nodes selected by all such grey nodes constitutes the set B'. Observe that $S \cup B'$ is now a dominating set in G because all the orphan nodes have at least one neighbor in B'. Change the color of the nodes in $B \setminus (S \cup B')$ into grey.

Although $S \cup B'$ dominates the nodes in G, it is not necessarily connected. Therefore, a set of connectors C from $V \setminus S$ is sought to connect the MIS nodes S in order to find a CDS in G. A node $u \in S$ connects to a node $v \in S$, $u \neq v$ with the minimum id, which is at most 3 hops away. Therefore, at most two intermediate nodes between u and v are selected to be in C. All such selected intermediate nodes form the connector set C. The nodes in C color themselves black. Finally, the union $S \cup B' \cup C$ forms the CDS* in G.

For an illustration, we provide a complete example of Phase I and Phase II of algorithm *A* is given in Figure 2:

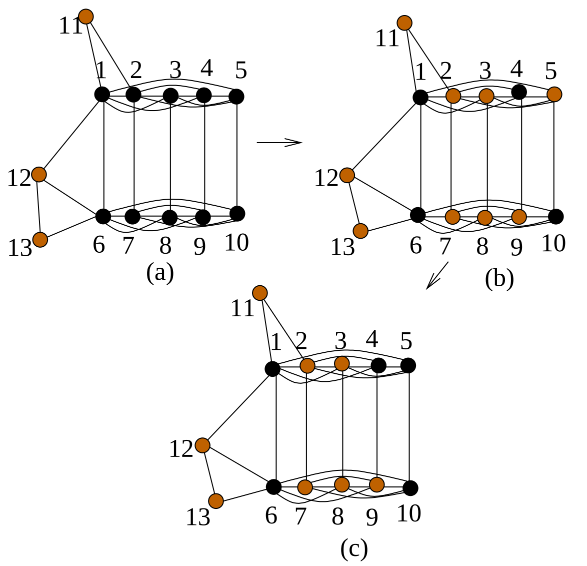


Figure 2. (a) After Phase I, $B = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. (b) S consists of nodes $\{4, 10\}$. Additional node set is $B' = \{1, 6\}$. (c) After Phase II the solution is $S \cup C \cup B' = \{1, 4, 5, 6, 10\}$, where $C = \{5\}$. The MCDS consists of $\{1, 6\}$.

The proposed distributed algorithm is given in Figure 3 which is executed at each node u.

Distributed Algorithm for CDS

PHASE I:

If $\exists v \in N(u)$ such that $CH(N[u]) \subset CH(N[v])$ **Then**
    Color u grey **Endif**
If $\nexists v \in N(u)$ such that $CH(N[u]) \subset CH(N[v])$ **Then**
    Color u black **Endif**
If $\exists v \in N(u)$ such that $CH(N[u]) = CH(N[v])$ **Then**
    If $id(u) < id(v), \forall v \in N(u)$ **Then**
        Color u black
    Else
        Color u grey
    Endif Endif

PHASE II:

If u is black **Then** Run MIS [16] **Endif**
If u is grey **Then** Select a node in $B(u)$ **Endif**
If u is in S **Then** Select connectors between u and v **Endif**
// $v \in S$ is at most 3 hops away and $id(u) > id(v)$

Figure 3. A Distributed algorithm for the CDS problem.

5. Theoretical Analysis

As mentioned before, the first phase of the algorithm is finished when each node is either black or grey. Here we provide a lemma which is essential for the proofs of other results. This lemma guarantees that the set of black nodes B found through Phase I is a CDS.

Lemma 5.1 *The set B forms a CDS in G.*

Proof We prove the lemma by contradiction. Suppose u is grey. Then from the algorithm (consider Phase I only) we derive either **Case 1**: $CH(N[u]) \subset CH(N[v])$ for some $v \in N[u])$ or **Case 2**: if such v does not exist, there are some nodes v with $CH(N[u]) = CH(N[v])$ and there is a v with $id(v) < id(u)$. In Case 1, if there are several such nodes with this property, take a node v with the maximal size (in terms of the number of vertices) $CH(N[v])$ (using the partial order "subset of or equal"). If there is still a tie, take the one with the minimum id. Node v cannot be colored grey at the beginning of the algorithm since this would mean the presence of a node w, where $w \in N(u)$ and $CH(N[u]) \subset CH(N[v]) \subset CH(N[w])$ implying that $CH(N[v])$ is not maximal. Also it can not be colored grey in the next step since that implies that there is a node w with $CH(N[u]) \subset CH(N[v]) = CH(N[w])$ and $id(w) < id(v)$, so v does not have the minimum id. Therefore, v is colored black.

Of all such nodes (v) with the property of Case 2, take the one with the minimum id. Node v is not colored grey at the beginning of the algorithm, since this would imply the presence of a node w with $CH(N[u]) = CH(N[v]) \subset CH(N[w])$ and $w \in N(v)$. However, it will be colored black in the next step but this is a contradiction. So B is a dominating set.

In order to prove that the node set B is connected, we show that there is a path between any two nodes $u, v \in B$ containing only black nodes. For the sake of contradiction assume such a path, $\pi = (u, \cdots, v_{i-1}, v_i, v_{i+1}, \cdots, v)$ contains a grey node $v_i \notin B$. Here the node that precedes v_i is labeled as v_{i-1} and the one that follows v_i is labeled as v_{i+1} on the path. Node v_i is colored grey since it has some neighbor $v_t \in N(v_i)$ (v_i is not an isolated node and has at least one neighbor) such that $CH(N[v_i]) \subseteq CH(N[v_t])$, where either $id(v_t) < id(v_i)$ and $CH(N[v_i]) = CH(N[v_t])$ or $CH(N[v_i]) \subset CH(N[v_t])$. If that is the case then v_t's neighborhood must include all of v_i's neighbors, that is, $v_{i-1}, v_{i+1} \in N(v_t)$. Thus instead of taking v_i in π, we can include black node v_t in π. If there are other grey nodes in the path π, then we can apply the same argument for all such grey nodes in the path. So the path $\pi = (u, \cdots, v_{i-1}, v_t, v_{i+1}, \cdots, v)$ contains only black nodes which is a contradiction to the original assumption that π contains some grey node. ∎

Since B is connected, we can apply the distributed MIS algorithm [16] to B which takes $O(\log^* |B|)$ time to compute an MIS S in B. We will prove that the cardinality of the union of S and a set of additional nodes, B' (i.e., $|S \cup B'|$) is at most 2.66 times the size of an MCDS of G.

Before that we need the following lemma:

Lemma 5.2 $|S| \leq 1.66 * |MCDS| + 2$

Proof Consider a node u in the MCDS and its 2-hop neighborhood $N_2[u]$ such that for any node $v \in N_2[u]$ we have $N[v] \subseteq N_2[u]$ (see Figure 4). In other words, v's 1-hop neighborhood is contained in $N_2[u]$.

Node u can have at most 5 MIS nodes, labeled u_i, $1 \le i \le 5$ as shown, in its 1-hop neighborhood. A node $v \in N_2[u]$ in the 2-hop neighborhood can be colored black if v is adjacent to both u_i and u_{i+1} (assume modulo arithmetic). This is because v's convex hull $CH(N[v])$ will not be included in any of its neighbors' convex hulls. Similarly, any node v^* which is 2-hop away from u and not adjacent to both u_i and u_{i+1} will not be colored black since $CH(N[v^*])$ must be included in the convex hull $CH(N[u^*])$ of some node $u^* \in N[u]$.

According to the above arrangement of the nodes we can have at most 11 black nodes in $N_2[u]$. As we apply an MIS algorithm, we can have at most six MIS nodes in $N_2[u]$ (the MIS nodes are labeled '1' in the figure).

Now we determine the minimum number of MCDS nodes that u must have in $N[u]$ in order to cover all the MIS nodes in $N_2[u]$. In order to do that we need to investigate the following four cases:

Case 1: At least four MCDS nodes, namely, u, u_i, u_{i+2} and u_{i+3} are required to cover (dominate) at most six MIS nodes (Figure 4(a)).

Case 2: At least three MCDS nodes (u, u_i, u_{i+2}) are required to cover at most five MIS nodes (Figure 4(b)).

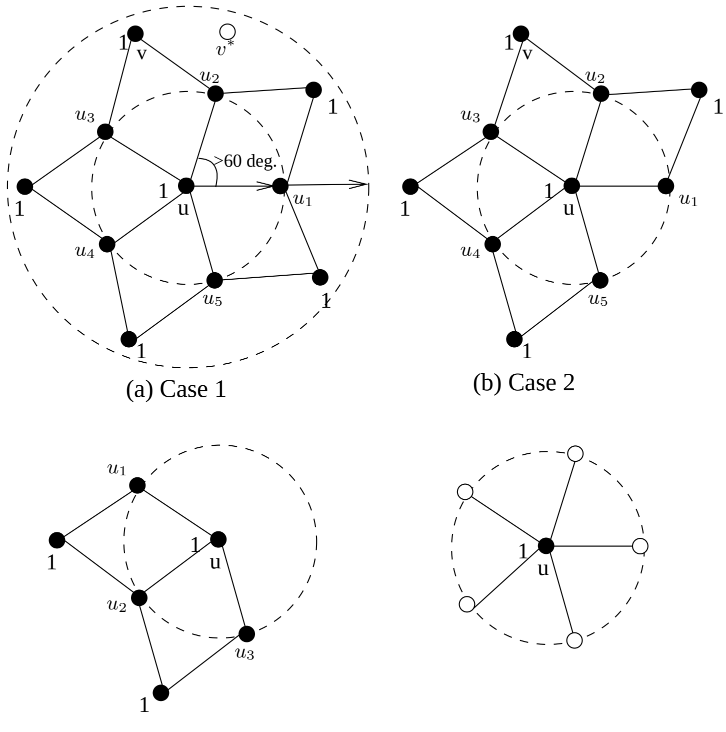
Case 3: At least two MCDS nodes (u, u_i) are required to cover at most three MIS nodes (Figure 4(c)).

Case 4: At least one MCDS node (u) is required to cover one MIS node (Figure 4(d)).

Analyzing the above relations we have the following:

Thus $|S| \le 5 * \lceil \frac{|MCDS|}{3} \rceil \le \lceil 1.66 * |\text{MCDS}| \rceil + 1 \le 1.66 * |\text{MCDS}| + 2$. This completes our proof. ∎

Theorem 5.3 *Let B be a CDS in G and S be any MIS in B. If B' denotes the set of additional nodes, then $|S \cup B'| \le 2.66 * |MCDS| + 2$.*



(a) Case 1 (b) Case 2 (c) Case 3 (d) Case 4

Figure 4. Proof of Lemma 5.2.

Proof Recall that for any orphan node u (u is grey) there must be at least a black node $v \in N(u)$ such that $CH(N(u)) \subseteq CH(N(v))$, otherwise u would not be grey. In order for the MCDS to cover u, either u must belong to an MCDS or there will be at least a node $v' \in \text{MCDS} \cap N(u)$. In any case an MCDS must have at least one node from $N[u]$. Since u selects the node with the minimum id from $N(u)$, the number of nodes in MCDS will not be greater than the number of nodes selected by all orphan nodes in the graph. Therefore, $|B'| \le |\text{MCDS}|$, where B' is the set of nodes selected by all orphan nodes in the graph.

Therefore, combining the results from Lemma 5.2 we obtain the following, $|S \cup B'| \le 2.66 * |\text{MCDS}| + 2$. ∎

Now we have $S \cup B'$ black nodes in G and all other nodes are dominated by them. Recall that $B \setminus (S \cup B')$ are colored grey. As $S \cup B'$ may not be a CDS, we find a set C of connectors to connect the nodes in S so that $S \cup B' \cup C$ becomes the desired CDS*.

In order to build the set C of connectors we adopt the following strategy: Initially, the set C is empty. Each node $b \in S$ finds shortest-distance paths to all neighbors $b' \in S$ ($b \ne b'$ and $id(b) > id(b')$) which are two and three hops away from b. If the length of a shortest path between b and b' is 2 and they are already connected via an intermediate node in B', then b does not choose any connector, else a connector is chosen. Ties are broken when there are more than one such connectors by selecting the one which has the smallest id. Similarly if they are three hops away and already connected by two (or one) nodes in B' then no (or one) connectors are sought. Otherwise, they select two connectors. All these connectors color themselves black. C denote the set of all such connectors.

In the following lemma, we prove that CDS$^* = S \cup B' \cup C$ in G forms a CDS.

Lemma 5.4 *The set CDS$^* = S \cup B' \cup C$ forms a CDS in G.*

Proof We first show that $S \cup B' \cup C$ is a dominating set in G and then prove that it is connected. By construction, $S \cup B'$ is a dominating set in G and so is $S \cup B' \cup C$.

Assume for the sake of contradiction that the subgraph G_s induced by the nodes in $S \cup B' \cup C$ is not connected. Consider two neighboring non-empty components G'_s and G''_s of G_s. There are two cases:

Case 1: There is no element of S in one of the components G'_s and G''_s. Without loss of generality assume G''_s is that component. Since G''_s is non-empty and does not have any element from S, the orphan nodes in G''_s must have selected some black nodes in B' to dominate them, otherwise G_s would not be a dominating set. If such a black node $u \in B', G''_s$ would not have a neighbor $v \in S, G'_s$, then u must belong to S, a contradiction.

Case 2: There is at least one node from S in each component $G's$ and G''_s. Find two nodes $s_1 \in S, G'_s$ and $s_2 \in S, G''_s$ such that the distance between them is at most three hops. Without loss of generality, let $id(s_1) > id(s_2)$. For each node $u \in S$ the algorithm selects at most two intermediate nodes to all other nodes in $v \in S$ which are at most 3 hops away and $id(u) > id(v)$. If s_1 and s_2 are not connected by any node in $B' \cup C$ then either their distance is more than three or their ids are the same, both cases yield a contradiction. ∎

We use the following lemma from [5] which bounds the size of C in G in terms of the cardinality of an MIS.

Theorem 5.5 *The cardinality of C is at most $18 * |S|$ [5].*

Theorem 5.6 *For any maximal independent set S in B, a set B' of additional nodes and a set C of connectors in a UDG $G = (V, E)$, we have $|S \cup B' \cup C| \leq 50.54 * |MCDS| + 38$.*

Proof Combining the results of Theorems 5.3 and 5.5 we get:

$$\begin{aligned} |C| &\leq 18 * |S| \\ &\leq 18 * |S \cup B'| \\ &\leq 18 * 2.66 * |MCDS| + 36 \\ &= 47.88 * |MCDS| + 36. \end{aligned}$$

Then,

$$\begin{aligned} |S \cup B' \cup C| &\leq |S \cup B'| + |C| \\ &\leq 2.66 * |MCDS| + 2 + 47.88 * |MCDS| + 36 \\ &= 50.54 * |MCDS| + 38. \end{aligned}$$

∎

One feature of our algorithm *A* is that it guarantees an MCDS (minimum CDS) for trees. The interesting thing is that we do not need to check whether G is a tree to compute an MCDS. This means that if G is a tree, algorithm *A* generates the optimal CDS. Other algorithms such as [2] require G to be checked for being a tree in order to construct a CDS. That is, the problem with their algorithm is that the nodes need to construct a spanning tree from the underlying graph which is an expensive operation in terms of energy usage.

We prove the optimality result in the following theorem.

Theorem 5.7 *If the underlying unit disk graph G is a tree with $|V| > 2$ algorithm A produces an MCDS.*

Proof It is well known that for a tree an MCDS consists only of the internal nodes of the tree. We prove the theorem by contradiction. Suppose a CDS produced by algorithm *A* consists of some leaves of the tree. Let ℓ be such a leaf and p its parent. Since ℓ is in the CDS, its computed convex hull $CH(N[\ell])$ is not included in any of its neighbor's convex hull and its color is black. However, its only neighbor is p and p's neighborhood contains at least two neighbors including ℓ, otherwise p is a leaf. Obviously, the convex hull $CH(N[\ell])$ is included in $CH(N[p])$, since $N[\ell]$ includes only ℓ and p. So ℓ would be colored grey and cannot be in the CDS, which is a contradiction. ∎

We analyze the performance of our algorithm in grids where we show that the cardinality of the CDS produced by algorithm *A* is less than twice the size of an MCDS.

Theorem 5.8 *Given a $m \times n$ grid, where nodes are placed at the intersection points and the Euclidean distance between two adjacent nodes equals 1, algorithm A produces a CDS such that $|CDS^*| \leq 2 * |MCDS|$.*

Proof By analyzing the $m \times n$ grid, the size of the minimum connected dominating set can be calculated as: $|\text{MCDS}| = \frac{m-1}{2} * n + \frac{m-1}{2} - 1 = \frac{mn-n+m-3}{2}$, when m is odd and $|\text{MCDS}| = \frac{m}{2} * n + \frac{m}{2} - 1 = \frac{mn+m-2}{2}$, when m is even. The cardinality of the CDS generated by algorithm *A* is mn. Thus if m is even the approximation factor is $2 \times \frac{mn}{mn+m-2} < 2$. However, if m is odd and $m > n$, the approximation factor is $2 \times \frac{mn}{mn-n+m-3} < 2$. ∎

Corollary 5.9 *For any induced subset of an $m \times n$ grid, algorithm A produces a CDS with at most twice as many nodes as an MCDS.*

6. Time and Message Complexities

According to our model $\mathcal{LOCAL}_S$, the time complexity of algorithm *A* is derived as follows. Each node u receives the location information of its neighbors and sends its location to them which can be done in a single round. Then u computes the convex hull $CH(N[u])$ and checks whether the convex hull is included in any of the convex hulls of its neighbors. However, the whole process of computing and comparing $CH(N[u])$ with $CH(N[v])$, $\forall v \in N(u)$, for inclusion is negligible in synchronous systems [15], where the time complexity is measured in terms of number of rounds. After computing $CH(N[u])$, u sends the list of vertices of $CH(N[u])$ to all its neighbors and receives theirs' as well. Since the message size is arbitrary, the exchange of convex hulls can be done in a single round. The computation of the distributed MIS [16] takes O($\log^* |B|$) time, where B is the set of black nodes. The distributed MIS algorithm in B generates the set S. Each node in S finds at most two intermediate nodes in S which are at most three hops away from it. However, the number of such neighbors is a constant [18] for a node $u \in S$. Thus computing such shortest paths takes O(1)-time. Therefore, the overall time complexity of the algorithm is O($\log^* |B|$)=O($\log^* n$).

To compute the message complexity, we see that each node has to send its location, id (all these can be packed in a single message) to its neighbors, which takes O(n) messages to be sent. Again, after computing the convex hull $CH(N[u])$, node u sends $CH(N[u])$ (the nodes in the convex hull are put in a message) to its neighbors which also takes O(n) messages to be exchanged in the network. However, to compute an MIS it takes O($|E| * \log^* n$) [16] messages. Thus the message complexity of the entire algorithm is O($|E| * \log^* n$).

7. Conclusion

Computing a CDS distributedly in WSNs is a very well studied problem. However, devising distributed algorithms for constructing such CDSs become more challenging when it is aimed to obtain a small constant approximation factor with each node knowing a limited amount of information about the topology of the underlying network. Several distributed algorithms are known in this context and interestingly almost all of them use the same approach: find an MIS in the network graph and then connect them through other nodes to construct the set.

Instead of finding an MIS in the whole network, our approach first builds a connected subgraph in the network graph, then finds an MIS in the subgraph and finally connects the nodes in the MIS through other nodes. By this technique, we could remove a number of unwanted nodes from being in the connected dominating set. This way our simple distributed algorithm guarantees a relatively small constant approximation factor which is a good improvement over the best known constant factor [5]. The main difficulty in lowering the constant factor lies in the reduction of the size of the connectors used to connect the nodes in the MIS. Exploring further geometric properties of unit disk graphs may help achieve a better bound for the size of the set of connector nodes. In future work, we look forward to finding distributed constant factor CDS algorithms in special classes of graphs, such as for example, planar graphs.

References

[1] K. Alzoubi, P.-J. Wan, and O. Frieder. Message-optimal connected-dominating-set construction for routing in mobile ad hoc networks. In *Proc. the Third ACM International Symposium on Mobile Ad Hoc Networking and Computing*, June 2000.

[2] K. 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*, 4(1):22–29, March 2002.

[3] K. Alzoubi, P.-J. Wan, and O. Frieder. New distributed algorithm for connected dominating set in wireless ad hoc networks. In *Proc. IEEE HICSS35*, January 2002.

[4] J. Blum, M. Ding, A. Thaeler, and X. Cheng. *Connected Dominating Set in Sensor Networks and MANETs, Handbook of Combinatorial Optimization (Editors)*, pages 329–369. Kluwer Academic Publisher, 2004.

[5] D. Chen, X. Mao, X. Fei, K. Xing, F. Liu, and M. Song. A convex-hull based algorithm to connect the maximal independent set in unit disk graphs. *Wireless Algorithms, Systems, and Applications*, pages 363–370, 2006.

[6] X. Cheng, M. Ding, D. Du, and X. Jia. On the construction of connected dominating set in wireless networks. *Wireless communications and Mobile Computing*, 6:183–190, 2006.

[7] X. Cheng, X. Huang, D. Li, and D. Du. Polynomial-time approximation scheme for minimum connected dominating set in ad hoc wireless networks. *Networks*, 42(4):202–208, June 2003.

[8] B. Clark, C. Colborn, and D. Johnson. Unit disk graphs. *Discrete Mathematics*, 86:165–177, March 1990.

[9] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press and McGraw-Hill, 2001.

[10] B. Das and G. Bharghaban. Routing in adhoc networks using minimum connected dominating set. In *Proc. IEEE International Conference for Communication*, 1997.

[11] S. Funke, A. Kesselman, U. Meyer, and M. Segal. A simple improved distributed algorithm for minimum connected dominating set in unit disk graphs. *ACM Transactions on Sensor Networks*, 2(3):444–453, August 2006.

[12] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, pages 374–387, 1998.

[13] Y. Li, S. Zhu, M. Thai, and D. Du. Localized construction of connected dominating set in wireless networks. In *Proc. TAWN*, June 2004.

[14] M. Marathe, H. Breu, H. Hunt, S. Ravi, and D. Rosenkrantz. Simple heuristics for unit disk graphs. *Networks*, pages 59–68, 1995.

[15] D. Peleg. *Distributed Computing: A Locality-Sensitive Approach*. SIAM, Philadelphia, PA, second edition, 2000.

[16] J. Schneider and R. Wattenhofer. A log-star distributed maximal independent set algorithm for growth-bounded graphs. In *Proc. PODC*, 2008.

[17] I. Stojmenovic, M. Seddigh, and J. Zunic. Dominating sets and neighbor elimination based routing algorithms in wireless sensor networks. *IEEE Transaction Parallel Distributed Systems*, 13(1):14–25, 2002.

[18] P. Wan, K. Alzoubi, and O. Frieder. Distributed construction of connected dominating set in wireless adhoc networks. In *Proc. INFOCOM*, 2002.

[19] J. Wu and H. Li. On calculating connected dominating sets for efficient routing in ad hoc wireless networks. In *Proc. Discrete Algorithms and Methods for Mobile Computing*, volume 3, pages 7–14, 1999.