An Efficient Distributed Node Clustering Protocol for High Dimensional Large-Scale Wireless Sensor Networks[*]

Xudong Zhu
Dept. of Computer Science
Shanghai Jiao Tong University,
Shanghai, China
xudongzhu42@gmail.com

Jun Li
Dept. of Computer Science
Shanghai Jiao Tong University,
Shanghai, China
lijun2009@sjtu.edu.cn

Yuanfang Xia
Dept. of Computer Science
Shanghai Jiao Tong University,
Shanghai, China
yuanfang3.xia@gmail.com

Xiaofeng Gao[†]
Dept. of Computer Science
Shanghai Jiao Tong University,
Shanghai, China
gao-xf@cs.sjtu.edu.cn

Guihai Chen
Dept. of Computer Science
Shanghai Jiao Tong University,
Shanghai, China
gchen@cs.sjtu.edu.cn

ABSTRACT

In the past few yeas, wireless sensor networks (WSNs) have been widely used in many areas. In these applications, sensors are remotely deployed to gather related environmental information for further analysis. To support higher scalability and better data aggregation, sensor nodes are often grouped into disjoint and mostly non-overlapping clusters. All nodes in a cluster can send their data to the cluster head within d-hop distance, and the head should communicate with other cluster heads and pass all data to base station. For better communication between these cluster heads, lower maintenance cost and easier management, it is necessary to make the number of the clusters as small as possible. Moreover, in many environments like mountainous area or underwater monitoring, node deployment is often not flat, resulting in a high dimensional network. In this paper, we focus on proposing a scheme to select cluster heads for a homogenous network in three-dimension situation. The scheme meets two requirements: The number of cluster heads is minimum; and the head nodes can communicate with each other. These requirements can be formed as an NP-complete problem named *d-hop connected dominating set.* Correspondingly, we proposed a distributed approximation algorithm, and proved its approximation ratio as $(d+1)\beta$, where β is a calculated parameter w.r.t. d. We also analyzed the performance of our algorithm with corresponding numerical experiments.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

General Terms

Algorithms, Management, Performance, Theory

Keywords

Wireless Sensor Network, d-CDS, Spanning Tree, Cluster, Homogeneous Network, Distributed Algorithm

1. INTRODUCTION

Wireless sensor networks (WSNs) are autonomous and self-organized communication systems consisting of many small, inexpensive, and battery-powered embedded devices called sensor nodes, each with sensing, computing, communication capabilities. These sensor nodes serve not only as mobile hosts but also as routers. Because of such characteristics, WSNs can be widely used in lots of applications such as disaster management, battlefield reconnaissance, border patrol and surveillance, etc. In these applications, sensor nodes are mainly used to gather vital information from the surrounding areas such as temperature, sound, vibration, pressure, motion or pollutants, etc. Also, sensor nodes need to transmit their collected data to base stations directly or indirectly. The related researches have been widely studied in past decades [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11].

In wireless sensor networks, sensor nodes have constraint in terms of processing power. In most case, the unattended nature of WSNs makes it quite difficult to recharge node batteries or replace sensor nodes. Therefore, energy conservation is a major design goal in these networks. To prolong the lifetime of WSNs, a considerable number of researches have been done [2, 3, 4, 6, 8, 12]. In addition, sensor nodes are limited by communication bandwidth and storage space, which makes it difficult to transmit message throughout the whole network and to process a mass of data. In order to

[*]This work has been supported in part by the National Natural Science Foundation of China (Grant number 61202024), Shanghai Pujiang Program (No.13PJ1403900), the Natural Science Foundation of Shanghai (Grant No.12ZR1445000), and Shanghai Educational Development Foundation (Chenguang Grant No.12CG09).

[†]X.Gao is the corresponding author.

IMCOM (ICUIMC) '14, January 9-11, 2014, Siem Reap, Cambodia.
Copyright 2014 ACM 978-1-4503-2644-5 ...$15.00.

overcome such shortcomings, the efficient approach that is commonly agreed by most of researchers is clustering. We can divide the whole network into disjoint and mostly non-overlapping sets called clusters. Each cluster elects a leader called cluster head. All ordinary nodes in a cluster transmits their data to the cluster head, and the head passes all data to the base station. Additionally, each cluster header can communicate with other cluster heads. Clustering schemes offer reduced communication overheads and efficient resource allocations, thus decreasing the overall energy consumption and reducing interferences among sensor nodes.

With clustering in WSNs, energy consumption, lifetime and scalability of the network can be greatly improved. Because all the ordinary nodes transmit their data to the cluster, a lot of energy can be saved by absence of flooding and multiple routes. Furthermore, any changes within a cluster effects only this cluster locally, so clustering can ensure efficient resource allocation, with greater network scalability. To make the best use of clustering, one of the most important task is to partition a given network into disjoint and mostly non-overlapping groups. Numbers of clustering algorithms were proposed according to this idea. In [5], the authors proposed a clustering algorithm which based on cell combination for the networks. Based on the fact that we can improve the system's lifetime by improving each cluster's lifetime, [6] alternates cluster head in a cluster to maximize the cluster's lifetime and finally determines each cluster and the corresponding head. In [7], through coordination of nodes belonging to the same cluster which can effectively avoid redundant sensing or processing, Alaei et.al proposed a clustering method to optimize the energy conservation and prolong network lifetime. Other related clustering algorithms can refer to [8, 12, 13].

Easily to know, small size clusters leads to a large number of clusters, which will congest the area; while a few number of clusters will exhaust the cluster head with large amounts of messages transmitted from cluster members. An effective way to control the average size of clusters is to assume that each cluster head can be at most d-hop away from the nodes within its dominating range, where d can be manually set based on the real circumstance. When sensor nodes in a network are uniformly and independently distributed, [9] gives a solution to evaluate the average hop distance. In this paper, we focus on constructing d-hop clusters for a given WSN. Usually, the set of cluster heads can be viewed as a *connected dominating set* (CDS) of graph G, which is abstracted from the corresponding network. For a given graph $G = (V, E)$, a CDS of G is a subset $C \subseteq V$ such that for each vertex $v \in V \backslash C$, there exists a vertex $u \in C$ satisfying $(u, v) \in E$. Moreover, the subgraph induced by C is connected. Similarly, to construct d-hop clusters for a given WSN is equivalent to choose a d-CDS for a given graph. The formal definition of d-CDS is: For a given graph $G = (V, E)$, a d-CDS of G is a subset $C \subseteq V$ such that for each vertex $v \in V \backslash C$, there exists a vertex $u \in C$ and a path p from v to u satisfying the length of p is at most d. Moreover, the subgraph induced by C is connected. To reduce the message redundancy, finding a minimum d-CDS is of great significance.

In most cases, people assume that wireless nodes are on a two-dimensional plane, and use a *unit disk graph* (UDG) to model the network, where each node has the same communication range and two nodes can communicate with each other only when they are located within the communication range of the other. However, in many environment like mountainous areas or underwater region, node deployment is often not flat, resulting in a three dimensional network. Correspondingly, we can use an *unit ball graph* (UBG) to model such network in a 3D space. In an UBG $G = (V, E)$, each node also has the same communication range, denoted as a ball, and any two vertices are adjacent if and only if the Euclidean distance between them is at most 1. As far as we know, there is not yet specialized works on d-CDS in unit ball graph before this paper. The only related work is the work done by Kim [14] which can be seen as 1-CDS example in UBG.

Our goal in this paper is to partition a wireless sensor network into d-hop clusters by finding a minimum d-CDS for an UBG $G = (V, E)$ derived from the given network. In this paper, we proposed a two-phase distributed algorithm to find a minimum d-CDS. The first part of this algorithm is to select a d-MIS from a given network. Afterwards, we add some extra nodes to connect the d-MIS to make it a d-CDS.

Our contributions are threefold: (1) As far as we know, this is the first work focusing on finding a minimum d-CDS in three-dimensional situation. (2) Since distributed algorithms become more and more important for self-organized WSNs, we proposed a distributed algorithm for minimum d-CDS problem. As for the details of our algorithm, we first introduce concepts of "level id" and "neighbor information" to overcome some bug in design similar distributed algorithms. (3) As the minimum d-CDS problem is NP-complete which is proved by Vuong and Huynh [15], we analyzed the performance of our algorithm and provided approximation ratio for our algorithm. Moreover, the approximation ratio is proved to be $(d + 1)\beta$, where β is a parameter given in Section 5.

The rest of this paper is organized in the following structure: In Section 2, we describe the related work in detail. Section 3 introduces some definitions and notations that will be used in later sections. In Section 4, we present our four-phase distributed algorithm to select a d-CDS for the given graph. Afterwards, the corresponding performance analysis is described in Section 5 and Section 6 gives the simulation. Finally, Section 7 concludes this paper.

2. RELATED WORK

Since our goal is to find a minimum d-CDS for a given graph $G = (V, E)$, we will introduce some related work about CDS in this section. Since CDS is commonly used to construct virtual backbone for WSNs, lots of efforts have been made in the past decades. In two-dimensional space, researchers often use unit disk graph (UDG) to model wireless sensor networks. In an UDG $G = (V, E)$, any two vertices are adjacent if and only if the Euclidean distance between them is at most 1. In order to improve the performance of WSNs, we usually choose a minimum CDS (MCDS) to act as the networks' virtual backbone.

Clark et.al. [16] proved that the MCDS problem is NP-hard even in UDG. Hence, a commonly used approach is to use approximation algorithms to solve such problem. In 2002, Wan et.al [17] first found that MCDS problem has polynomial-time constant-factor approximation solution and proposed a two-phase algorithm to select a CDS. Based on this design, a lot of similar two-phase algorithms were proposed in literature. All these algorithms first choose a maximal independent set (MIS), and the second phase is to add some extra nodes to connect them. An MIS for graph $G = (V, E)$ is a subset $M \subseteq V$ such that any two vertices in M are not directly connected, and if we insert a node u from $V \backslash M$ into M, M will not be an MIS any more. Obviously, MIS is also a dominating set (DS). Besides, the related theoretical approximation ratio of such algorithms are also been widely studied [18]. On the other side, Cheng et.al [19] first found that MCDS problem has PTAS solutions. Even though PTAS has a better approximation ratio, the time cost in [19] is too high to implement in reality. Hence, the main focus of this field is still on constructing effective approximation algorithms to find a feasible solution within polynomial time. Except for those maximal independent set-based algorithms, there also exist other kinds of algorithms, such as greedy algorithms [20], Steiner tree-based algorithms [10], pruning-based algorithms [21] and connected clustering-based algorithms [22].

As for the two-phase algorithm, it is very common to apply color-marking algorithm to first select an MIS from a given graph $G = (V, E)$ [17]. After we have obtained a DS, connecting this DS into a CDS is equivalent to finding a Steiner Tree for G. The formal definition of Stenner Tree is as follows: Given a graph $G = (V, E)$, a selected subset S, a Steiner Tree is a tree in G includes all vertices in S. In those MIS-based algorithms, Steiner Tree is commonly used in the connecting part. The detailed introduction about Steiner Tree used in constructing MCDS can refer to [10].

In a wireless sensor network, implementing cluster-based hierarchical structure is much more helpful to achieve efficient routing, increase the lifetime of networks and improve the network's scalability. Further, such structure can be modeled as d-hop CDS. Most of related works on d-hop CDS were finished within recent decades. In 2006, Huynh et.al [23] first proved that finding a minimum d-hop CDS in UDG is NP-complete. For a period of time since then, people could only propose some heuristic algorithms which has no exact performance analysis, especially the quantitative description about the gap between optimal solution and feasible solution. Until recently, some theoretical researches came out. In [11], Gao et.al proposed a two-phase approximation algorithm to compute a d-hop CDS in a UDG with a constant-factor approximation which is relevant with $O(d^3)$. Later, Zhang et.al [24] improved the approximation ratio into $O(d^2)$ level.

Obviously, UBG can formulate a network environment more precisely than UDG because UBG model can reflect more detail of real world. Although the design of algorithms for MCDS in UBG seems to be similar with the design in UDG, the analysis part of these approximation algorithms could be much harder than that in UDG. Because of such difficulty, few papers study MCDS approximation in 3D space to the best of our knowledge. Actually, in most case, when people begin to study MCDS in UBG, they usually modify and generalize the corresponding approximation algorithms in 2D space, such as the two-phase algorithms mentioned above. In [25], Butenko and Ursulenko first proved that the ratio between the size of MIS and MCDS is at most 11 which leads to an approximation ratio of 22 for MCDS in UBG. Later, Kim [14] improved that ratio into 14.937.

As far as we know, there is few work studying minimum d-hop CDS in UBG. This paper is the first to study the relevant research.

3. PRELIMINARIES

In this section we will introduce some definitions and notations, which will be used later in our algorithms and analysis.

As mentioned before, for any given graph, we hope to find a minimum d-CDS to gather data and cluster the network. When designing algorithms for CDS, many researchers adopt a two-step algorithm as follows:

1. Construct a maximal independent set (MIS).
2. Connect this MIS into a CDS.

We follow a similar pattern in the design of our d-CDS algorithm. We first select a d-MIS, then connect this d-MIS into a d-CDS. The following are the definition of d-MIS, d-DS and the relation between them.

DEFINITION 1. *A d-IS (independent set) for a given graph $G = (V, E)$ is a vertex set I such that for any pair of nodes in I, the distance between them is greater than or equal to d hops.*

DEFINITION 2. *A d-MIS (maximal independent set) for a graph $G = (V, E)$ is a d-IS such that if we insert any vertex from $V \setminus I$, I is no longer a d-IS.*

By definition, it is easy to get the following lemma.

LEMMA 1. *A d-MIS for a graph G is also a d-DS for G.*

In the following sections, the distance between any two nodes u, v always means the smallest number of hops needed from u to v, and we use $dis(u, v)$ to denote this distance. For any node u, we use d-hop neighbors for u to denote the set of nodes within d hops from u (except itself), i.e.,

$$N^d(u) = \{v \in V \mid 1 \leq dis(u, v) \leq d\}.$$

4. A DISTRIBUTED ALGORITHM

Our distributed algorithm has two subroutines, which is a generalization of the algorithm described in [26, 11]. Based on a spanning tree, we first construct a d-MIS, then insert additional nodes to connect the selected nodes as a d-CDS. Specifically, all previous works failed to clarify the details of message interchange processes in their distributed designs, which might easily bring deadlock and termination problems. Our algorithm design avoids such problems, which can be implemented in any asynchronous systems.

4.1 Algorithm Description

Before entering our algorithm, there are some preparations to be finished. Given a UBG $G = (V, E)$, we select an arbitray root r(usually with maximum node degree and locates at the center of the network), and then use the distributed leader-election algorithm mentioned in [27] to construct a spanning tree. In the meanwhile, we calculate the *level* of each node, which is the number of hops from root to this node. As for the root r, $level = 0$. Then we can give a rank to every node by using the pair of its *level* and its *id*, $(level, id)$. For any two node n_x, n_y, suppose $n_x.level$, $n_y.level$ is their *level* information, $n_x.id$, $n_y.id$ is their *id* information, then n_x has a lower rank than n_y if and only if one of the following conditions holds:

1. $n_x.level < n_y.level$; or

2. $n_x.level = n_y.level$ and $n_x.id < n_y.id$.

Additionally, each n_i has a variable $n_i.children$ for counting its children. It also uses a set $n_i.nb$ to record its d-hop neighbors. Each entry of nb is the in the form of $(ID_u, level_u)$. Table 1 summarizes the variables used for each node n_i, some of which will be defined later.

Table 1: Variables for each node n_i

Name	Explanation
level	number of hops from root to n_i.
id	the ordering number of n_i.
parent	n_i's parent in tree.
children	the number of n_i's children.
nb	the set of n_i's d-hop neighbors.
color	n_i's color.
blackpath	the path from one black node to another black node which actives it.

Subroutine 1 is a coloring process to select a d-DS for the given network. All the nodes are initially colored white, and will be colored black or grey eventually. We use three special variable $WHITE$, $BLACK$, $GREY$ to denote these three colors. When a node u colors itself black, it will broadcast a *black* message, which will reach all nodes in $N^{d+1}(u)$, and each node v in $N^d(u)$ will color itself grey and broadcast a *grey* message which will also reach all nodes in $N^d(v)$. Each node in $N^{d+1}(u) \setminus N^d(u)$ will record the path to u in *blackpath* if it is still white and its *blackpath* has not been determined. Once a node knows that all its lower rank d-hop neighbors have been colored grey already, it will color itself black and broadcast a *black* message. All the black nodes will consists of a d-MIS for G after this subroutine. The detailed description is shown in Alg. 1. We use $n_i.x$ to count the unterminated children.

In subroutine 2, we will connect d-DS into a d-CDS. For each black node except the root, we color the nodes in its *blackpath* as black to connect it with another black node. The function *pop*(*blackpath*) means to take out the last entry of *blackpath*. After this subroutine, all the black nodes will consists of a d-CDS for G. The detailed description is shown in Alg. 2.

The two subroutines form our algorithm, and we will refer to this algorithm as d-CDS algorithm in the following section. Alternatively, instead of connecting each black dominator directly via *blackpath*, we may use distributed Steiner tree algorithm to connect black nodes (denoted as Steiner nodes), which could improve the final approximation ratio.

Algorithm 1 d-MIS coloring

```
$n_i.color = WHITE$;                                   ▷ Initialization
$n_i.x = n_i.children$;
if $n_i$ is root then
    broadcast $black(\{n_i.id\})$;
end if
if receive $black(path)$ then
    if $length(path) \leq d$ then
        push $n_i.id$ to $path$;
        broadcast $black(path)$;
        if $n_i.color = WHITE$ then
            $n_i.color = GREY$;
            broadcast $grey(n_i.id, d)$;
        end if
    else if $length(path) = d + 1$ & $n_i.color = WHITE$
    & $n_i.blackpath$ is not set then
        $n_i.blackpath = path$;
    end if
end if
if receive $grey(n_j.id, k)$ & $k > 1$ then
    mark $n_j.id$ as colored in $n_i.nb$;
    broadcast $grey(n_j.id, k - 1)$;
end if
if all lower rank $d$-hop neighbors are colored grey &
$n_i.color = WHITE$ then
    $n_i.color = BLACK$;
    broadcast $black(\{n_i.id\})$;
end if
if $n_i.children = 0$ & $n_i.color \neq WHITE$ then
    broadcast $colored(n_i.parent)$; terminate;
end if
if receive $colored(n_j.parent)$ & $n_i.id = n_j.parent$ then
    $n_i.x = n_i.x - 1$;
    if $n_i.x = 0$ then
        if $n_i$ is not root then
            broadcast $colored(n_i.parent)$;
        end if
        terminate;
    end if
end if
```

Algorithm 2 d-CDS connecting

```
if $n_i.color = BLACK$ then
    $nexthop = pop(n_i.blackpath)$;
    broadcast $join(nexthop, n_i.blackpath)$;
end if
if receive $join(nexthop, blackpath)$ then
    if $n_i.id = nexthop$ and $blackpath \neq \varnothing$ then
        $nexthop = pop(blackpath)$;
        broadcast $join(nexthop, blackpath)$;
        $n_i.color = BLACK$;
    end if
end if
```

4.2 An Example

In this subsection, we use an example to illustrate our algorithm. To keep it simple and precise, we just plot the example in two-dimensional space as is shown in Fig. 1.

Originally, we have a given UBG $G = (V, E)$ with 17 nodes, as is shown in Fig. 1 (a). We want to find a 2-hop CDS for G. Before all of the work starts, we need to construct a spanning tree T for G. As Fig. 1 (b) shows, a spanning tree is formed according to the original graph and every node has a unique id number. A solid line between two nodes means there is an edge between them in the spanning tree T, and a dashed line between two nodes means there is an edge between them in the original graph G, but this edge is not included in T. Before actually processing coloring subroutine, each node should have already got its level and d-hop neighbor information. Then we can start to color the network.

In the coloring procedure as is shown in Alg. 1, initially all nodes are white. In Fig. 1 (b), root r colors itself black, and broadcasts a *black* message. In Fig. 1 (c), upon receiving a *black* message and detecting that there is a black node within its 2-hop distance, node 2, 3, 4, 5, 6 color themselves grey, and broadcast a *grey* message to inform all their d-hop neighbors that they have already been colored. At this time, node 8 notices that all its lower rank 2-hop neighbors have been colored grey (they are node 2, 4, 5, respectively), so node 8 colors itself black and broadcasts a *black* message. In Fig. 1 (d), node 9, 10, 13, 14, and 15 receive the *black* message initiated from node 8, and thus color themselves as grey and broadcast *grey* messages. Now node 11 can color itself black and broadcast a *black* message, which will lead node 12, 16, and 17 to color themselves as grey in Fig. 1 (e). Finally, the set of node {1, 8, 11} is our selected 2-MIS for G, also a 2-DS for G.

Fig. 1 (f) shows a possible way to connect these black nodes into a 2-CDS, since both node 8 and node 11 record the path from node 1 as their blackpaths. As a result, the set of nodes {1, 2, 3, 4, 6, 8, 11} is our selected 2-CDS.

5. PERFORMANCE ANALYSIS

In this section we will analyze the performance of our d-hop CDS algorithm. We first prove the correctness of this design, and then discuss its approximation ratio.

Theorem 1. *d-CDS algorithm can successfully terminate with consistent agreement.*

Proof. First, we prove the termination for the d-CDS algorithm above. According Line 1-25 in Alg. 1, after enough time, all nodes will be colored either black or grey. From Line 25-26, leaf nodes will first terminate. Then leaf nodes will feed the messages of termination to their parents. Their parents will also terminate when all their children terminate according to Line 29-37. As for Alg. 2, the entire network will terminate when all blackpaths pop out.

Next, we will prove the property of agreement. Since each message in d-CDS algorithm except for Alg. 1 is broadcasted only once by each node, it is easy to understand the agreement in this situation. Besides, in Alg. 1, Line 22 ensures the sequence of the chosen black nodes is deterministic. Hence, the algorithm must meet the requirement of agreement. □

Theorem 2. *All the black nodes after Alg. 1 consists of a d-MIS for G.*

Proof. We noticed that every node must be colored as black or grey after Alg. 1, otherwise Alg. 1 cannot terminate.

Then we prove that the distance between any two black nodes is greater than or equal to d. We prove it by contradiction. Suppose there are two black nodes u and v with distance less than d. Because all the nodes are totally ordered by rank, without loss of generality we assume u has a higher rank than v. Then u cannot color itself unless it knows that v has been already colored. However, since the distance between u and v is less than d, in all *black* messages initiated from v and reached u, there must be one traveling less than d hops. Then by Alg. 1, u will be colored grey, which yields a contradiction.

We also cannot insert any more black nodes because if we change any grey nodes into black, there must exist a black node within d hops.

Thus all the black nodes form a d-MIS for G. □

Theorem 3. *All the black nodes after Alg. 2 consists of a d-CDS for G.*

Proof. Denote the set of black nodes after Alg. 2 as C. According to Lemma 1 and Lemma 2, the set of black nodes after Alg. 1 is a d-MIS, so it is a d-DS for G by Lemma 1. Hence we only need to prove the connectivity of C. In Alg. 2, we use the nodes in *blackpath* to connect each black node(except the root) to another black node, the connectivity just follows. □

Then we move on to the discussion of the approximation ratio. First we find an upper bound for the number of independent vertices for in a node's d-hop neighbors. Then we use this upper bound to establish a connection between IS and DS for a graph. Finally, we prove the approximation ratio of our algorithm. During our derivation, we use a lemma proved in [28], which is described in Lemma 2.

Lemma 2. *([28]) For any vertex u in a UBG G, the neighborhood $N(u)$ contains at most 12 independent vertices.*

Lemma 3. *I is a d-IS of a UBG G, then for any vertex u, $N^d(u)$ contains at most β vertices from I, where*

$$\beta = \begin{cases} 12 & \text{if } d = 1, \\ 125 & \text{if } d = 2, \\ 12 + \dfrac{8d^3 + 12d^2 + 6d}{\left\lceil \frac{1}{2} \left\lfloor \frac{d-1}{2} \right\rfloor \right\rceil} & \text{if } d \geq 3. \end{cases}$$

Proof. It is easy to see that when $d = 1$, the result is valid by Lemma 2. For $d = 2$, if we place a ball centered at z with radius 0.5 for each vertex z from I, then all these balls are mutually disjoint. Next, based on the knowledge that node u's any 2-hop neighbor (as a ball with radius 0.5) should located within the ball centered at u with radius 2.5, we can calculate the upper bound of β for $d = 2$ as

$$\beta \leq \frac{\frac{4}{3}\pi \cdot 2.5^3}{\frac{4}{3}\pi \cdot 0.5^3} = 125.$$

Now suppose $d \geq 3$. For any $u \in V$, let

$$A = N^{\lceil d/2 \rceil}(u) \cap I = \{a_1, a_2, ..., a_t\}$$

denote the set of u's d-hop independent neighbor within $\lfloor \frac{d}{2} \rfloor$ hops, and we will show that $t \leq 12$.

Suppose $t > 12$, then for any two vertices w, $v \in A$, denote the shortest path from w to u and from v to u as P_w and P_v, and denote w_0, v_0 as the last vertices on P_w and P_v. Because $N^1(u)$ contains at most 12 independent vertices, if $t > 12$, we could always choose two vertices w and v such that w_0 and v_0 are within each other's transmission range (could be the same vertex). Then we could construct a path from w to v by travel in this order: w, w_0, v_0, v, and the length of this path is at most $2\lceil d/2 \rceil - 1 \leq d$, contradicting with the fact that w, v are d-hop independent. Thus

$$\left| N^{\lceil d/2 \rceil}(u) \cap I \right| = t \leq 12. \quad (1)$$

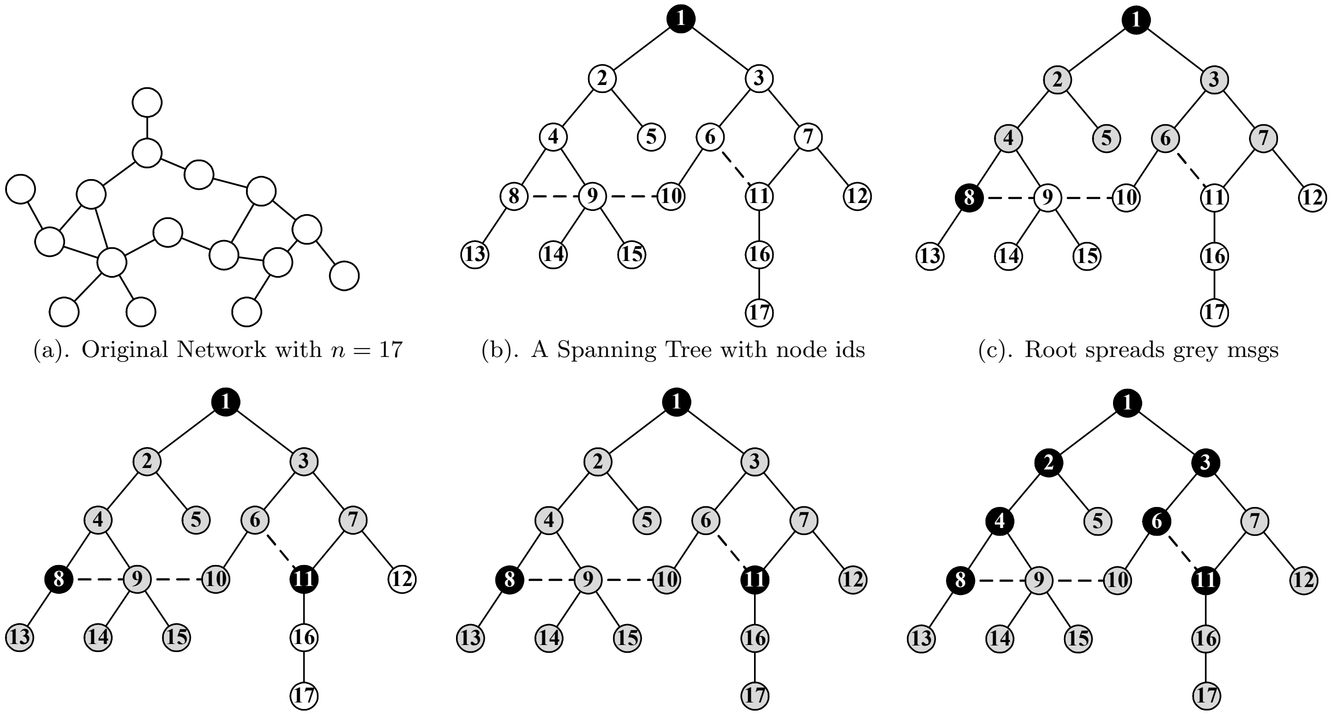
Then consider the rest part of u's d-hop neighbors. Let

$$B = N^d(u) \setminus N^{\lceil d/2 \rceil}(u) \cap I = \{b_1, b_2, ..., b_m\}.$$

For each $1 \leq i \leq m$, let Q_i be a shortest path from b_i to u. Denote D_b as a ball centered at b with radius 0.5 and define a region C_i such that

$$C_i = \bigcup_{b \in N^{\lfloor (d-1)/2 \rfloor}(b_i) \cap V(Q_i)} D_b.$$

Here $V(Q_i)$ is the set of nodes on path Q_i. We then claim that C_i cannot intersect with C_j for any $i \neq j$. If it happens, then we can construct a path between b_i and b_j, which has



(a). Original Network with $n = 17$ (b). A Spanning Tree with node ids (c). Root spreads grey msgs

(d). New nodes are colored as black (e). Continue coloring to form a 2-DS (f). Connect 2-DS as a 2-CDS

Figure 1: An example to construct a 2-CDS with 17 nodes

length at most $2\lfloor (d-1)/2 \rfloor + 1 \le d$, contradicting the fact that b_i, b_j are d-hop independent. Thus we have that C_i does not intersect with each other.

Suppose $Q_i = w_1 w_2 w_3 \ldots$. Since Q_i is a shortest path between b_i and u, then $D_{w_1}, D_{w_3}, \ldots$ are disjoint with each other, and the volume of C_i is at least $\lceil \frac{1}{2} \lfloor \frac{d-1}{2} \rfloor \rceil \frac{4}{3}\pi \frac{1}{2^3}$.

Next, notice that $b_i \notin N^{\lceil d/2 \rceil}$, so the distance between u and b_i is greater than $\lceil \frac{d}{2} \rceil$, i.e. $dis(u, b_i) > \lceil \frac{d}{2} \rceil$. For $b \in N^{\lfloor (d-1)/2 \rfloor}(b_i) \cap V(Q_i)$, according to the triangle inequality, we have

$$\begin{aligned} dis(u,b) &\geq dis(u,b_i) - dis(b,b_i) \\ &> \lceil d/2 \rceil - \lfloor (d-1)/2 \rfloor = 1, \end{aligned}$$

which means C_i does not intersect with the ball at center u with radius 0.5. Since every C_i locates within the ball at center u with radius d+0.5, we have

$$m \leq \frac{\frac{4}{3}\pi(d+\frac{1}{2})^3 - \frac{4}{3}\pi\frac{1}{2^3}}{\left\lceil \frac{1}{2} \left\lfloor \frac{d-1}{2} \right\rfloor \right\rceil \frac{4}{3}\pi\frac{1}{2^3}} = \frac{8d^3 + 12d^2 + 6d}{\lceil \frac{1}{2} \lfloor \frac{d-1}{2} \rfloor \rceil} \tag{2}$$

From Eqn. (1), and Eqn. (2) we can get our final result. □

Lemma 4. *Suppose D is a d-DS of G and I is a d-IS of G. Then $|I \setminus D| \leq \beta |D \setminus I|$.*

Proof. Denote $X = I \setminus D$ and $Y = D \setminus I$. Construct a bipartite graph $H = (X, Y, E)$. Here $(x, y) \in E$ if and only if $x \in X$, $y \in Y$, and $dis(x, y) \leq d$. It is easy to see

$$\sum_{x \in X} deg_H(x) = \sum_{y \in Y} deg_H(y), \tag{3}$$

where $deg_H(x)$ represents the degree of x in the bipartite graph H. Since any vertex in X is d-hop dominated by some vertices, so for any $x \in X$,

$$deg_H(x) \geq 1. \tag{4}$$

Next, By Lemma 3, for any $y \in Y$,

$$deg_H(y) \leq \beta. \tag{5}$$

Then our lemma follows from Eq. (3), (4), and (5). □

Lemma 5. *Suppose I is a d-MIS for G, then we need at most $d(|I| - 1)$ nodes to connect I in Alg. 1.*

Proof. In Alg. 1, for every black node except the root, there exists a black node at d hops away, just follow the *blackpath*. Thus, the total nodes needed to connect I is at most $d(|I| - 1)$. □

Next we prove our main result.

Theorem 4. *Our d-CDS algorithm has an approximation ratio of $(d+1)\beta$.*

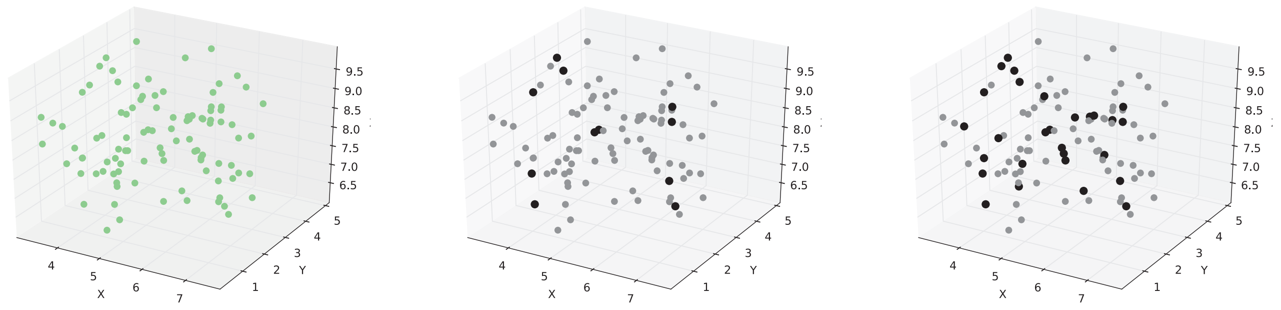
Proof. Denote the set of black nodes after Alg. 1 as I, and denote the set of all black nodes after Alg. 2 as C. Let C^* be the optimal solution of d-hop CDS in G. By Lemma 4,

$$\begin{aligned} |I \setminus C^*| &\leq \beta |C^* \setminus I| \\ |I| - |I \cap C^*| &\leq \beta |C^*| - \beta |C^* \cap I| \\ |I| &\leq \beta |C^*| - (\beta - 1)|C^* \cap I| \end{aligned} \tag{6}$$

Also by Lemma 5, we have

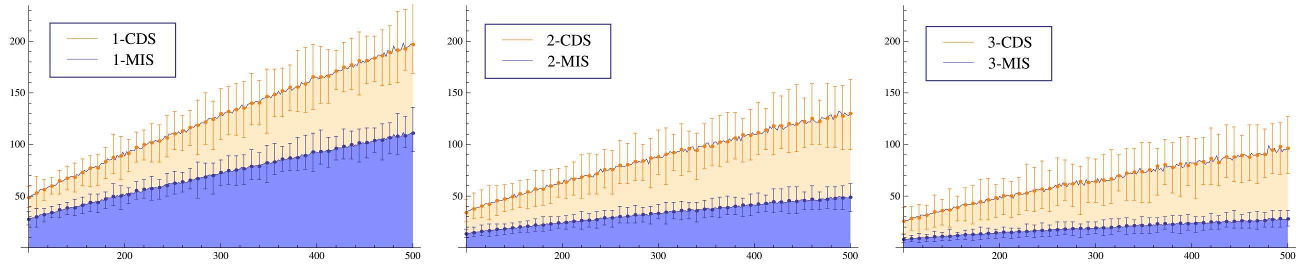
$$\begin{aligned} |C| &\leq d(|I| - 1) + |I| \\ &\leq (d+1)|I| - d \\ &\leq (d+1)\beta |C^*| \end{aligned}$$

Then the theorem holds. □



(a). Original Network with $n = 100$ (b). A 3-DS with 11 nodes (c). A 3-CDS with 28 nodes

Figure 2: An example to illustrate the procedure of our algorithm.



(a). MIS and CDS size w.r.t. n (b). 2-MIS and 2-CDS size w.r.t. n (c). 3-MIS and 3-CDS size w.r.t. n

Figure 3: An example to illustrate d-MIS and d-CDS size w.r.t. network size.

6. SIMULATION AND EVALUATION

In this section, we first present our experiment settings, then we show our evaluation results with corresponding analysis.

6.1 Experiment Settings

To evaluate the performance of our algorithm, we randomly generate different sets of nodes (with different node numbers and coordinates) on a space of size $100 \times 100 \times 100$ units. For each fixed hop setting ($d = 1$, $d = 2$, $d = 3$), we repeatedly run the algorithm for 50 different cases to achieve an average performance. We focus on the following impacts to evaluate the performance of our algorithm under different network sizes and different hop settings.

- The size of chosen CDS versus the size of the network, which is a direct reflection of algorithm performance. A smaller CDS is easier to maintain and has lower energy cost, bring benefits to the service providers.
- The d-MIS size versus the d-CDS size. This parameter reflects the construction of two-step d-CDS algorithm, and exposes the cost in each step.
- The hop count of chosen CDS, reflecting the concentration level of CDS. If d is smaller, then more nodes need to be selected into d-MIS, while if d is larger, then more nodes are needed to connect two cluster heads.

The various parameters used in our simulation are tabulated in Table 2.

6.2 Experiment Results

In this part, we will provide some numerical results on the distributed algorithm proposed above. First, Fig. 2 (a)-(c) shows the detailed procedure of how our algorithm successfully obtains a d-CDS. In this exhibition, we set the biggest monitoring ability of a dominator as 3 hops and the number of total nodes as 100. We try to construct a 3-CDS with our algorithms. Fig. 2 (a) exhibits the initial state of a wireless sensor network. In this state, every node is colored as green. After running Alg. 1, we get Fig. 2 (b). In this figure, the black nodes form a 3-MIS with 11 nodes among 100 candidates. These nodes also form a dominating set which can dominate all other nodes in graph within 3 hops. Then we connect nodes in 3-MIS with some extra nodes with Alg. 2. We color that extra nodes black and form a 3-CDS shown in Fig. 2 (c). Finally, 28 nodes are selected as 3-CDS for the given graph, which is only around 30% of the network.

Table 2: Simulation Parameters

System Parameter	Description
number of nodes in WSN	100-500
number of samples in each round	50
step length	2
number of hops	1-3

Next, let us compare the d-MIS size versus d-CDS size. This parameter reflects the construction cost for our algorithm. Fig. 3 (a)-(c) shows the error bar graph for d-MIS and d-CDS size w.r.t. network size where d is setting as 1, 2, and 3, respectively. In each case we take the best, the average, and the worst case results among 50 distinct experiments. From this figure we can see that as d increases, the size of d-MIS reduces significantly while the size of connectors is relatively increased. This phenomenon is easy to explain from the nature of our algorithm: (1) as d increases, less number of clusters are needed to dominate the whole network; (2) whereas more number of connectors are needed to connect pairwise cluster heads because the distance between them are increased according to d.

7. CONCLUSION

In this paper we propose a distributed algorithm for clustering problem in high dimensional homogeneous wireless sensor networks, which has an approximation ratio of $(d+1)\beta$, where d is the number of hops in each cluster. Our algorithm has 2 subroutines. We first use coloring process to select a d-hop minimum independent set (d-MIS) for a graph G, then connect this d-MIS as a tree. We propose an example to illustrate our idea, give a detailed analysis to this algorithm, and prove its approximation ratio theoretically. Finally, numerical experiments validate the efficiency of our design. To the best of our knowledge, we are the first work to design a distributed approximation algorithm for d-hop connected clustering problem in high dimensional space.

8. REFERENCES

[1] Chi-Fu Huang and Yu-Chee Tseng. The coverage problem in a wireless sensor network. *Mobile Networks and Applications*, 10(4):519–528, 2005.

[2] Vinay Kumar, Sanjeev Jain, Sudarshan Tiwari, et al. Energy efficient clustering algorithms in wireless sensor networks: A survey. *IJCSI International Journal of Computer Science Issues*, 8(5):1694–0814, 2011.

[3] Seema Bandyopadhyay and Edward J Coyle. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *INFOCOM 2003*, volume 3, pages 1713–1723. IEEE, 2003.

[4] Curt Schurgers and Mani B Srivastava. Energy efficient routing in wireless sensor networks. In *MILCOM 2001*, volume 1, pages 357–361. IEEE, 2001.

[5] Luo Chang-ri, Zhu Yun, Zhang Xin-hua, and Zhou Zi-bo. A clustering algorithm based on cell combination for wireless sensor networks. In *ETCS 2010*, volume 2, pages 74–77. IEEE, 2010.

[6] Xiaorong Zhu, Lianfeng Shen, and T-SP Yum. Hausdorff clustering and minimum energy routing for wireless sensor networks. *Vehicular Technology, IEEE Transactions on*, 58(2):990–997, 2009.

[7] Mohammad Alaei and Jose M Barcelo-Ordinas. Node clustering based on overlapping fovs for wireless multimedia sensor networks. In *WCNC 2010*, pages 1–6. IEEE, 2010.

[8] Ossama Younis and Sonia Fahmy. Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach. In *INFOCOM 2004*, volume 1. IEEE, 2004.

[9] Natalija Vlajic and David Xia. Wireless sensor networks: to cluster or not to cluster? In *WoWMoM 2006*, pages 258–268. IEEE Computer Society, 2006.

[10] Jeremy Blum, Min Ding, Andrew Thaeler, and Xiuzhen Cheng. Connected dominating set in sensor networks and manets. In *Handbook of Combinatorial Optimization*, pages 329–369. Springer, 2005.

[11] Xiaofeng Gao and Weili Wu. A constant–factor approximation for d–hop connected dominating sets in unit disk graph. *International Journal of Sensor Networks*, 12(3):125–136, 2012.

[12] Ya Xu, John Heidemann, and Deborah Estrin. Geography-informed energy conservation for ad hoc routing. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 70–84. ACM, 2001.

[13] Suman Banerjee and Samir Khuller. A clustering scheme for hierarchical control in multi-hop wireless networks. In *INFOCOM 2001*, volume 2, pages 1028–1037. IEEE, 2001.

[14] Donghyun Kim and Ding-Zhu Du. A better approximation algorithm for computing connected dominating sets in unit ball graphs. *Mobile Computing*, 9(8):1108–1118, 2010.

[15] Tasi HP Vuong and Dung T Huynh. Adapting d-hop dominating sets to topology changes in ad hoc networks. In *Computer Communications and Networks, 2000*, pages 348–353. IEEE, 2000.

[16] Brent N Clark, Charles J Colbourn, and David S Johnson. Unit disk graphs. *Discrete mathematics*, 86(1):165–177, 1990.

[17] Peng-Jun Wan, Khaled M Alzoubi, and Ophir Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. In *INFOCOM 2002*, volume 3, pages 1597–1604. IEEE, 2002.

[18] Jun Li and Xiaofeng Gao. Performance analysis for approximating mcds in wireless ad-hoc network. *International Information Institute(Tokyo). Information*, 16(2), 2013.

[19] Xiuzhen Cheng, Xiao Huang, and Ding-Zhu Du. A polynomial-time approximation scheme for the minimum-connected dominating set in ad hoc wireless networks. *Networks*, 42(4):202–208, 2003.

[20] Bevan Das and Vaduvur Bharghavan. Routing in ad-hoc networks using minimum connected dominating sets. In *Communications, 1997. ICC 97 Montreal,'Towards the Knowledge Millennium'*, volume 1, pages 376–380. IEEE, 1997.

[21] Fei Dai and Jie Wu. An extended localized algorithm for connected dominating set formation in ad hoc wireless networks. *Parallel and Distributed Systems, IEEE Transactions on*, 15(10):908–920, 2004.

[22] Mario Gerla and Jack Tzu-Chieh Tsai. Multicluster, mobile, multimedia radio network. *Wireless networks*, 1(3):255–265, 1995.

[23] Trac N Nguyen and Dung T Huynh. Connected d-hop dominating sets in mobile ad hoc networks. In *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006*, pages 1–8. IEEE, 2006.

[24] Zhao Zhang, Qinghai Liu, and Deying Li. Two algorithms for connected r-hop k-dominating set. *Discrete Mathematics, Algorithms and Applications*, 1(04):485–498, 2009.

[25] Sergiy Butenko, Sera Kahruman-Anderoglu, and Oleksii Ursulenko. On connected domination in unit ball graphs. *Optimization Letters*, 5(2):195–205, 2011.

[26] Peng-Jun Wan, Khaled M Alzoubi, and Ophir Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. In *INFOCOM 2002*, volume 3, pages 1597–1604. IEEE, 2002.

[27] Israel Cidon and Osnat Mokryn. Propagation and leader election in a multihop broadcast environment. In *Distributed Computing*, pages 104–118. 1998.

[28] Hai Huang, Andréa W Richa, and Michael Segal. Approximation algorithms for the mobile piercing set problem with applications to clustering in ad-hoc networks. *Mobile Networks and Applications*, 9(2):151–161, 2004.