Distributed Construction of Connected Dominating Sets with Minimum Routing Cost in Wireless Networks

Ling Ding*, Xiaofeng Gao*, Weili Wu*, Wonjun Lee†, Xu Zhu‡, and Ding-Zhu Du*

*Dept. of Computer Science, Univ. of Texas at Dallas, Dallas, USA
ling.ding@student.utdallas.edu, {xiaofenggao, weiliwu, dzdu}@utdallas.edu

†Dept. of Computer Science & Engineering, Seoul, Korea Univ., Republic of Korea; wlee@korea.ac.kr

‡Dept. of Mathematics, Xi'an Jiaotong Univ., Xi'an, P. R. China; zhuxu@mail.xjtu.edu.cn

Abstract*—In this paper, we will study a special *Connected Dominating Set* (CDS) problem — between any two nodes in a network, there exists at least one shortest path, all of whose intermediate nodes should be included in a special CDS, named *M*inimum r*O*uting *C*ost CDS (MOC-CDS). Therefore, routing by MOC-CDS can guarantee that each routing path between any pair of nodes is also the shortest path in the network. Thus, energy consumption and delivery delay can be reduced greatly. CDS has been studied extensively in *Unit Disk Graph* (UDG) or *Disk Graph* (DG). However, nodes in networks may have different transmission ranges and some communications may be obstructed by obstacles. Therefore, we model network as a bidirectional general graph in this paper. We prove that constructing a minimum MOC-CDS in general graph is NP-hard. We also prove that there does not exist a polynomial-time approximation algorithm for constructing a minimum MOC-CDS with performance ratio $\rho ln\delta$, where ρ is an arbitrary positive number ($\rho < 1$) and δ is the maximum node degree in network. We propose a distributed heuristic algorithm (called as FlagContest) for constructing MOC-CDS with performance ratio $(1 - ln2) + 2ln\delta$. Through extensive simulations, we show that the results of FlagContest is within the upper bound proved in this paper. Simulations also demonstrate that the average length of routing paths through MOC-CDS reduces greatly compared to regular CDSs.

***Index Terms*—Connected dominating set; shortest path; wireless network; virtual backbones; obstacle; general graph; NP-hard;**

I. Introduction

Different from wired networks, the topology of wireless networks may change from time to time and in some networks (e.g. wireless sensor networks), the energy of nodes is very limited. Therefore, Table Driven and On-Demand routing protocols are unpractical choices [1]. Inspired by contributions of physical backbones to wired networks, a virtual backbone [2] is believed to be very helpful in wireless networks. The reason is that, 1). we can constrain the searching space for routing problems from the whole network to a backbone to reduce routing path searching time and routing table size, and 2). we can utilize such virtual backbones to do shortest path routing.

Looking into techniques for constructing virtual backbones in wireless networks, a *Connected Dominating Set* (CDS) is a good option [3], [4]. CDS also has many other applications which will be introduced in Section II. Because CDS can benefit so much to wireless networks, it has always been a hot topic since it was touched in the first place. CDS is a subset of node set from the original network. Let $G = (V, E)$ denote the original network and S denote a CDS. If the subnetwork induced by S is denoted as $G[S]$, then the rest nodes can induce another subnetwork named $G[V \backslash S]$. According to the definition of CDS, it should be a connected subnetwork and $\forall v \in V \backslash S$, there exists at least one adjacent node u where $u \in S$. Therefore, routing between any nodes can be done through CDS. For example, if node v has packets to send out, it will send to u. Inside CDS, u will forward packets to the destination's adjacent node in S along the shortest path in $G[S]$ if the destination is not in S. When the destination is in S, the routing is done directly within $G[S]$.

How to choose a CDS will determine backbone based routing protocols' performance in wireless networks. If the size of the CDS is too large, it is difficult to maintain it and searching time and routing table size cannot be reduced significantly. If the size of the CDS is too small, some characteristics in original networks may be lost. For instance, in routing protocols, the length of routing path is an important factor. The smaller the length is, the fewer nodes will be involved in routing process. The benefit is that delivery delay, energy cost and interference will be reduced since fewer nodes will participate in forwarding packets. In [5], Mohammed *et al.* also pointed out that in wireless networks, the probability of message transmission failure often increases when a packet is sent through a longer path. Therefore, when designing a routing protocol, we should take path length into consideration, which can be viewed as routing cost. The shorter the routing path is, the less the routing cost will be. That's why shortest path is applied to many routing protocols. However,

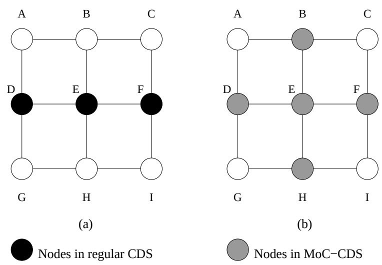


Fig. 1. Illustration of regular CDS and MOC-CDS. (a) A minimum Regular CDS. (b) A minimum MOC-CDS.

most current CDS-based routing protocols only focus on how to reduce the size of CDS while sacrificing shortest path properties in original networks. For example, in Fig. 1 (a), the shortest path between A and C is $\{A, B, C\}$ with length of 2. However, through the chosen minimum CDS, the routing path between A and C will become $\{A, D, E, F, C\}$, with length twice as the original shortest path. Therefore, the routing cost between A and C will increase twice if we use a CDS-based routing protocol.

To solve the problem of increasing routing cost, [5] proposed a concept of *diameter*, which was used to evaluate the length of the longest path between any pair of nodes in a given connected network. Based on *diameter*, [6] presented another concept — *Average Backbone Path Length* (ABPL). Both of the two papers proposed algorithms to achieve balance between the size of minimum CDS and diameter or ABPL in subnetworks induced from CDS. Unfortunately, they did not touch the field of a CDS with minimum routing cost, even though they noticed the importance of routing path length. Routing paths through their CDSs are not guaranteed to be the shortest path within the original network.

To keep the advantages of CDS and conquer the augment of path length in CDS, in this paper, we study a special CDS problem — *M*inimum r*O*uting *C*ost *C*onnected *D*ominating *S*et (MOC-CDS). Besides the constraints of CDS, MOC-CDS has an additional constraint that between any pair of nodes, there exists at least one shortest path in the network, all of whose intermediate nodes must be included in MOC-CDS. Thus, packages can be delivered through MOC-CDS with the same routing cost as that in the original network. For example, in Fig. 1 (b), according to the constraints of MOC-CDS, node (B, D, E, F, H) should be selected as a minimum MOC-CDS. Hence, the shortest path between A and C will still be $\{A, B, C\}$ with same length of 2 as that in the original network.

In addition, due to the instability of topology in wireless networks, it is necessary to update nodes' information periodically to adapt to the change of networks' topology. However, if we update topology in a centralized way, the cost is extremely high. Instead, we should implement a distributed local update strategy. On the other hand, a huge amount of computation cost — for instance, computing a shortest path in the whole network, is also an obstacle to centralized algorithms when the number of nodes in networks are dramatically large. All in all, proposing an efficient distributed algorithm for MOC-CDS becomes practical in wireless networks.

In wireless networks, it is assumed that nodes are homogeneous, sharing the same transmission range. However, in practice, transmission ranges of all nodes are not necessarily identical [7]. On the other hand, radio wave transmission range [8] is not the only reason to determine whether two nodes can communicate or not. In fact, in radio wave transmission based wireless networks, communications among nodes may be obstructed by some obstacles such as buildings and mountains. Therefore, *Unit Disk Graph* (UDG) or *Disk Graph* (DG) may not be a good choice for modeling wireless networks. In addition, communication between two nodes should base on the fact that the two nodes can receive messages from each other. Based on the above factors, we model our networks as a bidirectional general graph.

Our contributions include four aspects in this paper, as shown below:

1) We propose a special CDS problem (MOC-CDS) which not only has constraints of classical CDS but also has minimum routing cost constraint.
2) We prove that selecting a minimum MOC-CDS from a given network is NP-hard.
3) We prove that the approximation ratio in terms of CDS size for MOC-CDS cannot achieve $\rho ln\delta$ unless $NP \subseteq DTIME(n^{O(\log\log n)})$, where ρ is an arbitrary positive number ($\rho < 1$) and δ is the maximum node degree in network.
4) We design an efficient distributed algorithm (named FlagContest) to select a MOC-CDS with approximation ratio of $(1 - ln2) + 2ln\delta$.

The rest of the paper will be organized as follows: in Section II, we will review some related work on CDS. In Section III, we will introduce the communication model and discuss the detailed description of MOC-CDS. An equivalent problem (named 2hop-CDS) to MOC-CDS will be introduced and we will prove that MOC-CDS is NP-hard, by proving that 2hop-CDS is NP-hard. In Section IV, we will introduce how to collect local information and a distributed algorithm (named FlagContest) will be proposed to select a MOC-CDS, based on the local information we collect. Section V will prove the lower bound to approximate MOC-CDS. The approximation ratio of FlagContext will also be proved. In Section VI, simulations show that the results of FlagContest is within the upper bound proved in this paper. Simulations also demonstrate that the average length of routing paths through MOC-CDS reduces greatly compared to regular CDSs. Finally, the paper is concluded in Section VII.

II. Related Work

The research work on selecting minimum CDS has never been interrupted because of its dramatic contributions to wireless networks. It is also well-known that computation of a minimum CDS in a general graph is an NP-hard problem [9] and it is even an NP-hard problem in *Unit Disk Graph* (UDG) [10]. Thus, much work has been devoted to achieving a better approximation ratio.

We first introduce some centralized algorithms for selecting minimum CDS. We can category centralized CDS algorithms into two types — one is 1-stage and the other is 2-stage. In 2-stage algorithms, the first step is to select a minimum *Dominating Set* (DS) and the second step is to construct a minimum CDS using the technique of Steiner Tree [11]. DS is a subset of nodes in original network, where nodes outside DS have at least one adjacent node inside DS. Different from CDS, subnetwork induced by DS may be disconnected. In contrast, 1-stage algorithms aim to select a CDS directly, skipping the step of finding a DS. In [12], two centralized greedy algorithms were proposed. The first algorithm is 1-stage strategy with approximation ratio of $2H(\delta)+2$ where δ is the maximum node degree in the network and H is harmonic function. The second strategy proposed in [12] is a 2-stage strategy and yields a approximation ratio of $H(\delta)+2$. Later, based on the main idea of [12], Ruan *et al.* [13] made a modification of the selection standard of DS. Therefore, 2-stage is reduced to 1-stage, with approximation ratio of $3+ln(\delta)$. Recently, Min *et al*. [14] applied *maximum independent set* (MIS) to the selection of minimum DS because MIS is also a minimum DS in undirectional graph. Min *et al*. [14] used an approximation algorithm proposed by [15] for selecting MIS to obtain a minimum DS with size of $3.8|OPT|+1.2$ and *Steiner Tree with minimum number of Steiner nodes* (ST-MSN) [16][17], was used in the second stage. In [14], Min *et al*. extended the *3*-approximation algorithm in Euclidean plane [16] to a unit-disk graph while keeping the approximation ratio the same. This extended algorithm was applied to construct a Steiner Tree in which terminal points are nodes selected from the first stage. As a result, Min achieved an algorithm for selecting a minimum CDS with size of $6.8|OPT|$ at most.

Due to the limitations of centralized algorithms mentioned in Section I, many efficient distributed algorithms are proposed for selecting a minimum CDS. Motivated by [18], we can divide unweighted CDS into three categories. The first one is greedy CDS construction. Das *et al.* [19] implemented the two centralized algorithms in [12] in a distributed way. They approximated a minimal CDS C^* with a performance ratio of $2H(\delta)+1$ in $O((n+|C^*|)\delta)$ time, using $O(n|C^*|+m+nlogn)$ messages, where m is the cardinality of the edge set. The second one is DS based CDS construction. Most algorithms in this type are divided into two phases. The first phase is to construct a DS using the technique of MIS. And add more nodes to make DS be a CDS in the second phase using the technique of Steiner Tree. Butenko *et al.* [20] proposed a Leader algorithm to achieve an approximation ratio of $8|OPT|+1$ same as that in [21] with time complexity of $O(n)$ and message complexity of $O(n\log n)$. The last type should be pruning based CDS construction. The main idea of this type is that a CDS is constructed firstly with many more redundant nodes. Then prune the redundant nodes from selected CDS to construct a minimum CDS. A typical algorithm of this type is that proposed in [22]. They achieved an approximation of $O(n)$ with time complexity of $O(\delta^3)$.

Due to the fact that not all nodes are homogenous in a network, different nodes may have different transmission ranges. Therefore, [7] studied CDS in Disk Graph (DG).

In addition, CDS has many applications in wireless networks. It can be used in routing [23], broadcasting [24], and topology control [25].

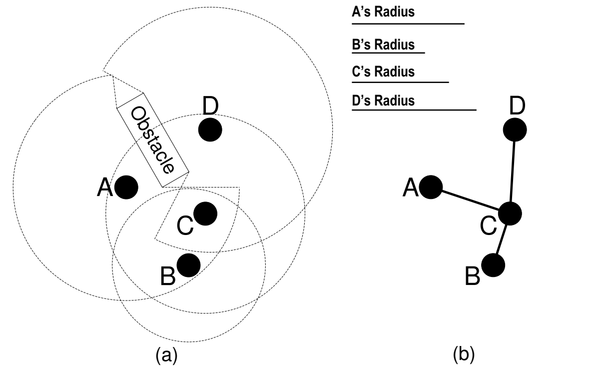


Fig. 2. An example of our network model. (a) A network in which nodes have different transmission ranges and an obstacle exists between A and D. (b) The corresponding graph.

III. Problem Statement

In the section, we'll introduce communication model using graph theory. We will also define MOC-CDS formally. Moreover, we transfer MOC-CDS into an equivalent problem named 2hop-CDS and prove that they are NP-hard in general graph.

A. Network Model

In wireless networks, nodes may have different transmission ranges. For example, in Fig. 2 (a), nodes A, B, C, and D have transmission ranges of r_A, r_B, r_C and r_D respectively, where $r_D > r_A > r_C > r_B$. A is out of B's transmission range while B is in A's transmission range. As a result, A and B cannot communicate with each other normally because even though B can receive messages from A, but A cannot hear from B. Thus, two nodes can communicate only when each of them is within the other's transmission range. That's why A and C can communicate with each other while A and B cannot communicate.

Besides transmission range, presence of obstacle can also prevents communications between two nodes. Two nodes with spatial position close to each other may not be able to communicate directly, since radio wave transmission can be obstructed by an obstacle. In [8], Frank *et al.* introduced that obstacles can cause diffraction, scattering, blocking, and reflection. In our paper, we only consider blocking. In Fig. 2 (a), A and D are within each other's transmission range, however, there is a tall wall between A and D and the wall prevents radio wave transmission between A and D. Therefore, A cannot communicate with D.

Based on the two facts above, we model a network as a connected bidirectional general graph $G=(V,E)$ in which V represents node set in network and E represents link set in network. $\forall u,v\in V$, there exists an edge (u,v) in G if and only if: 1). u is in v's transmission range in the network, 2). v is also in u's transmission range, and 3). there is no obstacle preventing radio wave transmission between u and v. In Fig. 2(b), a general graph is built up from the network in Fig. 2(a), based on the rules mentioned above. In addition, we assume that the graph corresponding to the network is connected. Given a node set $D \subseteq V$, D is said to be connected only when D can induce a connected subgraph from G.

B. Problem Definition

In this paper, a shortest path between u and v is a path whose number of hops is the smallest among all paths between u and v. Distance between u and v is the number of hops on the shortest path between them, denoted as $H(u,v)$. $H(u,v)$ is also viewed as the routing cost between u and v. Let $p(u,v) = \{u, w_1, w_2, ..., w_k, v\}$ be a shortest path between node u and v. $H(u,v) = |p(u,v)| - 1$, where $|p(u,v)|$ represents the number of nodes in $p(u,v)$. All nodes in path $p(u,v)$ except u and v are intermediate nodes. For nodes u and v, there may be more than one shortest paths with the same number of intermediate nodes. Thus, the shortest path set between u and v is defined as $P(u,v)$ including all shortest paths between u and v. For example, in Fig. 1, two shortest paths exist between A and E with $H(A,E) = 2$. The first shortest path is $p_1(A,E) = \{A, B, E\}$, and the second one is $p_2(A,E) = \{A, D, E\}$, so the shortest path set $P(A,E) = \{p_1(A,E), p_2(A,E)\}$.

MOC-CDS has all features in CDS. Besides those features, MOC-CDS has a special constraint. For any two nodes in a network, there exists at least one shortest path between them, all of whose intermediate nodes on the path should be included in MOC-CDS. Thus, MOC-CDS can be formally defined as follows.

Definition 1 (MOC-CDS)**.** *The* Minimum Routing Cost Connected Dominating Set *problem (MOC-CDS) is to find a minimum-size node set $D \subseteq V$ such that*

1) $\forall\, u \in V \backslash D$, $\exists\, v \in D$, *such that* $(u,v) \in E$.
2) *The induced graph* $G[D]$ *is connected.*
3) $\forall\, u, v \in V$, *if* $H(u,v) > 1$, *then* $\exists\, p_i(u,v) \in P(u,v)$, $p_i(u,v) \backslash \{u,v\} \subseteq D$.

We do not consider the situation of $H(u,v) = 1$. The reason is that, our MOC-CDS aims to reduce routing cost. When we select a MOC-CDS no matter in a centralized way or a distributed way, neighbors of $\forall v \in V$ must be known to v during selection process. As a result, when v has a packet destined to u, v will not inform adjacent nodes in MOC-CDS to help deliver the packet, because v knows u can receive packets from v directly and no consecutive forwarding will happen. However, once $H(u,v) > 1$, consecutive forwardings are needed to deliver packages to the destination node. Thus, a good selection of forwarding nodes will influence on network performance greatly. We hope to select a CDS set with minimum size, but keep the value of $H(u,v), \forall u, v \in V$ through this CDS the same as that in original graph. It is the goal of MOC-CDS.

In fact, there exists an equivalent problem to MOC-CDS. The equivalent problem is named *2-hop Shortest Path Connected Dominating Set* (2hop-CDS). We first introduce what is 2hop-CDS and then we prove that the two problems are indeed equivalent to each other.

C. 2hop-CDS Problem

2hop-CDS is also a CDS. It requires that, for any two nodes with distance equal to 2, there exists at least one shortest path between them, whose intermediate node should be included in 2hop-CDS.

The formal definition is shown in details as follows.

Definition 2 (2hop-CDS)**.** *The* 2-hop Shortest Path Connected Dominating Set *problem (2hop-CDS) is to find a minimum-size node set $D^{'} \subseteq V$ such that*

1) $\forall\, u \in V \backslash D^{'}$, $\exists\, v \in D^{'}$, *such that* $(u,v) \in E$.
2) *The induced graph* $G[D^{'}]$ *is connected.*
3) $\forall\, u, v \in V$, *if* $H(u,v) = 2$, *then* $\exists\, p_i(u,v) \in P(u,v)$, $p_i(u,v) \backslash \{u,v\} \subseteq D^{'}$.

Next, we will show that the MOC-CDS and 2hop-CDS are equivalent to each other by Lemma 1.

Lemma 1. *A dominating set D is a MOC-CDS if and only if it is a 2hop-CDS.*

Proof: (1) If D is a MOC-CDS, then for any two nodes with hop distance of 2, there exists a shortest path all of whose intermediate nodes belongs to D. It is trivial that D is a minimum size solution for 2hop-CDS. Otherwise if we eliminate one node u from D, then it is not a MOC-CDS, which means that $D \backslash \{u\}$ does not satisfy rule 1), 2), or 3) in Def. 1. Easy to prove that $D \backslash \{u\}$ does not satisfy rule 1), 2), or 3) in Def. 2 either. Thus, D is also a 2hop-CDS.

(2) Conversely, assuming D satisfies Def. 2, we show that D also meets Def. 1. Consider a shortest path $p(u,v) = \{u, w_1, w_2, ..., w_k, v\}$. Let's consider odd k firstly. According to Def. 2, there exist $s_1, s_3, ..., s_k \in D$ such that $\{u, s_1, w_2\}$, $\{w_2, s_3, w_4\}$, $\cdots$, $\{w_{k-1}, s_k, v\}$ are shortest paths. w_i and w_{i+2} are not adjacent to each other, otherwise we can connect w_i and w_{i+2} directly and $|p(u,v)|$ will reduce. That is, $p(u,v)$ is not a shortest path between u and v. Let $p'(u,v) = \{u, s_1, w_2, s_3, ..., w_{k-1}, s_k, v\}$, which is also a path between u and v (see Fig. 3 as an example). Easy to know that $|p(u,v)| = |p'(u,v)|$. It means that $p'(u,v)$ is also a shortest path between u and v. In $p'(u,v)$, some intermediate nodes are in D while others are not. Next, we will use similar way to $p'(u,v)$ to find a $p''(u,v)$ whose all intermediate nodes in $p''(u,v)$ should belong to D. According to Def. 2, there also exist $s_2, s_4, ..., s_{k-1} \in D$ such that $\{s_1, s_2, s_3\}$, $\{s_3, s_4, s_5\}$, $\cdots$, $\{s_{k-2}, s_{k-1}, s_k\}$ are shortest paths. According to the definition of shortest path $p''(u,v) = \{u, s_1, s_2, ..., s_k, v\}$ is a shortest path between u and v where all intermediate nodes belong to D, since $|p'(u,v)| = |p''(u,v)|$. When k is even, proof is similar to the situation of odd k. There exists at least one shortest path between u and v, having all intermediate nodes included in D. Next, we need to show that D is minimum. This is easy to achieve since if we remove a node $u \in D$, $D \backslash \{u\}$ is not a 2hop-CDS, which does not satisfy rule

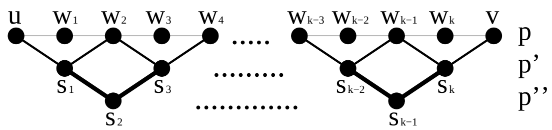


Fig. 3. An example to prove the equivalence of MOC-CDS and 2hop-CDS.

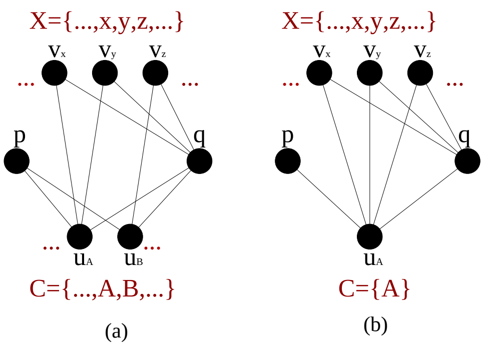


Fig. 4. Reduction from Set-Cover to 2hop-CDS (a) Reduction with $|\mathcal{C}| > 1$. (b) Reduction with $|\mathcal{C}| = 1$.

1), 2), or 3) in Def. 2. Thus it cannot satisfy rules in Def. 1, either. Therefore, we conclude that when D meets Def. 2, it also meets Def. 1.

Based on (1) and (2) above, we get that 2hop-CDS and MOC-CDS are equivalent to each other. ■

In fact, finding a 2hop-CDS cannot be done in polynomial time. Next, we prove that 2hop-CDS in general graph is NP-hard. It has already been proved that selecting a minimum Set-Cover is an NP-hard problem [26]. Motivated by this, we construct a reduction from Set-Cover to 2hop-CDS. We first clarify the concept of Set-Cover, given in Def. 3. Then, we will give the proof in Theorem 1.

Definition 3 (Set-Cover)**.** *Given a collection $\mathcal{C}$ of subsets of a finite set X such that $\bigcup_{A\in\mathcal{C}} A = X$, find a minimum subcollection $\mathcal{A} \subseteq \mathcal{C}$ such that $\bigcup_{A\in\mathcal{A}} A = X$.*

Theorem 1. *Selecting a 2hop-CDS in general graph is NP-hard.*

Proof: We first show that 2hop-CDS$\in NP$. Given a graph $G = (V, E)$ and an integer k. The certificate we choose is the 2hop-CDS $V' \subseteq V$. The verification algorithm affirms that $|V'| = k$, and then it checks, for each pair of nodes $u, v \in V$ having $H(u, v) = 2$, $\exists w \in V'$ and $\{u, w, v\}$ is a path between u and v. This verification can be performed straightforwardly in $O(n^2)$ — polynomial time.

Next, we prove that 2hop-CDS is NP-hard by showing that Set-Cover$\leq_P$ 2hop-CDS. It is important to note that Set-Cover is NP-hard even for the special case of $|\mathcal{C}| \leq |X|$ in [26]. Our following proof is based on the special case.

For each $A \in \mathcal{C}$, we create a node u_A and for each element $x \in X$, we create a node v_x. In addition, we create two nodes p and q. Connect p to every u_A for $\forall A \in \mathcal{C}$, connect q to every u_A for $\forall A \in \mathcal{C}$, and also connect q to every v_x for $\forall x \in X$. If and only if $x \in A$, an edge between v_x and u_A will be added. The resulting graph is denoted as G. In Fig. 4 (a), v_x, v_y, v_z represent elements $x, y, z \in X$ respectively, u_A, u_B represent elements in $A, B \in \mathcal{C}$ respectively. Edge between v_x and u_A represents $x \in A$.

We claim that $\mathcal{C}$ has a Set-Cover solution $\mathcal{A}$ of size at most k if and only if G has a dominating set of size at most $k + 1$ satisfying Def. 2.

(1). We first prove when $|\mathcal{A}| \leq k$ then we can obtain a D in G having $|D| \leq k + 1$. Our claim holds trivially in case of $|\mathcal{C}| = 1$ as shown in Fig. 4 (b). The reason is that, $\mathcal{A} = \mathcal{C}$ since only one element $A \in \mathcal{C}$ such that $k = |\mathcal{A}| = 1$. Thus, a 2hop-CDS in Fig. 4 (b) should be $\{u_A, q\}$. The size of the minimum 2hop-CDS is 2 which is equal to $k + 1$. We next prove correctness of our claim when $|\mathcal{C}| > 1$. First, assume $\mathcal{C}$ has a Set-Cover $\mathcal{A}$ of size at most k. Then, it is easy to verify that $D = \{u_A|A \in \mathcal{A}\} \bigcup \{q\}$ is a connected dominating set of G satisfying 2hop-CDS. "Dominating" is because p can be dominated by any node in $\{u_A|A \in \mathcal{A}\}$, and any node in $\{v_x|x \in X\}$ or $\{u_A|A \in \mathcal{C}\}$ is dominated by q. "Connected" is because any two nodes in $\{u_A|A \in \mathcal{A}\}$ can be connected through q. q is connected to any other node directly. Lastly, we prove that D is a 2hop-CDS. For any pair of nodes m and n in G with $H(m, n) = 2$, $p \neq m$, and $p \neq n$, $\{m, q, n\}$ must be a shortest path which is obvious in Fig. 4 (a). For a pair of nodes p and m with $H(p, m) = 2$, m must belong to $\{v_x|x \in X\} \bigcup \{q\}$. If $m \in \{v_x|x \in X\}$, then there must exist $A \in \mathcal{A}$ having $m \in A$. Thus, an edge between m and u_A should be in G. Therefore, path $\{m, u_A, p\}$ should be in G which is also a shortest path between m and p. (Existence of A can be proved by contradiction. If no such $A \in \mathcal{A}$ exists, then $X \neq \bigcup_{A\in\mathcal{A}}(A)$ which means subcollection $\mathcal{A}$ is not Set-Cover. Contradiction happens.) If $m = q$, then for any $A \in \mathcal{A}$, (p, u_A, m) is a shortest path. Thus, $D = \{u_A|A \in \mathcal{A}\} \bigcup \{q\}$ is a 2hop-CDS with $|D| \leq k + 1$.

(2). Conversely, suppose that G has a 2hop-CDS D of size at most $k + 1$, then its corresponding Set-Cover problem has $|\mathcal{A}| \leq k$. Note that distance between p to $\forall v \in \{v_x|x \in X\}$ is 2 and every shortest path from p to v must pass a node u_A for some $A \in \mathcal{C}$. Therefore, $\mathcal{A} = \{A|u_A \in D \text{ and } A \in \mathcal{C}\}$ is a Set-Cover. For $A, B \in \mathcal{C}$ with $A \neq B$, distance between u_A and u_B is 2 and for every shortest path between u_A and u_B, but the intermediate node is not in $\{u_E|E \in \mathcal{C}\}$. This means that there exists a node in D, but not in $\{u_E|E \in \mathcal{A}\}$. Hence, $|\mathcal{A}| \leq k$.

In summary, 2hop-CDS is NP-hard. Based on Lemma 1, MOC-CDS is NP-hard as well. ■

IV. Algorithm Description

In this paper, we study networks in which nodes may have different transmission ranges. In a homogeneous network, if one node v receives messages from node w, v can determine that w is its neighbor and w can receive messages from v. However, in our networks, v receives messages from w does not mean w can receive messages from v. Only when two nodes can receive each other's messages, the two nodes can be neighbors. To maintain 1-hop neighbor information for every node, 2-round "Hello" message is needed. After 1-hop information is collected, one more round of "Hello" message is needed to construct 2-hop inforamtion. Before introducing our algorithm, we first show how to maintain neighbor information as follows.

A. Neighbor Information Maintenance

$N_{in}(v)$ is used to denote the set of nodes from which node v can receive messages. In contrast, $N_{out}(v)$ denote the set of nodes which can receive messages from v, and $N(v) = N_{in}(v) \cap N_{out}(v)$. At the beginning, every node knows nothing of others. Each node v sends periodical "Hello" messages out. "Hello" message is piggybacked with v's id, $N_{in}(v)$, and $N_{out}(v)$. At first, $N_{in}(v)$ and $N_{out}(v)$ are empty. v can construct its $N_{in}(v)$. In the following round, by exchanging non-empty $N_{in}(v)$, if v receives $N_{in}(w)$ and $v \in N_{in}(w)$, then add w to $N_{out}(v)$. Once we collect $N_{in}(v)$ and $N_{out}(v)$, $N(v)$ can be obtained trivially.

Technically, $N^2(v) = N(v) \bigcup \cup_{u \in N(v)} N(u)$. According to $N^2(v)$, v can decide whether two nodes u and w in $N(v)$ have a link (u, w). In the last round, 2-hop neighbor information can be collected based on 1-hop neighbor information which has already been collected. In this round, if v receives $N(w)$, then v adds $\{u | u \in N(w) \wedge u \notin N(v)\}$ to $N^2(v)$.

Finally, we can collect 1-hop and 2-hop neighbor information for all nodes in a network.

B. FlagContest

The reason we introduce 2hop-CDS is because 2hop-CDS only considers shortest paths with length of 2, so that we can design an algorithm based on local information. On the contrary, in MOC-CDS, the whole topology may be needed to find a shortest path between some two nodes. As a result, it is difficult to find an efficient distributed algorithm for MOC-CDS. Fortunately, we find an equivalent problem 2hop-CDS which can be solved in localized way, then the localized solution also applies to MOC-CDS.

The basic idea of FlagContest is a greedy strategy. Each node v has to fight with its neighbors for a "flag". According to the 2-hop neighbor information of v, v can compute the times it can act as an intermediate node of other two nodes u and w where $H(u, w) = 2$, denoted as $f(v)$. v will send flag to the node with highest $f(u)$, where $u \in N(v) \cup \{v\}$. When and only when one node collect all flags from its neighbors, then it will be selected as a node in 2hop-CDS. Therefore, we name this algorithm *FlagContest*. Flag contests only happen among neighbors. Therefore, FlagContest is distributed since each node can make decision only based on its 2-hop neighborhood information.

Before introducing FlagContest, we first introduce some definitions used in the algorithm. For a given graph $G = (V, E)$, assign each node v a unique id, denoted as $id(v)$. id is used for pruning redundant tie in Step 2. Each node v has a storage $P(v)$. Initially, $P(v) = \{(u, w) | u, w \in N(v)$ and $H(u, w) = 2\}$. Algorithm will stop when $P(v) = \phi$, $\forall v \in V$. The algorithm description is given in Alg. 1.

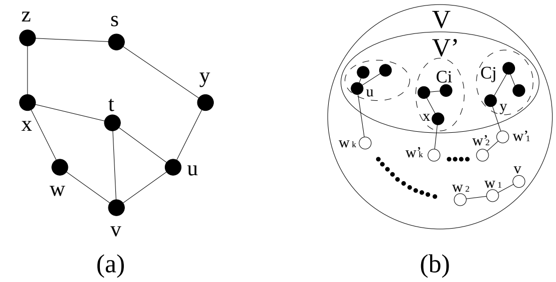


Fig. 5. (a) Illustration of forwarding. (b) Proof of correctness of FlagContest.

Algorithm 1 distributed Selection of MOC-CDS & 2hop-CDS (FlagContest)

Step 1. Each node v with nonempty $P(v)$, calculates $f(v) = |P(v)|$ and sends $f(v)$ to its neighbors;

Step 2. Each node v computes maximum value m among received $f(u)$'s from its neighbors in Step 1 (including itself). Choose a neighbor u with $|f(u)| = m$ and send a flag to u. If there are more than one such u, then break tie by choosing the one with highest id;

Step 3. If a node v receives flags from all its neighbors, then change color to black and send $P(v)$ to all of its neighbors. Lastly, set $P(v) = \phi$;

Step 4. If a node u receives $P(v)$ from some neighbor v, then pass $P(v)$ to all neighbors of u;

Step 5. If a node u receives $P(v)$ from a neighbor w in Step 4, then compute union U of such $P(v)'s$ and update $P(u)$ by setting $P(u) \leftarrow P(u) - U$.

At the end of algorithm, output the set of all black nodes which is a 2hop-CDS, also a MOC-CDS.

According to Alg. 1, if v is selected as a node in MOC-CDS, then only those nodes in $N(v) \bigcup \{v\}$ need to send out $P(v)$ to their neighbors. That is, when and only when a node u receives $P(v)$ directly from v, u needs to forward $P(v)$ to others. For example, in Fig. 5 (a), suppose v will be selected as black, then $P(v) = \{(u, w), (w, t)\}$ will be sent out from v. When w receives $P(v)$, it will forward $P(v)$ to its neighbors. However, since x cannot receive $P(v)$ directly from v, x does not need to forward $P(v)$. The reason is that, there are two kinds of neighbors of x. One is the neighbors which are also v's neighbors (e.g. in Fig. 5 (a), t is a neighbor of v and is also a neighbor of x). The situation change of v will directly affect such kind of neighbors once v sends $P(v)$ to them. The other one is the neighbors which do not contain any pair of nodes in $P(v)$ in their "P" sets, that is $P(v) \bigcap P(\text{neighbor of } x) = \phi$, referring to z in Fig. 5 (a). $P(z) = \{(s, x)\} \bigcap P(v) = \phi$. No action will be taken in z's view, even though it receives $P(v)$. This means the situation change of v will not affect this kind of neighbors. Therefore, forwarding of $P(v)$ at x's view will be useless.

Fig. 6 shows a large scale example in a 9×8 area. 20 nodes are deployed in the area. Transmission range may vary from node to node. Dark nodes, as shown in the figure represent

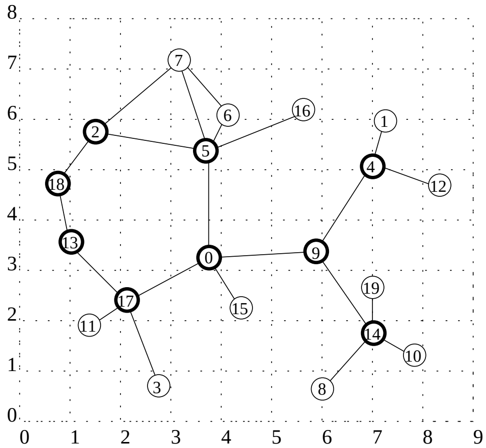


Fig. 6. An example of MOC-CDS by FlagContest.

a MOC-CDS obtained by FlagContest. In the resultant graph, there are 9 members in the selected MOC-CDS. In the first round, every node v calculates $f(v)$, and send $f(v)$ to its neighbors. For example, when node 2 receives all "f" values from its neighbors, it finds that node 5 has the biggest "f", so it sends a flag to node 5. Similarly, node 0, 6, 7, and 16 also send flags to node 5 because node 5 has the biggest "f" from their viewpoints. After node 5 collects flags from all its neighbors, it will be colored as black, according to Step 3 in Alg. 1. Using the same strategy, we find that node 14 will also be colored as black in the first round. Then, $P(5) = \{(0,2), (0,6), (0,7), (0,16), (2,6), (2,16), (6,16), (7,16)\}$ and $P(14) = \{(8,9), (8,10), (8,19), (9,10), (9,19), (10,19)\}$ will be sent to their neighbors respectively. After this, $P(5)$ and $P(14)$ will be set as empty. Nodes 0, 2, 6, 7, and 16 will forward $P(5)$ they receive to their neighbors while nodes 8, 9, 10, and 19 will forward $P(14)$ to their neighbors. Here, we use node 7 as an example to show how nodes update their set "P". When node 7 receives $P(5)$, it will recalculate $P(7)$. Originally, $P(7) = \{(2,6)\}$, which is included in $P(5)$, so we will delete $(2,6)$ from $P(7)$. Finally, $P(7) = \emptyset$. All nodes receive $P(5)$ or $P(14)$ will update their set "P"s like node 7. Till now, the first round is done. Since not all nodes' set "P"s are empty, the following rounds will continue till all "P"s are empty.

V. Theoretical Analysis

In this section, we will prove the correctness and calculate the approximation ratio of FlagContest algorithm. We also prove that the approximation ratio of MOC-CDS cannot achieve $\rho ln\delta$ unless $NP \subseteq DTIME(n^{O(\log\log n)})$, where ρ is an arbitrary positive number ($\rho < 1$) and δ is the maximum node degree in network. Firstly, we will prove that all black nodes colored by Alg. 1 construct an MOC-CDS.

Theorem 2. *Given a graph $G = (V, E)$, FlagContest will select a node set V' satisfying three rules in Def. 2.*

Proof: This proof uses method of contradiction based on the fact that Alg. 1 stops normally when $P(v) = \phi$, $\forall v \in V$.

Firstly, we prove that all nodes outside V', as white nodes shown in Fig. 5 (b), can be dominated by nodes in V'. On the contrary, assume v is not dominated by V', then $H(v,u) \geq 2, \forall u \in V'$. Find a nearest u to v from V'. Choose a $p(v,u) = \{v, w_1, ..., w_k, u\} \in P(v,u)$. Since $H(v,u) \geq 2$, there exist at least one w_1, such that $w_1 \neq v$, and $w_1 \neq u$. Thus, we have $(v, w_2) \in P(w_1)$, or $(v,u) \in P(w_1)$, which means $P(w_1) \neq \emptyset$. This contradicts to $P(v) = \phi$, $\forall v \in V$. Therefore, V' should be a dominate set.

Next, we prove that $G[V']$ is connected. Similarly, assume $G[V']$ is not connected. Denote every connected components in V' as C_i. Find one pair of disconnected nodes in $G[V']$ as $x \in C_i$ and $y \in C_j$ ($i \neq j$), such that the distance between x and y are the shortest among all pairs of connected components in $G[V']$. Then choose one $p(x,y) = \{x, w_1', ..., w_k', y\} \in P(x,y)$, we have $w_i' \in V \backslash V'$ ($1 \leq i \leq k$), shown in Fig. 5 (b). Similarly, there exists at least one w_1', such that $w_1' \neq x$, and $w_1' \neq y$. Now we have $(x, w_2') \in P(w_1')$ or $(x,y) \in P(w_1')$, otherwise x and w_2' will connect directly, which means $p(x,y)$ is not a shortest path. Or x and y will connect directly. However, now $P(w_1') \neq \emptyset$, which violates the strategy of FlagContest. Therefore, $G[V']$ should be connected.

Lastly, we check whether every pair of nodes with distance 2 having at least one intermediate node colored black. If two nodes $u, v \in V$ having $H(u,v) = 2$ and no intermediate node w of any path between u and v belongs to V', then $P(w)$ should not be empty. This contradicts to $P(v) = \phi$, $\forall v \in V$.

Therefore, V' selected by FlagContest satisfies three rules in Def. 2. ∎

In [26], Feige proved that for Set-Cover problem, there does not exist a polynomial-time approximation with performance ratio $\rho \ln n$, where $\forall \rho < 1$ and n is number of elements in $\mathcal{C}$, unless $NP \subseteq DTIME(n^{O(\log\log n)})$. Based on [26],we show that there exists a lower bound that no polynomial time algorithm can achieve for MOC-CDS in Theorem 3.

Theorem 3. *Neither MOC-CDS nor 2hop-CDS has polynomial-time approximation with performance ratio $\rho \ln \delta$ where $\forall \rho < 1$ and δ is the maximum node degree of input graph, unless $NP \subseteq DTIME(n^{O(\log\log n)})$, for sufficient large n and $opt_{\mathcal{A}}$.*

Proof: Based on proof of Theorem 1, an immediate corollary of our claim is that optimal Set-Cover $\mathcal{A}$ contained in $\mathcal{C}$ has size $opt_{\mathcal{A}}$ if and only if optimal MOC-CDS D of corresponding G has size of $opt_D = opt_{\mathcal{A}} + 1$. We next use contradiction method to prove that MOC-CDS cannot have approximation algorithm with a performance ratio of $\rho \ln \delta$, $\forall \rho < 1$.

Assume G has a polynomial-time approximation solution D' for 2hop-CDS with size at most $(\rho \ln \delta)(opt_D)$ for some constant $\rho < 1$. The solution and ratio can also be applied to MOC-CDS. Note that we assume in Fig. 4, $|\mathcal{C}| \leq n$ where $n = |X|$, and node q has the biggest degree in G obviously. Thus, $\delta = |\mathcal{C}| + |X| \leq 2n$. Based on Theorem 1's proof, when MOC-CDS in G has a solution with size of k at most, corresponding Set-Cover problem should have a solution with size of $k - 1$. Thus, we can find a polynomial-time approximation solution for Set-Cover with size at most $(\rho \ln \delta)(opt_D) - 1 < (\rho \ln 2n)(opt_{\mathcal{A}} + 1) < 0.5(\rho + 1) \ln n \times opt_{\mathcal{A}}$ for sufficiently large n and sufficiently large $opt_{\mathcal{A}}$. (Note: for any constant α, we can check in polynomial-time whether $opt_D \le \alpha$ and if it is true, $opt_{\mathcal{A}}$ can be determined at the same time.) This implies that $NP \subseteq DTIME(n^{O(\log \log n)})$ which has proved to be wrong. Therefore, assumption G has a polynomial-time approximation solution D with size at most $(\rho \ln \delta)(opt_D)$ for some constant $\rho < 1$ is incorrect. All in all, Theorem 3 is proved. ■

We reduce Set-Cover problem to 2hop-CDS to prove a lower bound of MOC-CDS problem. Next, we will show there exists a polynomial time approximation with an upper bound performance ratio of MOC-CDS problem by converting 2hop-CDS to hitting set problem. If we can prove this upper bound of 2hop-CDS, then this upper bound is also applied to MOC-CDS.

Theorem 4. *A polynomial time approximation algorithm can be designed for 2hop-CDS with performance ratio of* $(1 - \ln 2) + 2 \ln \delta$ *at most, where* δ *is the maximum node degree of input graph.*

Proof: For each pair of nodes (u, v) with distance 2, define $m(u, v) = \{w | \{u, w, v\} \text{ is a path}\}$. Now, finding a minimum 2hop-CDS problem becomes finding a minimum hitting set [27], in $U = \{\bigcup m(u, v) | u, v \in V\}$. That means if we can find a minimum hitting set for a $m(u, v)$, the minimum hitting set is also a minimum solution to 2hop-CDS and MOC-CDS. In [27], the author proposed a greedy algorithm for finding a minimum hitting set with the performance ratio of $1 + \ln \gamma$ at most, where γ is the maximum number of $m(u, v)$'s that a node can appear. In addition, by deducing 2hop-CDS to hitting set problem, we have $\gamma \le \delta(\delta - 1)/2$. As a result, $1 + \ln \gamma \le (1 - \ln 2) + 2 \ln \delta$. Therefore, there exists a ploynomial time approximation algorithm for selecting a minimum MOC-CDS with performance ratio of $(1 - \ln 2) + 2 \ln \delta$ at most. ■

Next, we need to prove Alg. 1 has the same bound.

Theorem 5. *Alg. 1 (FlagContest) produces approximation solution with performance ratio* $H(\binom{\delta}{2})$*, where* H *is the Hamonic function and* δ *is the maximum node degree of input graph.*

Proof: Let $P_0(v)$ be the initial $P(v)$ and $X = \bigcup_{v \in V} P_0(v)$. Then problem of Def. 2 is equivalent to Set-Cover problem with base set X and collection $\mathcal{C} = \{P_0(v) | v \in V\}$. Suppose D^* gives the minimum solution $\{P_0(v) | v \in D^*\}$ to the Set-Cover problem of X and $\mathcal{C}$. We partition X into subsets $X(v)$ for $v \in D^*$ such that $\forall X(v)$, $X(v) \subseteq P_0(v)$.

Consider $v \in D^*$. Denote $f_0 = X(v)$ before the first round. We will make a charge to $(u, w) \in X(v)$ when (u, w) is removed from $P(v)$ during the computation of distributed algorithm. When u is colored in black, we charge $1/f(u)$ to $\forall (w, y) \in P(u)$.

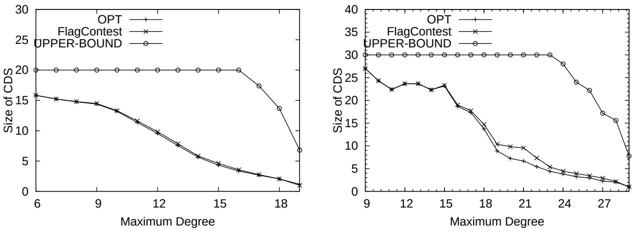
Suppose that at the end of Step. 5 in Alg. 1 in the first round, $f_0 - f_1$ elements of $X(v)$ are charged. Then $\forall (u, w) \in X(v)$ is charged by the value at most $1/f_0$. The total charge for those $f_0 - f_1$ removed elements is at most $(f_0 - f_1)/f_0$.

Similarly, let f_i be the number of uncharged elements in $X(v)$ at the end of Step 5 in the ith round. Then the total charge to elements of $X(v)$ is at most $(f_{i-1} - f_i)/f_i$.

Suppose $f_k = 0$. Then all elements of $X(v)$ are charged at total value as follows:

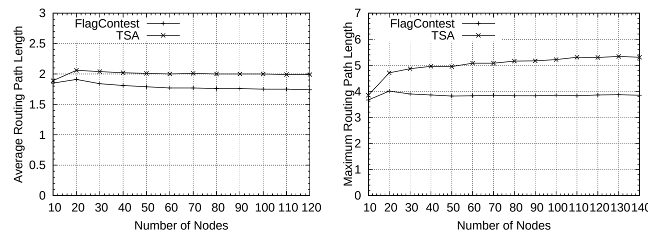
$$\sum_{i=0}^{k-1} \frac{f_i - f_{i+1}}{f_i} \le \sum_{i=1}^{f_0} \frac{1}{i} = H(f_0) \le H(\binom{\delta}{2}) \qquad (1)$$

Note that when a node v is colored in black, the total value of charging to $P(v)$ is one. Therefore, the total value charging to elements of X is exactly the number of black nodes at the end of distributed algorithm. This number is bounded by $H(\binom{\delta}{2}) \times |D^*|$. ■



(a) 20 nodes (b) 30 nodes

Fig. 7. Illustration of Bound of size of MOC-CDS in General Networks.

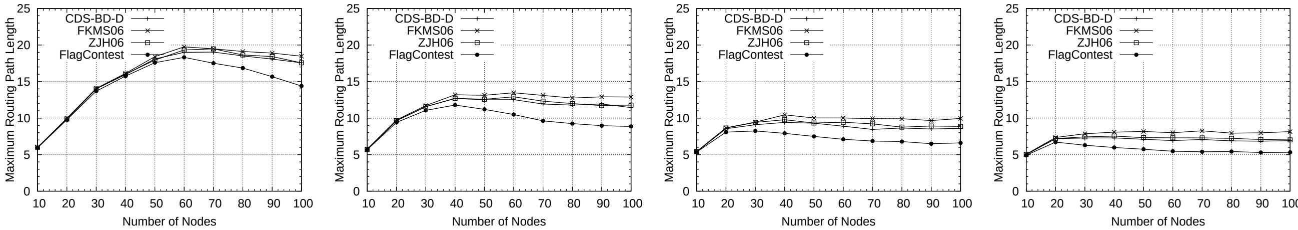


(a) Average Routing Path Length (b) Maximum Routing Path Length

Fig. 8. Comparison of Average Routing Path Length and Maximum Routing Path Length in DG Networks between FlagContest and TSA.

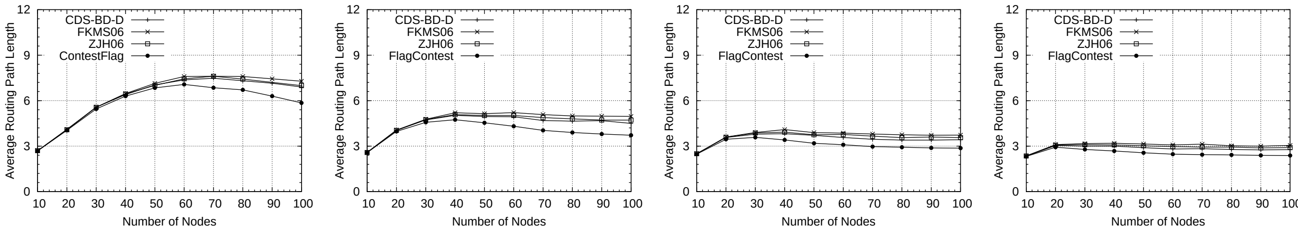
VI. SIMULATION

This part includes two subparts. The first one evaluates whether the size of MOC-CDS obtained from FlagContest is under the upper bound we have already proved. The second one evaluates FlagContest by comparing MOC-CDS with traditional CDS without shortest path constraint. We compare them in terms of *Maximum Routing Path Length* (MRPL) and *Average Routing Path Length* (ARPL). In the simulations, if node s in a network has a package to d, s will send the package to its adjacent nodes in the CDS, and a shortest path in the CDS will be chosen to forward the package to d's adjacent nodes in CDS, that is, forwarding is done within CDS. MRPL is defined as the maximum routing path length in the network, while ARPL is defined as the average length of routing paths in the network.



(a) Transmission Range = 15 (b) Transmission Range = 20 (c) Transmission Range = 25 (d) Transmission Range = 30

Fig. 9. Comparison of Maximum Routing Path Length among CDS-BD-D, SAUM06, ZJH06, and FlagContest in UDG Networks.



(a) Transmission Range = 15 (b) Transmission Range = 20 (c) Transmission Range = 25 (d) Transmission Range = 30

Fig. 10. Comparison of Average Routing Path Length among CDS-BD-D, SAUM06, ZJH06, and FlagContest in UDG Networks.

A. Simulation Environment

To evaluate FlagContest, we test three types of networks. The first type is a network in which nodes may have different transmission ranges and obstacles may obstruct communications among nodes. This type is named General Network because this type can be modeled as a general graph. In General Networks, we show that FlagContest is under the upper bound we have proved above. The second type is a network in which different transmission ranges are allowed, however, obstacles are not considered. The second type is named DG Network since this type can be modeled as a disk graph. In DG Network, we will compare FlagContest with TSA [7]. The last one is an ideal one in which all nodes should have the same transmission ranges and no obstacle exists. This one is named UDG Network because it can be modeled as a unit disk graph. In UDG Network, FlagContest will be compared with CDS-BD-D [6], FKMS06 [28], and ZJH06 [29].

1) General Network: To simulate network of this category, n nodes are randomly deployed to a fixed area of $100m \times 100m$. Since we have to use brute-force search, we can only get optimal solution when network size is limited ($n = 20$ or $n = 30$). For a certain n, the maximum degree of a network can vary from 1 to $n - 1$. Here, once we fix a certain n and a maximum degree, we generate 100 instances. Nodes are assigned a transmission range randomly. Definitely, we have to generate a connected network as our input. We take the average value among 100 instances as our results.

2) DG Network: To simulate networks of this category, n nodes are randomly deployed to a fixed area of $800m \times 800m$. n varies from 10 to 120 with increment of 10. Each node v is randomly assigned a transmission range $r \in [r_{min}, r_{max}]$, where $r_{min} = 200m$ and $r_{max} = 600m$. For each value of n, 1000 network instances are investigated. Results of the same n are averaged among 1000 instances.

3) UDG Network: To simulate network of this category, n nodes are deployed randomly in a fixed area of $100m \times 100m$ and all nodes have the same transmission range. n is incremented from 10 to 100 by 10, while transmission range varies among $15m$, $20m$, $25m$, and $30m$. For a certain n and transmission range, 100 instances are generated. Results are averaged among 100 instances.

B. Simulation Results

Fig. 7 shows that the size of MOC-CDS selected by FlagContest is significantly less than the upper bound and very close to that of the optimal solution. Note the bigger the maximum degree is, the smaller size of MOC-CDS is. The reason is that a node with bigger degree can be an intermediate node of more shortest paths between pairs of nodes, which can reduce the size of CDS greatly.

Fig. 8 shows that the MRPL and the ARPL of FlagContest are smaller than those of TSA. This illustrates that our FlagContest can also work well in DG Network. From this figure, the ARPL of FlagContest is about 12.5% less than that of TSA while the MRPL of FlagContest is about 20% less than that of TSA. TSA tends to include nodes with larger transmission range in CDS. However, large transmission range does not necessarily mean big node degree which is a selection criteria of FlagContest.

Fig. 9 and Fig. 10 show that FlagContest is also efficient in UDG Networks. As shown in Fig. 9 and Fig. 10, the MRPL of FlagContest is about 20%-40% better and the ARPL of FlagContest is around 10%-30% better, when the number of nodes exceeds 30. Note ARPL and MRPL increase firstly and then decrease. The reason is that in a connected network with small size of nodes, the routing path length is more likely to increase when a new node is added. For example, a network with 1 node inside has ARPL equal to 0. When a new node is connected to the network, both ARPL and MRPL will increase to 1. Hence, routing path length increases when n increases (n is relatively small). However, when n exceeds a certain value, newly added nodes are more likely to make distance between nodes smaller and the network more connected (considering physical space is fixed). That's why there is a decrease when network size becomes big enough. In addition, when transmission range increases, networks are more connected considering physical space is fixed. It is trivial to conclude that routing path will decrease when transmission range increases, which explains both MRPL and ARPL decrease while transmission range increases as shown in Fig. 9 and Fig. 10.

VII. Conclusion

In this paper, we propose a minimum routing cost connected dominating set (MOC-CDS). MOC-CDS aims to find a minimum CDS while assuring that any routing path through this CDS is shortest in networks. It is proved that selecting a minimum MOC-CDS is NP-hard. A lower bound of approximation ratio of MOC-CDS is proved in this paper. We also propose an efficient distributed algorithm for constructing MOC-CDS with performance ratio $1 - \ln 2 + 2 \ln \delta$, where δ is the maximum node degree in the network. Compared with traditional CDS, using a MOC-CDS as a virtual backbone in wireless networks can reduce routing cost significantly. Our future work includes further study of FlagContest by doing simulations in more realistic simulation environments like NS2.

Acknowledgment

This research was supported by National Science Foundation of USA under Grant CNS0831579 and CCF0728851. This research was also jointly supported by MEST, Korea under WCU (R33-2008-000-10044-0), by the KOSEF Grant funded by the Korea government (MEST) (No. R01-2007-000-11203-0), by KRF Grant funded by (KRF-2008- 314-D00354), and by MKE, Korea under ITRC IITA-2009-(C1090-0902-0046) and IITA-2009-(C1090-0902-0007). The contact author is Wonjun Lee.

References

[1] Y.-P. Tsai, T.-L Hsu, R.-S. Liu, and Y.-K. Ho. A backbone routing protocol based on the connected dominating set in ad hoc networks. *World Congress on Computer Science and Information Engineering*, 1:14–18, 2009.

[2] X. Cheng, M. Ding, D. H. Du, and X. Jia. Virtual backbone construction in multihop ad hoc wireless networks: Research articles. *Wireless Communications and Mobile Computing*, 6(2):183–190, 2006.

[3] P.-J. Wan, K. M. Alzoubi, and O. Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. *Mobile Networks and Applications*, 9(2):141–149, 2004.

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

[5] K. Mohammed, L. Gewali, and V. Muthukumar. Generating quality dominating sets for sensor network. In *Proc. of IEEE ICCIMA*, 2005.

[6] D. Kim, Y. Wu, Y. Li, F. Zou, and D.-Z. Du. Constructing minimum connected dominating sets with bounded diameters in wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 20(2):147–157, 2009.

[7] M. T. Thai, F. Wang, D. Liu, S. Zhu, and D.-Z. Du. Connected dominating sets in wireless networks with different transmission ranges. *IEEE Transactions on Mobile Computing*, 6(7):721–730, 2007.

[8] F. Reichenbach, R. Salomon, and D. Timmermann. Distributed obstacle localization in large wireless sensor networks. In *Proc. of ACM IWCMC*, 2006.

[9] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*. W. H. Freeman, 1979.

[10] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86(1-3):165–177, 1990.

[11] D.-Z. Du, B. Lu, H. Ngo, and P. M. Pardalos. Steiner tree problems, 2000.

[12] S. Guha and S. Khuller. Approximation algorithms for connected dominating sets. *Algorithmica*, 20(4):374–387, 1996.

[13] L. Ruan, H. Du, X. Jia, W. Wu, Y. Li, and K.-I. Ko. A greedy approximation for minimum connected dominating sets. *Theoretical Computer Science*, 329(1-3):325–330, 2004.

[14] M. Min, H. Du, X. Jia, C. X. Huang, S. C.-H. Huang, and W. Wu. Improving construction for connected dominating set with steiner tree in wireless sensor networks. *Journal of Global Optimization*, 35(1):111–119, 2006.

[15] W. Wu, H. Du, X. Jia, Y. Li, and S. C.-H. Huang. Minimum connected dominating sets and maximal independent sets in unit disk graphs. *Theoretical Computer Science*, 352(1):1–7, 2006.

[16] D. Chen, D.-Z. Du, X.-D. Hu, G.-H. Lin, L. Wang, and G. Xue. Approximations for steiner trees with minimum number of steiner points. *Journal of Global Optimization*, 18(1-3):17–33, 2000.

[17] G.-H. Lin and G. Xue. Steiner tree problem with minimum number of steiner points and bounded edge-length. *Information Processing Letters*, 69(2):53–57, 1999.

[18] D.-Z. Du and P. Pardalos. *Handbook of Combinatorial Optimization*. Kluwer Academic, 2004.

[19] B. Das and V. Bharghavan. Routing in ad-hoc networks using minimum connected dominating sets. In *Proc. of IEEE ICC*, 1997.

[20] S. Butenko, X. Z. Cheng, D.-Z. Du, and P. M. Pardalos. On the construction of virtual backbone for ad hoc wireless network. In *Proc. of Conference on Cooperative Control and Optimization*, 2001.

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

[22] J. Wu, F. Dai, M. Gao, and I. Stojmenovic. On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks. *IEEE/KICS Journal of Communications and Networks*, 4:59–70, 2002.

[23] W. Di, Q. Yan, and T. Ning. Connected dominating set based hybrid routing algorithm in ad hoc networks with obstacles. In *Proc. of IEEE ICC*, 2006.

[24] L. Ding, Y. Shao, and M. Li. On reducing broadcast transmission cost and redundancy in ad hoc wireless networks using directional antennas. In *WCNC*, pages 2295–2300, 2008.

[25] B. Deb, S. Bhatnagar, and B. Nath. Multi-resolution state retrieval in sensor networks. In *Proc. of IEEE International Workshop on Sensor Network Protocols and Applications*, 2003.

[26] U. Feige. A threshold of ln n for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[27] D. S. Johnson. Approximation algorithms for combinatorial problems. In *Proc. of ACM STOC*, 1973.

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

[29] Y. Zeng, X. Jia, and Y. He. Energy efficient distributed connected dominating sets construction in wireless sensor networks. In *Proc. of ACM IWCMC*, 2006.