On Constructing k-Connected k-Dominating Set in Wireless Networks *

Fei Dai and Jie Wu
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431

Abstract

An important problem in wireless networks, such as wireless ad hoc and sensor networks, is to select a few nodes to form a virtual backbone that supports routing and other tasks such as area monitoring. Previous work in this area has focused on selecting a small virtual backbone for high efficiency. We propose to construct a k-connected k-dominating set (k-CDS) as a backbone to balance efficiency and fault tolerance. Three localized k-CDS construction protocols are proposed. The first protocol randomly selects virtual backbone nodes with a given probability p_k, where p_k depends on network condition and the value of k. The second protocol, a deterministic approach, extends Wu and Dai's coverage condition, which is originally designed for 1-CDS construction, to ensure the formation of a k-CDS. The last protocol, a hybrid of probabilistic and deterministic approaches, provides a generic framework that can convert many existing CDS algorithms into k-CDS algorithms. These protocols are evaluated via simulation study.

Keywords: Connected dominating set (CDS), k-vertex connectivity, localized algorithms, simulation, wireless networks.

*This work was supported in part by NSF grants CCR 0329741, CNS 0434533, CNS 0422762, and EIA 0130806. Contact address: jie@cse.fau.edu

1 Introduction

In wireless ad hoc and sensor networks, autonomous nodes form self-organized networks without centralized control or infrastructure. These networks can be modelled as unit disk graphs [9], where two nodes are neighbors if they are within each other's transmission range. To support various network functions such as multi-hop communication and area monitoring, some wireless nodes are selected to form a *virtual backbone*. In many existing schemes [1, 2, 4, 8, 11, 16, 26, 29, 30], virtual backbone nodes form a connected dominating set (CDS) of the wireless network. A set of nodes is a dominating set if all nodes in the network are either in this set or have a neighbor in this set. A dominating set is a CDS if the subgraph induced from this dominating set is connected. For example, both node sets $\{8\}$ in Figure 1 (a) and $\{5, 6, 7, 8\}$ in Figure 1 (b) are connected dominating sets in their corresponding networks. Applications of a CDS in wireless networks include:

- *Reducing routing overhead* [30]. By removing all links between non-virtual backbone nodes, the size and maintenance cost of routing tables can be reduced. By using only backbone nodes to forward broadcast packets, the excessive broadcast redundancy can be avoided.

- *Energy efficient routing* [8]. By putting non-backbone nodes into periodical sleep mode, the energy consumption is greatly reduced while network connectivity is still maintained by backbone nodes.

- *Area coverage* [7]. In densely deployed sensor networks, the node coverage of a CDS is a good approximation of area coverage. That is, the deployment area is within the sensing range of backbone nodes with high probability.

Previous study in this area has focused on finding a minimal CDS for higher efficiency. However, recent study [3, 5, 17, 21, 22] suggested that it is equally important to maintain a certain degree of redundancy in the virtual backbone construction for fault tolerance and routing flexibility. In wireless ad hoc networks, a node may fail due to accidental damage or energy depletion, and a wireless link may fade away during node movement. In a wireless sensor network, it is desirable to have several sensors monitor the same target, and let each sensor report the sensed data via different routes to avoid loss of an important event.

We propose to construct a k-connected k-dominating set (or simply k-CDS) as a backbone of wireless networks. A node set is k-dominating if every node is either in the set or has k neighbors in the set. A k-dominating set is a k-CDS if its induced subgraph is k-vertex connected. A graph is k-vertex connected if removing any $k - 1$ nodes from it does not cause a partition. For example, backbone nodes 5, 6, 7, and 8 in Figure 1 (b) form a 2-CDS. Every non-backbone node has at least two neighboring backbone nodes, and the subgraph consisting of all backbone nodes is 2-vertex connected. Similarly, node set $\{2, 4, 5, 6, 7, 8\}$ in Figure 1 (c) is a 3-CDS. Removing any $k - 1$ nodes from a k-CDS, the remaining nodes still form a CDS (i.e., 1-CDS). Therefore, a k-CDS as a virtual backbone can survive failures of at least $k - 1$ nodes.

Three k-CDS construction protocols are proposed in this paper. All those protocols are *localized algorithms* that rely on only neighborhood information. The first protocol, called k-Gossip, is a simple

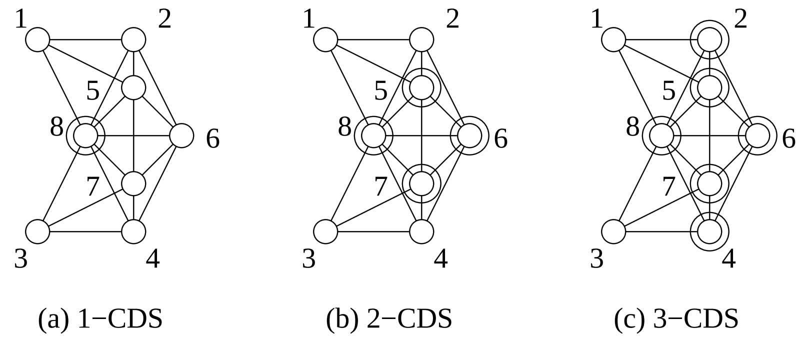


Figure 1. k-connected k-dominating sets constructed by applying k-coverage conditions with $k = 1, 2$, and 3. Virtual backbone nodes are represented by double circles.

extension of an exiting probabilistic algorithm [16], where each node becomes a backbone node with a given probability p_k. This algorithm has very low overhead, but the implementation parameter p_k that maintains a k-CDS with high probability depends on network size and density. In addition, the randomized backbone node selection process usually produces a large backbone. The second protocol extends our early deterministic CDS algorithm [29], where each node has a backbone status by default and becomes a non-backbone node if a *coverage condition* is satisfied. The proposed k-coverage condition guarantees all backbone nodes form a k-CDS but has relatively high computation overhead. We further introduce a hybrid paradigm to extend many existing CDS algorithms for k-CDS formation. In this scheme, a wireless network is randomly partitioned into k subgraphs consisting of nodes with different colors (the probabilistic part). A colored virtual backbone is constructed for each subgraph using a traditional CDS algorithm (the deterministic part). We prove that in dense wireless networks, the union of all colored backbones is a k-CDS with high probability. Simulation study is conducted to compare performances of these protocols.

The remainder of this paper is organized as follows. Section 2 reviews existing virtual backbone construction protocols, including both probabilistic and deterministic schemes, and introduces the concept of k-CDS. In Section 3, we propose extensions of two virtual backbone protocols for k-CDS construction. Section 4 presents the color-based k-CDS formation paradigm. Section 5 gives simulation results, and Section 6 concludes this paper.

2 Background and Related Work

In this section, we first introduce two existing localized virtual backbone formation algorithms, one probabilistic and another deterministic, that will be extended for k-CDS construction in the next section. Then we review concepts of k-connectivity and k-CDS, and algorithms that verify k-connectivity and form a k-CDS.

2.1 Virtual backbone construction

A wireless network is usually modelled as a unit disk graph [9] $G = (V, E)$, where V is the set of wireless nodes and E the set of wireless links. Each node in V is associated with a coordination in 2-D or 3-D Euclidean space, and a wireless link $(u, v) \in E$ if and only if the Euclidean distance between nodes u and v is smaller than a uniform transmission range R. In real wireless networks, the transmission range of each node may not be a perfect disk. In this case, the network is a quasi-unit disk graph [19], where a bidirectional link (u, v) definitely exists if the distance between u and v is less than a certain value $d < R$, and may or may not exist when the distance is larger than d but smaller than R.

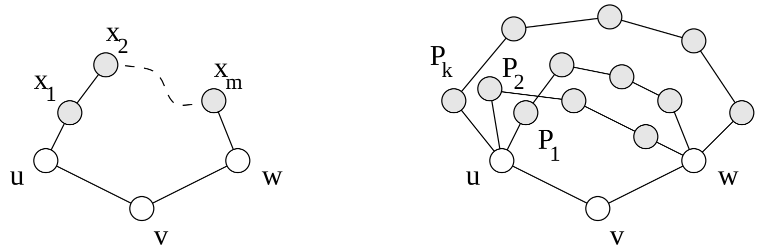
Many schemes have been proposed to construct a connected dominating set (CDS) as a virtual backbone to support routing activities in wireless networks. A set $V^{'} \subseteq V$ is a CDS of network G, if all nodes in $V - V^{'}$ are neighbors of (i.e., dominated by) a node in $V^{'}$ and, in addition, the subgraph $G[V^{'}]$ induced from $V^{'}$ is connected. The problem of finding a minimum CDS is NP-complete. Centralized [11] and cluster-based [2, 4] CDS algorithms provide hard performance guarantees (i.e., upper bounds on CDS size) in wireless networks. However, those schemes require either global information or global coordination, which limit their applications to static or almost static networks. In dynamic networks, most existing CDS formation algorithms are *localized*; that is, the status of each node, backbone or non-backbone, depends on its d-hop neighborhood information only with a small d.

Localized CDS algorithms are either *probabilistic* or *deterministic*. A typical probabilistic scheme is the gossip-based algorithm [14, 16].

Gossip [16]: Each node v has a backbone status with probability p.

The selection of backbone nodes in Gossip is purely random without using any neighborhood information. Simulation results show that when p is larger than a threshold, these backbone nodes form a CDS with very high probability. This threshold depends on network size and density and is determined based on experimental data. To maintain high success ratio (i.e, the probability of constructing a CDS) under unpredictable network situation, the selection of p is usually conservative, which produces a large backbone. In wireless networks with a non-uniform node distribution, grid-based [6, 24] algorithms can be used to control backbone node density. These schemes are originally proposed as topology control schemes, but can be modified for virtual backbone construction. The basic idea is that if every node has B backbone nodes, then these backbone nodes form a CDS with high probability. The value of B is also determined based on experimental data.

Deterministic algorithms [1, 8, 26, 30] guarantee a CDS in connected networks. They usually select fewer backbone nodes than probabilistic schemes, because their selections are "smarter" using 2-hop neighborhood information (or simply 2-hop information). For each node v, its 2-hop information consists of its neighbor set $N(v)$ and neighbor sets $N(u)$ of all neighbors $u \in N(v)$, and is collected via 2 rounds of "Hello" exchanges among neighbors. The *complete 2-hop information* of v is a subgraph of G, including v's entire 2-hop neighbor set, and all adjacent links of v's 1-hop neighbors. Some algorithms use v's *restricted 2-hop information*, which is the subgraph $G([N(v)]$ induced from v's 1-hop neighbor set. One reason to use restricted 2-hop information is that, in quasi-unit disk graphs, a bidirectional link (u, w) between a 1-hop neighbor u and a 2-hop neighbor w cannot be confirmed via 2 rounds of "Hello"



(a) 1−coverage condition (b) k−coverage condition

Figure 2. Replacement paths between two neighbors u and w of node v. Gray nodes have higher priorities than that of v.

exchanges. Another reason is that applying a localized algorithm on a smaller subgraph can reduce the computation cost.

In [30], Wu and Li proposed a deterministic CDS algorithm called marking process and two backbone node pruning rules called Rules 1 and 2, which were later replaced by an enhanced rule called Rule k [10]. Chen et al [8] designed a backbone formation protocol called Span, which is similar to the combination of the marking process and Rules 1 and 2. Qayyum et al [26] provided another backbone formation scheme called MPR, and Adjih et al [1] enhanced this scheme to construct a smaller CDS. In [29], Wu and Dai showed that all above algorithms are special cases of the following coverage condition.

Coverage Condition [29]: Node v has a non-backbone status if for any two neighbors u and w, a *replacement path* exists that connects u and w via several intermediate nodes (if any) with higher id's than v.

When applying the coverage condition, each node tries to find a replacement path between every pair of its neighbors. Figure 2 (a) shows a sample replacement path $(u, x_1, x_2, \ldots, x_m, w)$ that connects two neighbors of the current node v. Since node v knows only its 2-hop information, all intermediate nodes $x_1, x_2, \ldots, x_m$ are within 2 hops of v. In addition, all intermediate nodes must have a higher priority than node v. A priority is a unique attribute of each node, such as node id or the combination of node degree (i.e., $|N(v)|$) and node id. Node priorities establish a total order among nodes to avoid simultaneous withdrawals that may cause a partition in the virtual backbone. If every node pair of v's neighbors are connected via high priority nodes, then v can be safely removed from the backbone while the remaining nodes still form a CDS.

In Figure 1 (a), node 1 is a non-backbone node based on the coverage condition, because its neighbors 2, 5, and 8 are directly connected. Node 2 has two neighbors 1 and 6 that are not directly connected. However, nodes 1 and 6 are connected via a replacement path $(1, 5, 6)$. Here we assume node id is used as priority, and node 5 has a higher priority than 2. Therefore, node 2 has a non-backbone node. Similarly, nodes 3, 4, 5, 6, and 7 are also non-backbone nodes. The resultant backbone, consisting of node 8 only, is a CDS of the network.

2.2 k-connectivity

Many existing works [3, 5, 17, 21, 22] suggested to maintain k-vertex connectivity (or simply k-connectivity) in wireless networks for fault tolerance and/or high throughput.

Definition 1 (k-Vertex Connectivity) *A network G is k-vertex connected if it is connected and removing any $1, 2, \dots, k-1$ nodes from G will not cause partition in G.*

An equivalent definition is that a network is k-vertex connected if any two nodes in the network are connected via k node disjoint paths (Menger's theorem [25]) . The network in Figure 1 is 3-connected, where any two nodes are connected via three node disjoint paths. For example, nodes 1 and 3 are connected via node disjoint paths $(1, 8, 3)$, $(1, 5, 7, 3)$, and $(1, 2, 6, 4, 3)$. Maximal flow (minimal cut) algorithms [13] are usually employed to discover all node disjoint paths between a pair of source/sink nodes. A general purpose maximum flow algorithm has a computation complexity of $O(|V||E|)$. But a variation of the Edmonds and Karp's algorithm [12] using flow augmentation can verify if two nodes are connected via k node disjoint paths within $O(k|E|)$ time. That is because each augmentation (i.e., the process of finding a new path) is a breadth-first search in G, which takes $O(|E|)$ time, and it takes no more than k augmentations to find (or verify the non-existence of) k node disjoint paths.

Definition 2 (k-Connected k-Dominating Set) *A node set $V' \subseteq V$ is a k-dominating set (or simply k-DS) of G if every node not in V' has at least k neighboring nodes in V'. A k-DS is a k-connected k-dominating set (or simply k-CDS) of G if the subgraph $G[V']$ induced from V' is k-vertex connected.*

The previous definition of CDS (also called 1-CDS) is a special case of k-CDS with $k = 1$. Several schemes [3, 21, 22] have been proposed to maintain the k-connectivity in topology control. Basu and Redi [5] designed a centralized algorithm for achieving 2-connectivity in wireless networks using mobile nodes. Jorgic, Nayak, and Stojmenovic [17] suggested to use local k-connectivity to approximate global k-connectivity based on neighborhood information. The problem of constructing double dominating set and k-dominating set in general graph has been studied in [15, 23]. In [18], three heuristic algorithms are provided to construct a double dominating set. The problem of localized construction of a k-CDS has not been discussed.

3 k-Extensions of Existing CDS Algorithms

In this section, we extend both probabilistic and deterministic localized CDS algorithms (Gossip and the Wu and Dai's coverage condition) to construct k-CDS in wireless networks, and show limits of these extensions. In the next section, we will introduce a new approach, color-based coverage condition (CBCC), to overcome those limits. These three localized K-CDS algorithms are compared in Table 1.

Table 1. k-CDS algorithms.

Algorithm	Guarantees k-CDS	Backbone Size (expected)	Comm. Rounds	Message Size	Computation Cost
k-gossip	No	np_k	0	N/A	$O(1)$
k-coverage	Yes	unknown	2	$O(\Delta)$	$O(k\Delta^4)$
CBCC	No	$O(1)OPT$	2	$O(\Delta)$	$O(\Delta^3)$

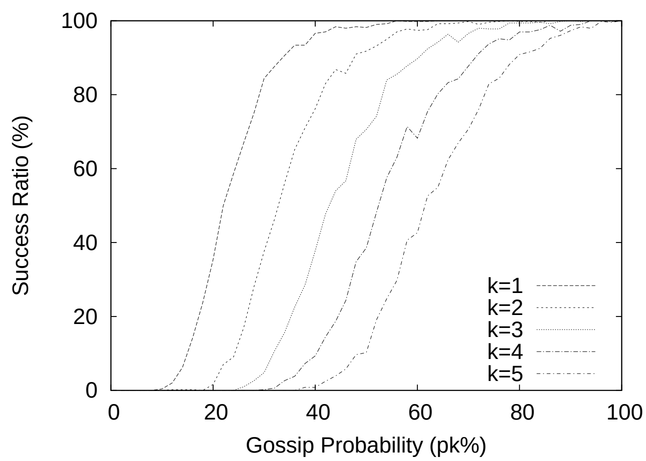


Figure 3. Success ratio of k-CDS construction under different gossip probability p_k.

3.1 Probabilistic approach

The gossip-based algorithm can be easily extended to construct a k-CDS with high probability. The extended rule for selecting backbone nodes is as follows:

k**-Gossip:** Each node v has a backbone status with probability p_k.

Note that the above rule is almost identical to its 1-CDS version. The difference is that the probability p_k that any node becomes a backbone node is now a function of k. In k-Gossip, the perfect value of p_k, which constructs a small virtual backbone while maintaining a k-CDS with high probability, depends not only on k, but also on total node number n, deploy area A, and transmission range R. Some analytical study has provided an upper bounds of p_k that almost always achieve k coverage in various networks [20]. However, these upper bounds are conservative estimations of the perfect p_k, which usually need to be refined based on experimental data. Figure 3 shows our experiment results in a sample network, where 200 nodes with transmission range $250m$ are randomly deployed in a $1000 \times 1000m^2$ region. For each k, there exists a p_k that almost always (i.e., with a probability very close to 1) selects a k-CDS. For example, when $k = 2$, using $p_k = 50\%$ constructs a 2-CDS with probability 98.2%. When $k = 3$, using $p_k = 60\%$ achieves a success ratio of 97.4%.

As in its 1-CDS counterpart, k-Gossip incurs very low overhead at each node. It requires no information exchange among neighbors and very low ($O(1)$) computation cost. Therefore, the backbone construction process completes almost instantaneously. The major drawback is that it requires some global information, such as network size and density, to be effective. The expected number of backbone nodes in k-Gossip is np_k. If different values of p_k are used under different circumstances, global network information, such as node number n and deployment area A, must be collected and broadcast to each node. If the above global information is unknown and a single p_k is used for different network situations, the selection of p_k must be very conservative to maintain a k-CDS in the worst case scenario, which yields a larger backbone size of $O(n)$.

3.2 Deterministic approach

The original coverage condition [29] that constructs a 1-CDS can be extended as follows to construct a k-CDS.

k-Coverage Condition: Node v has a non-backbone status if for any two neighbors u and w, k *node disjoint replacement paths* exist that connect u and w via several intermediate nodes (if any) with higher id's than v.

In the original coverage condition, a node can be removed from a CDS if all its neighbors are interconnected via a replacement path. In the k-coverage condition, the criterion is more strict: if a node is to be removed from a k-CDS, all its neighbors must be k-connected with each other via higher priority nodes. This criterion is shown by Figure 2 (b), where two neighbors u and w of the current node v are connected via node disjoint paths $P_1, P_2, \ldots, P_k$ consisting of high priority (gray) nodes. The following theorem shows that k-coverage condition guarantees a k-CDS in originally k-connected networks.

Lemma 1 *A node set V' is a k-CDS of network G if after removing any $k-1$ nodes from G, the remaining part of V' is a CDS of the remaining part of G.*

Proof: First, V' is a k-dominating set of G. Because otherwise, there exists a node v in G with less than k neighbors in V'. After removing all those neighbors from V', node v is no longer dominated by V', which contradicts the assumption that the remainder of V' dominates the remainder of G. Second, V' is still connected after removing any $k-1$ nodes; that is, V' is k-connected. □

Theorem 1 *If the k-coverage condition is applied to a k-connected network G, then the resultant virtual backbone V' forms a k-CDS of G.*

Proof: Let V be the set of all nodes and X be the set of any $k-1$ nodes from V'. Since G is k-connected, its subgraph G' induced from $V-X$ is also connected. Let v be any non-backbone node in $V-V'$. Based on the k-coverage condition, any two neighbors u and w of v are connected via k node disjoint replacement paths. After removing $k-1$ nodes from G, u and w are still connected via at least one replacement path in G'. Since all non-backbone nodes in G' satisfy the original coverage condition, the remaining nodes in $V-V'$ form a CDS of G' [29]. From Lemma 1, V' is a k-CDS of G. □

When $k = 1$, the k-coverage condition is equivalent to the original coverage condition. Figure 1 (b) shows a 2-CDS constructed by the k-coverage condition with $k = 2$. Here node 5 becomes a backbone node, because two of its neighbors, nodes 1 and 6, are connected by only one replacement path. On the other hand, nodes 1, 2, 3, and 4 are non-backbone nodes, because all their neighbors are connected via 2 node disjoint replacement paths. The resultant virtual backbone, containing nodes 5, 6, 7, and 8, is a 2-CDS of the network. Similarly, nodes 2, 4, 5, 6, 7, and 8 in Figure 1 (c) are selected as backbone nodes when $k = 3$. Here we assume each node uses complete 2-hop information; otherwise, both nodes 1 and 3 will be backbone nodes. When node 1 uses restricted 2-hop information, it can only find two replacement paths between neighbors 2 and 8: $(2, 8)$ and $(2, 5, 8)$. The third node disjoint path $(2, 6, 8)$ is invisible in restricted 2-hop information.

It has been proved in [10] that expected size of the resultant CDS derived from the original coverage condition is $O(1)$ times the size of a minimal CDS in an optimal solution. Unfortunately, we cannot prove a similar bound for k-CDS with $k > 2$. Another extension of the coverage condition that holds this bound will be discussed in the next section.

The k-coverage condition depends on local information only. No global information such as network size is required. The size of the resultant virtual backbone is barely affected by the network density. The k-coverage condition has the same message size and rounds of information exchange as the original coverage condition. When 2-hop information is collected, each node sends two messages with size $O(\Delta)$, where Δ is the maximal node degree. However, k-coverage condition is more complex than the original condition. Each node needs to compute the vertex connectivity among $O(\Delta^2)$ pairs of neighbors using the maximal flow algorithm with time complexity $O(k|E|)$ discussed in Section 3.1. When the algorithm uses restricted 2-hop information, $|E| = O(\Delta^2)$ and it takes $O(k\Delta^2)$ time to verify whether two neighbors are k-connected. The overall computation cost at each node is $O(k\Delta^4)$, which is much higher than that of the original coverage condition ($O(\Delta^3)$). Although some density reduction methods [28] can be employed to reduce Δ in very dense networks, these methods also introduce extra overhead and slower convergency.

4 Color-Based k-CDS Construction

This section introduces a hybrid paradigm that can easily convert an existing 1-CDS algorithm to construct a k-CDS with high probability in relatively dense networks. Unlike probabilistic schemes, this new approach does not depend on any network specific parameter. This approach is also easier to implement and has lower overhead than the deterministic algorithm discussed in the previous section. We use Wu and Dai's coverage condition [29] as an example to show how to convert a CDS algorithm using this paradigm.

4.1 A hybrid paradigm

As shown in the last section, when extending an existing CDS algorithm to construct k-CDS, the original algorithm needs to be re-calibrated or modified, and usually becomes more complex in concepts and implementation techniques. In this section, we propose a hybrid paradigm, called color-based k-CDS construction (CBKC), to make the migration process smoother and effortless. The basic idea is to randomly partition the network into several subnetworks with different colors, and apply a tradition CDS algorithm within each subnetwork. The first step is probabilistic; when the network is sufficiently dense, colored nodes in each partition almost always form a CDS of the original network. The second step is deterministic; each *colored backbone* constructed within a subnetwork by a CDS algorithm is still a CDS of the entire network. Together, k colored backbones form a k-CDS. Since any CDS algorithm $\mathcal{A}$ can be used in constructions of colored backbones, our color-based scheme provides a general framework for extending a wide range of existing CDS algorithms to construct k-CDS in relatively dense wireless networks.

Color-Based k-CDS Construction (CBKC)

1. Each node v selects a random color c_v ($1 \leq c_v \leq k$) for itself. As a result, the node set V is divided into k disjoint subsets $V_1, V_2, \ldots, V_k$, with each subset V_c containing nodes with color c.

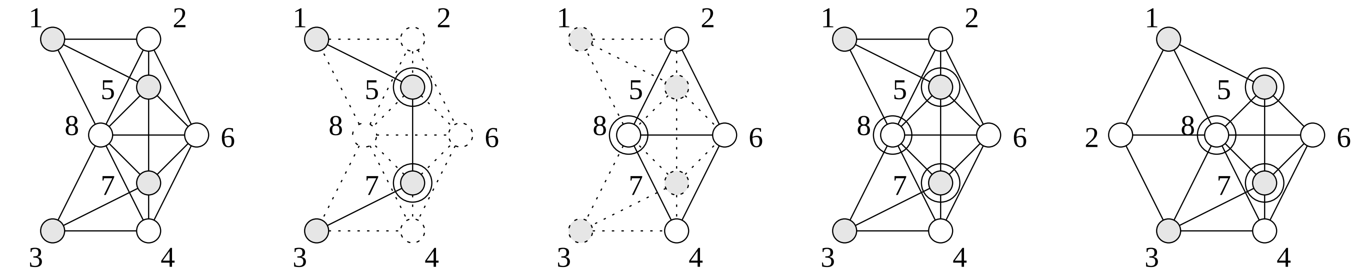
2. For each color c, a localized CDS algorithm $\mathcal{A}$ is applied to construct a virtual backbone $V_c' \subset V_c$ that covers the original network.

3. The final k-CDS is the union $\bigcup_{c=1}^{k} V'_c$ of all colored virtual backbones.

Figure 4 illustrates the color-based k-CDS construction process. In Figure 4 (a), all nodes are randomly assigned color (1) gray and (2) white. In Figure 4 (b), two gray nodes 5 and 7 are selected to form a CDS of the entire network. In Figure 4 (c), a single node 8 is selected from white nodes to form a CDS. The set of all backbone nodes $\{5, 7, 8\}$ forms a 2-CDS of the network, as shown by Figure 4 (d). The following theorem shows that the above generic scheme almost always construct a k-CDS in dense networks.

Theorem 2 *If all nodes in the network are randomly placed in a finite square region, then CBKC almost always constructs a k-CDS when the node number exceeds a constant n_k.*

Proof: We first show that each node set V_c formed at step 1 is a CDS of the network G with high probability when node number is sufficiently large. It has been proved in [20] that given a probability p and a radius r, there exists a $n(p, r)$ such that when $n \geq n(p, r)$ nodes are randomly deployed in a unit square, and each node is marked a color c with probability p, then the entire region is almost always covered by those marked nodes (i.e., every point in this region is within distance r of a marked node). Suppose both the actual square area A and the actual transmission range R are fixed. Let $n_k = n(\frac{1}{k}, \frac{R}{2\sqrt{A}})$. It is easy to see that when $n \geq n_k$ nodes are randomly and uniformly divided into k sets $V_1, V_2, \ldots, V_k$, each set set V_c almost always covers the region under transmission range $R/2$. It has been proved in [27] that a set achieving area coverage with covering radius $R/2$ is connected under transmission range R. Therefore, each V_c is a CDS of G.



(a) Color assignment (b) VB of color 1 (c) VB of color 2 (d) Final 2−CDS (e) Failure of CBCC−II

Figure 4. Color-based coverage condition. (a) Nodes with odd id numbers are of color 1 (gray), and nodes with even id's are of color 2 (white). (b-c) Two colored virtual backbones (represented by double circles) are constructed using the coverage condition. Nodes with different colors and their adjacent links (represented by dotted circles and lines) are not considered by CBCC-II (d) The final 2-CDS consists of all backbone nodes. (e) CBCC-II fails when a colored backbone does not form a CDS of the entire network.

When each set V_c is a CDS of G, the virtual backbone V'_c selected by algorithm $\mathcal{A}$ in step 2 is also a CDS of G. Let $V' = \bigcup_{c=1}^{k} V'_c$ be the union of k node disjoint CDS's of G. After removing $k-1$ from $V^{'}$, there is at least one $V_c^{'}$ untouched. Therefore, the remaining nodes in $V^{'}$ still form a CDS of G. From Lemma 1, $V^{'}$ is a k-CDS of G. □

4.2 Color-based coverage condition

We use the coverage condition as an example to illustrate the effectiveness of the color-based paradigm. When the original coverage condition is extended using the CBKC framework, only one modification is needed in the following revised rule:

Color-Based Coverage Condition (CBCC): Node v has a non-backbone status if for any two neighbors u and w, a replacement path exists that connects u and w via several intermediate nodes (if any) *with the same color* and higher priorities than that of v.

Figure 4 (a-d) shows an example of CBCC. Note that in color-based coverage condition, the search for a replacement path is now restricted to nodes with the same color. This modification actually reduces the average computation cost, but the worst case computation complexity is still the same ($O(\Delta^3)$). Color-based coverage condition also inherits the constant probabilistic bound of the original coverage condition [10].

Theorem 3 *The expected number of backbone nodes selected by color-based coverage condition is $O(1)$ times the optimal value.*

Proof: It was shown in [10] that the expected number of backbone nodes selected by the original coverage condition is $O(A/R^2)$, where A is the area of a rectangular deployment region and R is the transmission range. Since the virtual backbone constructed by color-based coverage condition consists of k colored backbones, the total number of backbone nodes is $O(kA/R^2)$. Note that any k-dominating set needs at least $O(kA/R^2)$ nodes to maintain k-coverage. Therefore, the expected backbone size of CBCC is $O(1)$ times the minimal k-dominating set, which is no larger than a minimal k-CDS. □

To further reduce the message and computation cost, we consider a more aggressive variation of CBCC. The original color based coverage condition (called CBCC-I) covers all neighbors regardless of their colors; that is, any two neighbors of a non-backbone node must be connected via a replacement place. For example, node 3 in Figure 4 (e) is a backbone node in CBCC-I, because it has two neighbors 2 and 7 that are not connected via a gray replacement path. In the more aggressive variation (called CBCC-II), only neighbors with the same color are considered. As shown in Figure 4 (b), when a gray node is applying CBCC-II, all white nodes are excluded from its 2-hop information. The same rule also applies in white backbone construction, as shown in Figure 4 (c).

Compared to CBCC-I, CBCC-II uses smaller "Hello" messages to collect 2-hop information, has lower computation cost, and constructs a smaller backbone. However, the worst case performance and overhead of both variations are the same. Since CBCC-II is more aggressive than CBCC-I, its probability of constructing a k-CDS is lower than CBCC-I. As shown in Figure 4 (e), when node 3 uses CBCC-II to determine its status, it becomes a non-backbone node because it has only one visible neighbor 7. However, the resultant gray backbone $\{5, 7\}$ is not a CDS of the entire network, and union of all backbone nodes $\{5, 7, 8\}$ is not 2-dominating. The failure of node 8 will leave node 2 uncovered. Note that, however, when the network is very dense and node coverage is a good approximation of area coverage, the probability is high that CBCC-II selects a CDS of the entire network for each color, and the final backbone is a k-CDS.

5 Simulation

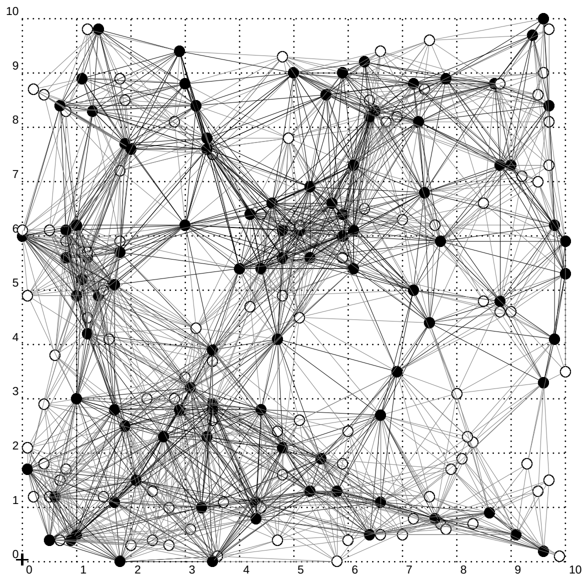
We conduct simulation study to evaluate the performance of three proposed k-CDS construction algorithms. Simulation results show that a small k-CDS can be formed with high probability and relatively low overhead in those schemes.

5.1 Implementation

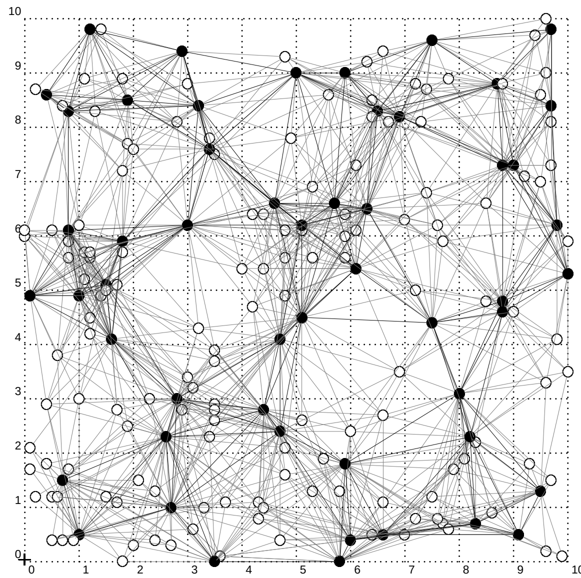
All proposed protocols have been implemented on a custom simulator *ds*[1]. All simulations are conducted in randomly generated static networks. To generate a network, n nodes are randomly placed in a $1000 \times 1000m^2$ region. The transmission range R is $250m$. Any two nodes with distance less than R are considered neighbors. Each simulation is repeated 500 times, and uses the average data as the final result. Both k-coverage condition and color based schemes use restricted 2-hop information to reduce computation overhead.

All k-CDS protocols, k-Gossip, k-coverage condition (k-Coverage), and two variations of the color-based coverage condition (CBCC-I and CBCC-II), are evaluated with $k = 2$ and 3, where the following performance metrics are compared:

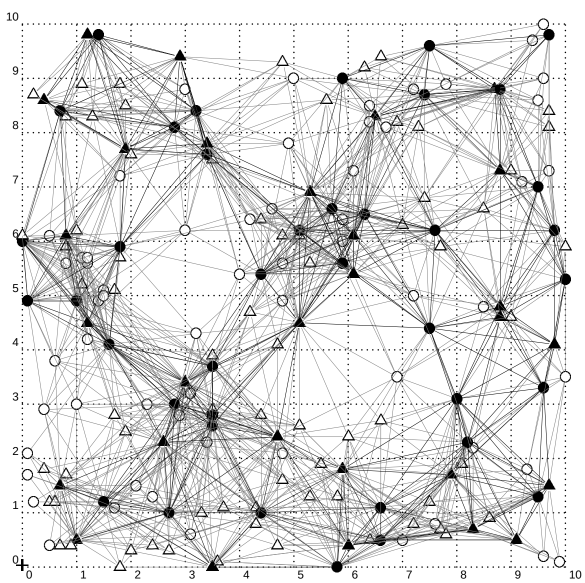
[1]The simulation code is downloadable from http://sourceforge.net/projects/wrss/



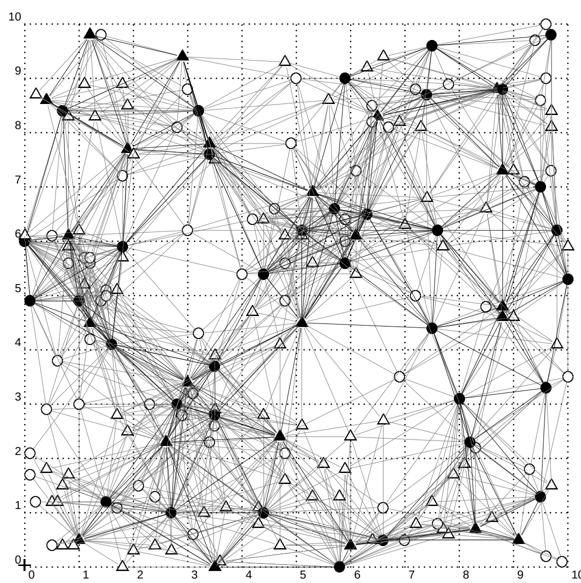
(a) k-Gossip, $p_k = 50\%$



(b) k-Coverage Condition



(c) CBCC-I



(d) CBCC-II

Figure 5. Sample virtual backbones constructed by different protocols with $k = 2$. The network consists of 200 nodes randomly and uniformly placed in a $1000 \times 1000m^2$ region. The transmission range is $250m$. Black nodes are backbone nodes and white nodes are non-backbone nodes. In color-based schemes (c,d), nodes in different colors are represented by circles and triangles, respectively.

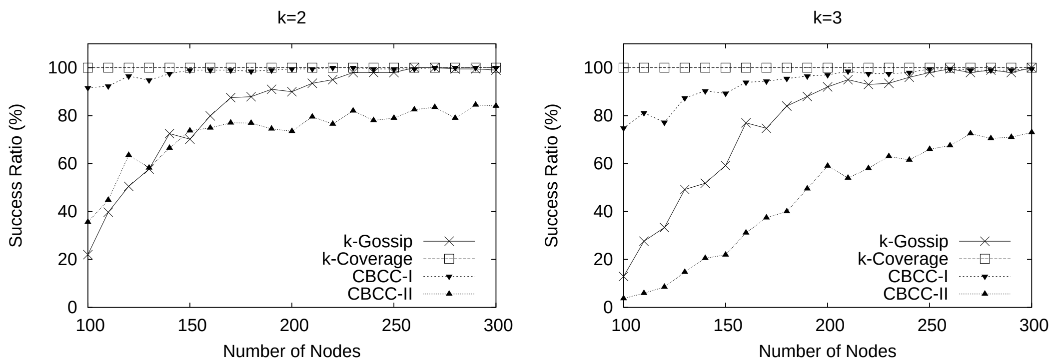


Figure 6. Success ratio.

- *Success Ratio*, defined as S/T, where T is total number of networks that are k-connected, and S is the count number that a protocol successfully forms a k-CDS of the network. High success ratio is essential for the reliability of a k-CDS protocol.

- *Backbone size*, i.e., average number of backbone nodes selected by a protocol. A smaller backbone size means lower bandwidth and energy consumption needed to maintain a k-CDS.

Figure 5 shows sample virtual backbones constructed by four protocols with $k - 2$ in a network with 200 nodes. We selected $p_k = 50\%$ in k-Gossip for high success ratio. The resultant virtual backbone consists of 101 nodes and a 2-CDS of the network (as shown in Figure 5 (a)). The k-coverage condition selects 53 nodes and also forms a 2-CDS (as shown in Figure 5 (b)). Both color based schemes divide the network into two equal partitions with different colors (represented by different node shapes). CBCC-I selects 68 backbone nodes and forms a 3-CDS (as shown in Figure 5 (c)). CBCC-I selects 58 backbone nodes and forms a 2-CDS (as shown in Figure 5 (d)). Overall, k-coverage condition has the smallest backbone size, and CBCC-I achieves the highest connectivity in this specific network.

5.2 Simulation results

Success ratio. Figure 6 compares the success ratio of four algorithms in constructing 2-CDS (the left graph) and 3-CDS (the right graph), when the node number n varies from 100 to 300. The probability p_k in k-Gossip is determined based on our previous experiment data in networks with 200 nodes (as shown in Figure 3). We assume that each node has no access to global information such as the value of n, and uses a fixed p_k in all networks. Here we chose $p_k = 50\%$ for $k = 2$ and $p_k = 69\%$ for $k = 3$.

As shown in Figure 6, k-Coverage has 100% success ratio in all circumstances, which confirms our claim in Theorem 1. That is, k-Coverage guarantees a k-CDS in all k-connected networks. CBCC-I has very high success ratio in relatively dense networks. For $k = 2$, it has 99% success ratio in networks with more than 150 nodes. For $k = 3$, its success ratio is larger than 97% when $n \geq 200$. Again, these results confirm our conclusion in Theorem 2: the original color-based scheme almost always forms a k-CDS in dense networks.

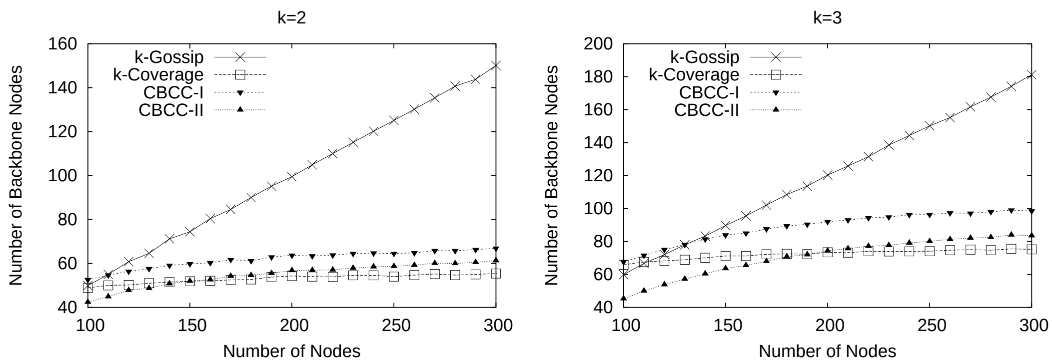


Figure 7. Backbone size.

The success ratio of k-Gossip is low in sparse networks. When $n = 100$, its success ratio is 22.0% when $k = 2$ and 12.9% when $k = 3$. However, its success ratio improves as the network density increases, and exceeds 90% after $n \geq 200$. CBCC-II has the lowest success ratio, except when $k = 2$ and $n \leq 150$. Its best performance is 84% for $k = 2$ and 73%. The assumption behind CBCC-II is that node coverage is a good approximation of area coverage in very dense networks. Obviously, the simulated networks are not sufficiently dense to make this scenario really happen.

Backbone size. Figure 7 compares virtual backbone size in 2-CDS (the left graph) and 3-CDS (the right graph) construction. k-Gossip usually produces the largest backbone. This is because we use a fixed p_k in the simulation, which selects np_k nodes on average. That is, the backbone size increases in the same speed of n. Using a variable p_k in k-Gossip is possible, but requires global information and experiment data to determine a perfect values of p_k for each network situation. The first requirement incurs extra runtime overhead, and the second increases the preparation cost.

The other three algorithms have a relatively small backbone size, which increases very slowly as n increases. Among them, k-Coverage has the best performance in dense networks, CBCC-II produces the smallest backbone in sparse networks, and CBCC-I has the worst performance in all scenarios. Since CBCC-II can merely maintain a k-CDS in sparse networks, k-Coverage is actually the best choice in terms of virtual backbone size. Our explanations to this phenomenon are: First, all coverage condition-based schemes seems to have probabilistic upper bound in dense networks (even though we cannot prove it for k-Coverage). Therefore, we will not see a proportional increase of the backbone size as in k-Gossip. Second, maintaining k separate 1-CDS's incurs higher redundancy than preserving a single k-CDS, which results in more backbone nodes in the color-based schemes.

Simulation results can be summarized as follows

1. k-Gossip has the lowest overhead and high success ratio in dense networks, but it also has serious problems. When a fixed p_k is used, it has a low success ratio in sparse networks and a large backbone size in dense networks.

2. k-Coverage guarantees 100% success ratio and selects a smallest backbone in most scenarios. Its

only weakness is the relatively complicated algorithm and high computation cost.

3. CBCC-I has lower overhead than k-Coverage, and almost always constructs a k-CDS in relatively dense networks. The resultant backbone size is larger than in k-Coverage, but much smaller than k-Gossip.

4. CBCC-II has lower overhead than CBCC-II, but does not show a satisfactory success ratio in our simulation. However, high success ratio may still be observed in very dense networks.

6 Conclusion

This paper proposes three localized protocols that construct a k-connected k-dominating set (k-CDS) as a virtual backbone of wireless networks. Two protocols are extensions of existing CDS algorithms. The third scheme is a generic paradigm, which enables many existing virtual backbone formation algorithms to produces a k-CDS with high probability. Our simulation results show that these protocols can select a small k-CDS with relatively low overhead. As future work, we plan to conduct extensive simulation study on the performance of k-CDS in carrying out important tasks such as routing and area monitoring. We will also try to find a probabilistic approximation ratio of the k-coverage condition (if one exists).

References

[1] C. Adjih, P. Jacquet, and L. Viennot. Computing connected dominated sets with multipoint relays. Technical Report 4597, INRIA-Rapport de recherche, Oct. 2002.

[2] K. M. Alzoubi, P.-J. Wan, and O. Frieder. Distributed heuristics for connected dominating sets in wireless ad hoc networks. *Journal of Communications and Networks*, 4(1):22–29, Mar. 2002.

[3] M. Bahramgiri, M. Hajiaghayi, and V. S. Mirrokni. Fault-tolerant and 3-dimensional distributed topology control algorithms in wireless multi-hop networks. In *Proc. of ICCCN*, pages 392–397, 2002.

[4] L. Bao and J. J. Garcia-Luna-Aceves. Topology management in ad hoc networks. In *Proceedings of MobiHoc*, pages 129–140, June 2003.

[5] P. Basu and J. Redi. Movement control algorithms for realization of fault tolerant ad hoc robot networks. *IEEE Networks*, 18(4):36–44, July/Aug. 2004.

[6] D. Blough, M. Leoncini, G. Resta, and P. Santi. The K-Neigh protocol for symmetric topology control in ad hoc networks. In *Proc. of MobiHoc*, pages 141–152, June 2003.

[7] J. Carle and D. Simplot-Ryl. Energy-efficient area monitoring for sensor networks. *IEEE Computer*, 37(2):40–46, Feb. 2004.

[8] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris. Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *ACM Wireless Networks Journal*, 8(5):481–494, Sep. 2002.

[9] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86:165–177, 1990.

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

[11] B. Das, R. Sivakumar, and V. Bhargavan. Routing in ad hoc networks using a spine. *Proc. of ICCCN*, pages 1–20, Sep. 1997.

[12] J. Edmonds. Paths, trees and flowers. *Can. J. Math.*, (17):449–467, 1965.

[13] A. Gibbons. *Algorithmic Graph Theory*. Cambridge University Press, 1985.

[14] Z. J. Haas, J. Y. Halpern, and L. Li. Gossip-based ad hoc routing. In *Proc. of INFOCOM*, volume 3, pages 1707–1716, June 2002.

[15] F. Harary and T. Haynes. Double domination in graphs. *ARS Combinatoria*, 55:201–213, 2000.

[16] X. Hou and D. Tipper. Gossip-based sleep protocol (GSP) for energy efficient routing in wireless ad hoc networks. In *Proc. of WCNC*, Mar. 2004.

[17] M. Jorgic, A. Nayak, and I. Stojmenovic. Localized algorithms for detection of k-connectivity in ad hoc networks. To appear.

[18] H. Koubaa and E. Fleury. On the performance of double domination in ad hoc networks. In *Proc. of IFIP Medhoc*, June 2000.

[19] F. Kuhn, T. Moscibroda, and R. Wattenhofer. Initializing newly deployed ad hoc and sensor networks. In *Proc. of MobiCom*, pages 260–274, Sep./Oct. 2004.

[20] S. Kumar, T. H. Lai, and J. Balogh. On k-coverage in a mostly sleeping sensor network. In *Proc. of MobiCom*, pages 144–158, Sep./Oct. 2004.

[21] N. Li and J. C. Hou. FLSS: a fault-tolerant topology control algorithm for wireless networks. In *Proc. of MobiCom*, pages 275–286, Sep./Oct. 2004.

[22] X.-Y. Li, P.-J. Wan, Y. Wang, and C.-W. Yi. Fault tolerant deployment and topology control in wireless networks. In *Proc. of MobiHoc*, pages 117–128, June 2003.

[23] C. S. Liao and G. J. Change. k-tuple domination in graphs. *Information Processing Letters*, 87(1):45–50, 2003.

[24] J. Liu and B. Li. MobileGrid: Capacity-aware topology control in mobile ad hoc networks. In *Proc. of ICCCN*, pages 570–574, Oct. 2002.

[25] K. Menger. Zur allgemeinen kurventheorie. *Fund. Math.*, (10):96–115, 1965.

[26] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying for flooding broadcast message in mobile wireless networks. In *Proc. of HICSS*, volume 9, page 298, Jan. 2002.

[27] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *Proc. of SenSys*, pages 28–39, Nov. 2003.

[28] J. Wu and F. Dai. A distributed formation of a virtual backbone in manets using adjustable transmission ranges. In *Proc. of ICDCS*, pages 372–379, Mar. 2004.

[29] J. Wu and F. Dai. A generic distributed broadcast scheme in ad hoc wireless networks. *IEEE Transactions on Computers*, 53(10):1343–1354, Oct. 2004.

[30] J. Wu and H. Li. On calculating connected dominating sets for efficient routing in ad hoc wireless networks. *Proc. of DialM*, pages 7–14, 1999.