Handbook of Combinatorial Optimization
D.-Z. Du and P. Pardalos (Eds.) pp. 329 - 369

Connected Dominating Set in Sensor Networks and MANETs

Jeremy Blum Min Ding Andrew Thaeler Xiuzhen Cheng
Department of Computer Science
The George Washington University, Washington, DC 20002
E-mail: {blumj,minding,athaeler,cheng}@gwu.edu

Contents

1 Introduction

Recently developed classes of wireless networks have blurred the distinction between the network infrastructure and network clients. Sensor networks, for example, consist of one or more base stations and a large number of inexpensive nodes, which combine sensors and low power wireless radios. Due to limited radio range and battery power, most nodes cannot communicate directly with a base station, but rather rely on their peers to forward messages to and from base stations. Likewise, in mobile ad hoc networks (MANETs), the routing of messages is also performed by ordinary nodes. In fact, a MANET typically has no network infrastructure, therefore all routing and network management functions must be performed by ordinary nodes.

The key to scalability and efficiency in traditional computer networks is the organization of the network infrastructure into a hierarchical structure. However, due to the lack of a network infrastructure, sensor networks and MANETs are inherently flat. In order to achieve scalability and efficiency, new algorithms have emerged that rely on a virtual network infrastructure, which organizes ordinary nodes into a hierarchy. The construction of this infrastructure is the primary application of Connected Dominating Sets (CDSs) in wireless networks.

The utility of CDSs in wireless ad hoc networks has been demonstrated in protocols that perform a wide range of communication functions. CDSs have formed an underlying architecture used by protocols including media access coordination [8, 43, 64]; unicast [35, 33, 34, 76], multicast/broadcast [51, 52, 67, 73, 78, 79, 80], and location-based routing [36]; energy conservation

[19, 38, 63, 72, 82]; and topology control [37, 38]. CDS can also be used to facilitate resource discovery in MANET [45, 48].

In this chapter, we are going to survey the CDS construction techniques proposed in the context of sensor networks and MANETs. Sections 3 and 4 address details of centralized and distributed algorithms respectively. Theoretically any centralized algorithm can be implemented in a distributed fashion, with the tradeoff of higher protocol overhead. We are going to examine several centralized algorithms and their corresponding distributed implementations in detail. Distributed or even localized algorithms are very important for sensor networks and MANETs. CDS must be constructed efficiently to be applicable in a mobile or large scale network. Due to the dynamism of wireless links and nodal mobility, algorithms should rely on limited knowledge of the current network topology.

Note that the design goals of different algorithms vary based on the needs of the protocols making use of the CDS. When designing a CDS algorithm, one must take the following parameters into consideration: performance bounds, degree of localization, time and message complexities, and stability with respect to nodal movement. We are going to analyze these algorithms in Section 5. We will provide a toolbox of techniques (Subsection 5.1) elicited from the examination of CDS construction algorithms, and present guidelines (Subsection 5.2) to aid in the selection of particular techniques based on the design goals of an application.

Actually many works seek a *minimum connected dominating set (MCDS)* in unit-disk graphs as their major design goal. Thus performance bounds is their primary design parameter. The rationale of this problem formulation can be justified as follows. The foot print of an ad hoc network with fixed transmission range for each host can be modelled by a unit-disk graph [25]. And minimizing the cardinality of the computed CDS can help to decrease the control overhead since broadcasting for route discovery [47, 60] and topology update [30] is restricted to a small subset of nodes [25]. Therefore *broadcast storm problem* [59] inherent to global flooding can be greatly decreased.

Other works seek a connected dominating set that provides good resource conservation property [19, 76]. Thus performance bound is not their primary consideration. Instead, the hop count of communication path between nodes is taken into consideration for load balance [19] and power conservation [78, 79]. In the following section (Section 2), we will present network model and useful definitions needed for algorithm elaboration. We also will give an overview of CDS applications.

2 Network Model and CDS Applications

The following section provides background information for the analysis of CDS applications in ad hoc wireless networks. We first present a mathematical model for the networks under consideration and introduce useful terminologies and definitions from graph theory. Then we sketch the various wireless network applications utilizing connected dominating sets.

2.1 Network Model

An ad hoc wireless network can be represented by a graph $G(V, E^t)$ comprised of a set of vertices V and time-varying edges E^t. For each pair of vertices $u, v \in V$, $(u, v) \in E^t$ if and only if the nodes u and v are within communication range. Due to nodal movement, the topology of the network is dynamic, as reflected by E^t.

Given omni-directional antennae, the communication range of a node in a wireless network is typically modelled as a disk centered at the node with radius equal to the transmission range of the radio. Consequently, when transmission range is fixed for all nodes, the network has the property of a unit-disk graph (UDG), where an edge exists if and only if two nodes have inter-nodal distance less than or equal to 1 unit (the fixed communication range). Many of the CDS algorithms use the properties of UDG's to prove their performance bounds.

Each node v has an associated set of nodal properties. Typical properties include the following:

- ID_v, the unique ID for node v.
- loc_v^t, the location of node v at time t.
- $velocity_v^t$, the velocity vector for node v at time t.

A number of definitions from graph theory are used in this chapter. Figure 1 can help to illustrate the following concepts:

- *Open Neighbor Set*, $N(u) = \{v \mid (u, v) \in E\}$, is the set of nodes that are neighbors of u. In Figure 1, the open neighbor set of e is $\{d, f, g\}$.
- *Closed Neighbor Set*, $N[u] = N(u) \cup \{u\}$, is the set of neighbors of u and u itself. In Figure 1, the closed neighbor set of e is $\{d, e, f, g\}$.
- *Maximum Degree*, Δ, is the maximum count of edges emanating from a single node. The maximum degree of the graph in Figure 1 is three, and occurs at nodes c, e, and g.

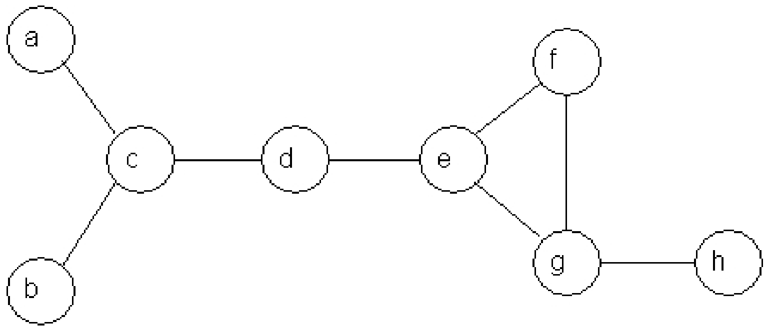

Figure 1: Representation of a wireless network with eight nodes as a graph.

- *Independent Set*, is a subset of V such that no two vertices within the set are adjacent in V. For example, $\{a, b, f, h\}$ is an independent set in Figure 1.
- *Maximal Independent Set (MIS)*, is an independent set such that adding any vertex not in the set breaks the independence property of the set. Thus, any vertex outside of the maximal independent set must be adjacent to some node in the set. The previous independent set $\{a, b, f, h\}$ must have node d added to become an MIS.
- *Dominating Set*, S, is defined as a subset of V such that each node in $V - S$ is adjacent to at least one node in S. Thus, every MIS is a dominating set. However, since nodes in a dominating set may be adjacent to each other, not every dominating set is an MIS. Finding a minimum-sized dominating set or MDS is NP-Hard [41].
- *Connected Dominating Set (CDS)*, C, is a dominating set of G which induces a connected subgraph of G. One approach to constructing a CDS is to find an MIS, and then add additional vertices as needed to connect the nodes in the MIS. A CDS in Figure 1 is $\{c, d, e, g\}$.
- *Minimum Connected Dominating Set* (MCDS) is the CDS with minimum cardinality. Given that finding minimum sized dominating set is NP-Hard, it should not be surprising that finding the MCDS is also NP-Hard [41]. In Figure 1, $\{c, g\}$ is a minimum connected dominating set.
- *Weakly Connected Dominated Set (WCDS)*, S, is a dominating set such that $N[S]$ induces a connected subgraph of G. In other words, the

subgraph weakly induced by S is the graph induced by the vertex set containing S and its neighbors. Given a connected graph G, all of the dominating sets of G are weakly connected. Computing a minimum WCDS is NP-Hard [41].

- *Steiner Tree*, is a minimum weight tree connecting a given set of vertices in a weighted graph. After finding an MIS, connecting the nodes together could be formulated as an instance of the Steiner Tree problem. Like many of the other problems that arise in CDS construction, this problem is NP-Hard [41].

2.2 Applications of CDS in Wireless Networks

Sensor networks and MANETs have unique characteristics that require the development of protocols specific to them. For efficiency reasons, many of these protocols first organize the network through the construction of dominating sets. These protocols address media access, routing, power management, and topology control.

At the Link Layer, clustering can increase spatial reuse of the spectrum, minimize collisions, and provide Quality of Service (QoS) guarantees [8, 43, 64, 65]. Correspondingly, the nodes in the dominating set can coordinate with one another and use orthogonal spreading codes in their neighborhoods to improve spatial reuse with code division spread spectrum techniques [42]. Furthermore, these nodes can coordinate access to the wireless media by their neighbors for QoS or collision avoidance purposes.

As first noted by Ephremedis *et al.*, a CDS can create a virtual network backbone for packet routing and control [40]. Messages can be routed from the source to a neighbor in the dominating set, along the CDS to the dominating set member closest to the destination node, and then finally to the destination. This is termed *dominating set based routing* [33, 76], or *Backbone based rouing* [34], or *spine based routing* [35, 66]. Restricting the routing to the CDS results in a significant reduction in message overhead associated with routing updates [18]. Furthermore, the dominating set can be organized into a hierarchy to further reduce control message overhead [56, 64, 65].

A CDS is also useful for location-based routing. In location-based routing, messages are forwarded based on the geographical coordinates of the hosts, rather than topological connectivity. Intermediate nodes are selected based on their proximity to the message's destination. With this scheme, it is possible for a message to reach a local maximum, where it has been sent to an intermediate node whose neighbors are all further from the destination than itself. In this case, the routing must enter a recovery phase, where the route may backtrack to find another path. However, if messages are only forwarded to nodes in the dominating set, the inefficiency associated with this recovery phase can be greatly reduced [36].

The efficiency of multicast/broadcast routing can also be improved through the utilization of CDSs. A big problem in multicast/broadcast routing is that many intermediate nodes unnecessarily forward a message. Nodes often hear the same message multiple times. This is the *broadcast storm problem* [59]. If the message is routed along a CDS, most of the redundant broadcasts can be eliminated [25, 51, 52, 67, 73, 78, 79, 80].

Nodes in a wireless network often have a limited energy supply. CDSs play an important role in power management. They have been used to increase the number of nodes that can be in an sleep mode, while still preserving the ability of the network to forward messages [19, 38, 82]. They have also been used to balance the network management requirements to conserve energy among nodes [63, 72, 77, 78, 79, 80].

In large-scale dense sensor networks, sensor topology information extraction can be handled by CDS construction [37, 38]. Other than routing, the virtual backbone formed by dominating set can also be used to propagate "link quality" information for route selection for multimedia traffic [64], or to serve as database servers [49], etc.

3 Centralized CDS Construction

The first instance of a dominating set problem arose in the 1850's, well before the advent of wireless networks [70]. The objective of the five queens problem is to find the minimum number of queens that can be placed on a chessboard such that all squares are either attacked or occupied by a queen. This problem was formulated as a dominating set of a graph $G(V,E)$, with the vertices corresponding to squares on the chessboard, and $(u, v) \in E$ if and only if a queen can move from the square corresponding to u to the square corresponding to v.

MCDS in general graphs was studied in [41], in which a reduction from the *Set Cover Problem* [41] to the MCDS problem was shown. This result implies that for any fixed $0 < \epsilon < 1$, no polynomial time algorithm with performance ratio $\leq (1-\epsilon)H(\Delta)$ exists unless $NP \subset DTIME[n^{O(log\ log\ n)}]$ [54], where Δ is the maximum degree of the input graph and H is the harmonic function. The MCDS remains NP-hard [29] for unit-disk graphs.

MCDS in unit-disk graphs has constant performance ratio, as proved by [4, 16, 23, 69]. A polynomial time approximation scheme for computing a MCDS in unit-disk graphs has been developed by Cheng *et al.* in [26]. A significant impact of this result is that a MCDS in unit-disk graphs can be approximated to any degree if computing time is permitted. Note that heuristics proposed for unit-disk graphs work well for general graphs, but their performance analysis is unapplicable. Thus in this section and next, we will focus on algorithm description and skip the corresponding performance analysis. Note that we intentionally omit the fact that these algorithms are proposed in either unit-disk graphs or general graphs because they are actually applicable in both graph models. Also note that we are going to use either the name of the first author or all authors' names of the paper to represent the algorithm.

In the following we will focus on centralized CDS construction algorithms. Distributed heuristics will be discussed in Section 4.

3.1 Guha and Khuller's Algorithm

In 1998, Guha and Khuller proposed two CDS construction strategies in their seminal work [44], which contains two greedy heuristic algorithms with bounded performance guarantees. In the first algorithm, the CDS is grown from one node outward. In the second algorithm, a WCDS is constructed, and then intermediate nodes are selected to create a CDS. The distributed implementations of both algorithms were provided by Das *et al.* in [33], which will be addressed in Subsection 4.2. Many algorithms designed latter [23, 69] are motivated by either of these two heuristics. We sketch the procedures in the following.

The first algorithm begins by marking all vertices white. Initially, the algorithm selects the node with the maximal number of white neighbors. The selected vertex is marked black and its neighbors are marked gray. The algorithm then iteratively scans the gray nodes and their white neighbors, and selects the gray node or the pair of nodes (a gray node and one of its white neighbors), whichever has the maximal number of white neighbors. The selected node or the selected pair of nodes are marked black, with their white neighbors marked gray. Once all of the vertices are marked gray or black, the algorithm terminates. All the black nodes form a connected dominating set. This algorithm yields a CDS of size at most $2(1 + H(\Delta)) \cdot |OPT|$, where H is the harmonic function, and OPT refers to an optimal solution – that is, a minimum connected dominating set.

For example, consider the graph in Figure 2. Initially, either node c, or e,

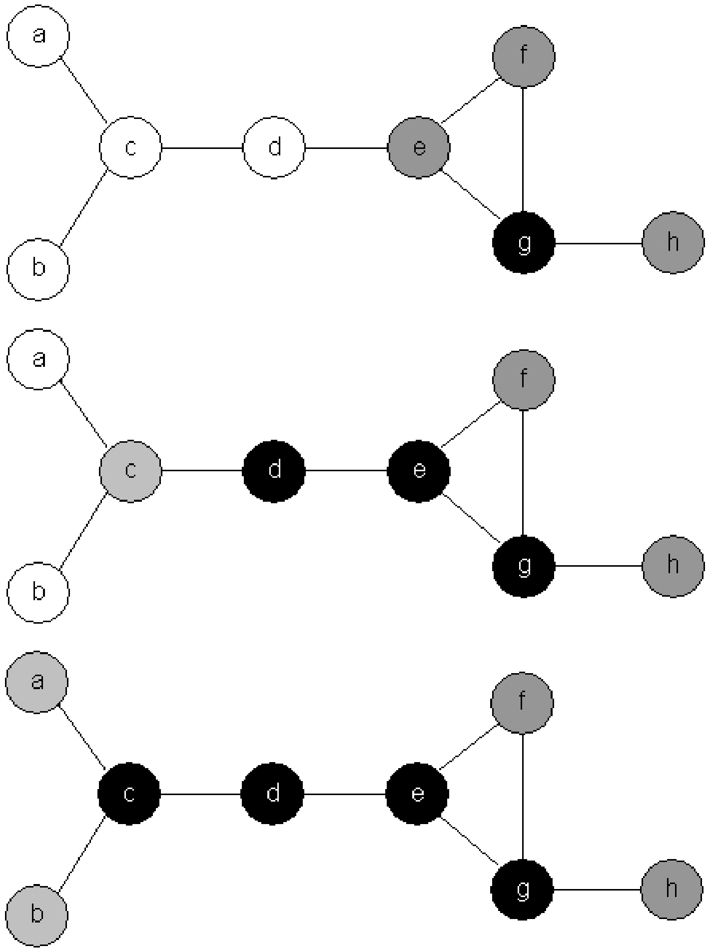

Figure 2: An example of Guha's first algorithm.

or g could be marked since they have maximal degree. Node g is arbitrarily picked from these candidates and marked black. All its neighbors are then marked gray. We now consider the gray nodes e, f, h, and the pair of gray and white nodes (d, e). Of all these, the pair (d, e) covers the most number of white neighbors - two. So we mark both d and e black. Finally, we consider the gray node c and the pairs (c, a) and (c, b). All these candidates have the same number of white neighbors. Therefore the single node c is selected. Now all the nodes are either black or gray, and the set of nodes in black $\{c, d, e, g\}$ forms a CDS.

The second algorithm also begins by coloring all nodes white. A *piece* is defined to be either a connected black component, or a white node. The algorithm contains two phases. The first phase iteratively selects a node that causes the maximum reduction of the number of pieces. In other words, the greedy choice for each step in the first phase is the node that can decrease the maximum number of pieces. Once a node is selected, it is marked black and its white neighbors are marked gray. The first phase terminates when no white node left. After the first phase, there exists at most $|OPT|$ number of connected black components. The second phase constructs a Steiner Tree that connects all the black nodes by coloring chains of two gray nodes black. The size of the resulting CDS formed by all black nodes is at most $(3 + ln(\Delta)) \cdot |OPT|$.

Figure 3 shows an example of the second algorithm. First, node g is marked as it is one of the nodes with the maximum number of white neighbors. Next, node c is marked because it can reduce the maximum number of pieces compared with any other node. Now the first phase ends as there is no white node left. In the second phase, a Steiner Tree is constructed by adding nodes d and e to connect nodes c and g.

3.2 Ruan's Algorithm

The potential function used in the second algorithm of Guha and Khuller [44] is the number of pieces. Each step seeks maximum reduction in the number of pieces in the first phase. By modifying the potential function, Ruan *et al.* [62] proposes a one-step greedy approximation algorithm with performance ratio at most $3+ln(\Delta)$. This algorithm also requires each node to be colored white at the beginning. If there exists a white or gray node such that coloring it black and its white neighbors gray would reduce the potential function, then choose the one that causes maximum reduction in the potential function.

The potential function plays a critical rule in this algorithm. It is defined

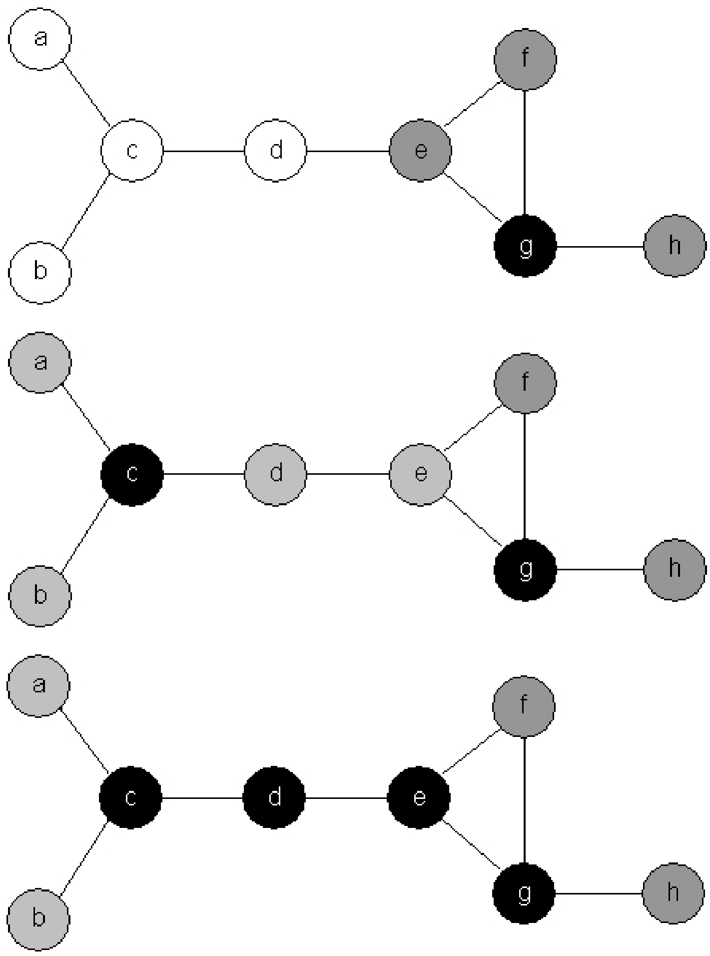

Figure 3: An example of Guha's second algorithm.

in the following way in [62]. Given a connected graph $G(V, E)$, define $p(C)$ be the number of connected black components in the subgraph induced by $C \subset V$. Let $D(C)$ be the set of all edges incident to vertices in C. Define $q(C)$ be the number of connected components in the subgraph $G(V, D(C))$. Then the potential function is defined to be $f(C) = p(C) + q(C)$.

In the greedy algorithm, let C be the set containing all black nodes. Thus initially $f(C) = |V|$ since $C = \phi$. The first step chooses a node x with maximum degree. Every other step selects a node x such that $f(C) - f(C \cup \{x\})$ is maximized. Color node x black and color all its white neighbors gray. The algorithm ends when $f(C) = 2$, where C is the resultant CDS.

3.3 Cheng's Greedy Algorithm

In [24], Cheng *et al.* propose a greedy algorithm for MCDS in unit-disk graphs. Compared to the many heuristics discussed in Subsection 4.3, this algorithm relies on an MIS but the resultant CDS may not contain all the elements in the MIS.

Assume initially all nodes are colored white. The construction of a CDS contains four phases. In the first phase, an MIS is computed and all its members are colored red. In the second phase, a node that can decrease the maximum number of pieces is selected, where a piece is either a red node, or a connected black component. This node is colored black and all its non-black neighbors are colored gray. When the second phase is over, we still have some white nodes left. The third phase will compute a spanning tree for each connected component in the subgraph reduced by all white nodes. Connect each tree to the nearest black component with black nodes accordingly. All non-leaf tree nodes are colored black while leaf nodes are colored gray. The last phase will seek chains of two gray nodes to connect disjoint black components.

The motivation of Cheng's algorithm are two fold. First, the greedy choice in Guha and Khuller's second algorithm [44] is the one that can decrease the maximum number of pieces, where a piece is either a connected black component, or a white node. Second, a unit-disk graph has at most 5 independent neighbors. Thus intuitively one can choose the greedy choice that can connect to as many independent nodes as possible. In other words, the node to be colored black at each step will try to cover more uncovered area, if we model vertices in a unit-disk graph as nodes in a flat area. Unfortunately Cheng's algorithm does not have a solid performance analysis.

3.4 Min's Algorithm

Recently Min *et al.* [58] propose to use a *Steiner tree with minimum number of Steiner nodes (ST-MSN)* [20, 39, 53] to connect a maximal independent set. This algorithm contains two phases. The first phase constructs an MIS with the following property: every subset of the MIS is two hops away from its complement. Color all nodes in the MIS black; color all other nodes gray. In the second phase, a gray node that is adjacent to at least three connected black components is colored black in each step. If no node satisfying this condition can be found, a gray node that is adjacent to at least two connected black components will be colored black. This algorithm has performance ratio 6.8 for unit-disk graphs. Its distributed implementation is sketched in Subsection 4.6.

ST-MSN in Euclidean plane is NP-hard [53]. A 3-approximation algorithm for ST-MSN in Euclidean plane is proposed in [20] and is extended to unit-disk graphs by Min *et al.* in [58]. Since the size of any MIS is at most $3.8 \cdot |OPT| + 1.2$ [81] in unit-disk graphs, where OPT is any MCDS, the computed CDS by Min's algorithm has size at most $6.8 \cdot OPT$ [58].

3.5 Butenko's Algorithm

The heuristic proposed in [14, 15] is pruning-based. In other words, the connected dominating set S is initialized to the vertex set of graph $G(V, E)$, and each node will be examined to determine whether it should be removed or retained. Assume all nodes in S are colored white at the beginning. Define the *effective degree* of a node to be its white neighbors in S. Consider a white node $x \in S$ with minimum effective degree. If removing x from S makes the induced graph of S disconnected, then retain x and color it black. Otherwise, remove x from S. At the same time, if x does not have a black neighbor in S, color its neighbor with maximum effective degree in S black. Repeat this procedure until no white node left in S. This algorithm has time complexity $O(|V| \cdot |E|)$. It does not have a performance analysis. We will discuss its distributed implementation in Subsection 4.4.2.

4 Distributed CDS Construction

For sensor networks and MANETs, distributed CDS construction is more effective due to the lack of a centralized administration. On the other hand, the large problem size (e.g. a sensor network may contain hundreds of thousands of sensors) also prohibits the centralized CDS computation. In this section, we survey a variety of distributed approaches that seek to balance the competing requirements of complexity, running time, stability, and overhead. Note that many published results contain algorithms that differ very little from each other. To be more focus, we decide only to cover the major distributed CDS construction techniques.

In wireless networks, CDS problems have been formulated in a number of ways, depending on the needs of the particular application. These formulations can be classified into WCDSs, non-localized CDSs, localized CDSs, and stable CDSs with nodal mobility. In this chapter, we choose to use a different classification containing the following categories based on the CDS construction techniques: WCDS, greedy CDS, MIS based CDS, pruning based CDS, multipoint forwarding based CDS, Steiner tree based CDS, and stable CDS. In the following, we will examine each category and explore example algorithms in detail. We also will sketch the maintenance of the computed CDS if available.

4.1 WCDS Construction

In a WCDS, the vertex set is partitioned into a set of clusterheads and cluster members, such that each cluster member is within radio range of at least one clusterhead. There are two ways of approaching this problem. If nodal location information is known, then the nodes can be clustered geographically. Otherwise, nodes can be clustered based solely on the graph topology. Chen and Liestman [21] propose a series of approximate algorithms for computing a small WCDS to be used to cluster mobile ad hoc networks.

Geographical clustering algorithms create a virtual grid in the geographical region where the network exists. Each cluster comprises all of the nodes in a given grid. The Grid Location Service, for example, uses this grid structure to disseminate the location information of nodes throughout the network. Using the grid structure, the density of nodes holding the location information of other nodes decreases as the distance from the node increases [50]. Geographical Adaptive Fidelity attempts to minimize energy consumption in a network by clustering the nodes based on a grid, and having only one node per grid responsible for routing at any given time [82].

In the first WCDS algorithm for wireless networks, nodes elect their neighbor with the lowest ID as their clusterhead [40]. Nodes learn about the ID's of their current neighbors through periodic beacons that each node broadcasts. Whenever a node with a lower ID moves into range of a clusterhead, it becomes the new clusterhead.

Highest Connectivity Clustering presents an improvement over Lowest ID Clustering by reducing the number of clusters [43]. In addition to periodically broadcasting its ID, each node also broadcasts the size of N(u), its open neighbor set. A node becomes clusterhead if it is the most highly connected among its neighbors. When nodes are moving, this selection criterion is less stable than Lowest ID Clustering, since ID's do not change, but nodal degree changes often.

One potential problem with Lowest ID Clustering is its unfairness. Nodes in the dominating set may be burdened with additional responsibilities. To be more equitable in clusterhead selection, the Min-Max D-Cluster algorithm selects nodes based on ID, but attempts to elect both low and high ID nodes [7]. Another interesting feature of the algorithm is the creation of d-hop dominating sets where each node is at most d hops from a node in the dominating set.

A serious problem with the Lowest ID Clustering is that clusterhead selection can be very unstable. The left side of Figure 4 displays a cluster headed by Node 2. As shown on the right side, once Node 1 moves into radio range, Node 2's cluster will be disbanded and replaced by three clusters. This reorganization is unnecessary, since Node 1 could have simply joined the existing cluster.

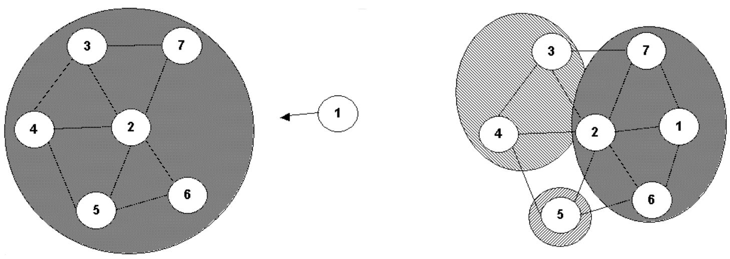

Figure 4: Instability of Lowest ID Clustering. As shown on the left, Node 1 is moving into range of Node 2's cluster. Once Node 1 is in radio range, Node 2's cluster will be reorganized.

In order to address this instability, the clustering algorithm introduced by the Clusterhead-Gateway Switch Routing Protocol preserves clusterhead elections under all but two conditions [27]. The set of clusterheads changes only when two clusterheads come into contact with each other, or when a node loses contact with all clusterheads.

4.2 Greedy CDS Construction

Das *et al.* [35, 33, 66] propose the distributed implementations of the two greedy algorithms given by Guha and Khuller in [44].

The first algorithm grows a CDS from one node with maximum degree. Thus the first step involves selecting the node with the highest degree. Therefore a node must know the degree of all nodes in the graph. On the other hand, each iterative step selects either a one- or two-edged path emanating from the current CDS, thus the nodes in the CDS must know the number of unmarked neighbors for all nodes one and two hops from the CDS. These two requirements force the flooding of degree information in the network. This algorithm generates a CDS with approximation ratio of $2H(\Delta)$ in $O(|C|(\Delta + |C|))$ time, using the $O(n|C|)$ messages, where the harmonic function $H(\Delta) = \sum_{i=1}^{\Delta} 1/i \leq \ln(\Delta) + 1$, n is the total number of vertices, and C represents the generated CDS.

The second algorithm first computes a dominating set and then selects additional nodes to connect the set. In order to locally select nodes for the dominating set in the first stage, an unmarked node compares its effective degree, the number of unmarked neighbors, with the effective degrees of all its neighbors in two-hop neighborhood. The greedy algorithm iteratively adds the node with maximum effective degree to the dominating set. The first stage terminates when a dominating set is achieved. The second stage connects the obtained components using a distributed minimum spanning tree algorithm, with the goal of adding as few nodes as possible. To do this, each edge is assigned a weight equal to the number of endpoints not in the dominating set. At the end, the interior nodes in the resulting spanning tree compose a connected dominating set. This algorithm has time complexity of $O((n+|C|)\Delta)$, and message complexity of $O(n|C|+m+n\log(n))$. It approximates the MCDS with a ratio of $2H(\Delta)+1$, where m is the cardinality of the edge set.

Das *et al.* [35, 33, 66] handle CDS maintenance differently in the case of single-node movement versus multiple-node movement. If a single node moves, the CDS can be updated locally. When more than one node moves, the moves can be treated as many single-node moves if they have no overlapping neighborhoods. However, an entirely new CDS computation may be needed in the case of overlapping neighborhoods.

4.3 MIS Based CDS Construction

Algorithms in this category compute and connect an MIS. But how an MIS is computed and connected differs from algorithm to algorithm. One can compute an MIS based on either single leader [2, 3, 4, 5, 13, 16, 23, 69] or multiple leaders [6, 25]. The MIS can be connected after the construction is over [2, 3, 4, 5, 6, 13, 16, 23, 25, 69], or one can compute and connect an MIS simultaneously [23, 25]. Note that single leader based MIS construction needs a leader-election algorithm, which takes $O(n\ log\ n)$ messages [10, 28]. Nodes with maximum degree or id among all neighbors can serve as leaders in multiple leader based MIS construction, thus the corresponding algorithms have lower message complexity [6, 25].

Note that the algorithm provided by [58] also relies on an MIS. But we will address it in detail in Subsection 4.6, as the major contribution in [58] is the exploitation of a Steiner tree with minimum number of Steiner points to connect an MIS.

4.3.1 Alzoubi and Wan's Single Leader Algorithm

Alzoubi and Wan's algorithms [4, 5, 69] utilize the properties of unit-disk graphs (UDGs) to prove their performance bounds. By definition, an MIS is adjacent to every node in the graph. Due to the geographic constraints imposed by a UDG, a node is adjacent to at most five independent neighbors [55]. Therefore, an arbitrary MIS can contain no more than five times the number of nodes in the minimum-sized MIS. This observation forms the basis of the proof of the performance bounds for all algorithms in [4, 5, 69].

Alzoubi *et al.* [4, 5, 69] provide two versions of an algorithm to construct the dominating set for a wireless network. In both algorithms, they first employ the distributed leader election algorithm [28] to construct a rooted spanning tree from the original network topology. Then, an iterative labelling strategy is used to classify the nodes in the tree to be either black (dominator) or gray (dominatee), based on their ranks. The rank of a node is the ordered pair of its level (number of hops to the root of the spanning tree) and its ID.

The labelling process begins from the root node and finishes at the leaves. The node with the lowest rank marks itself black and broadcasts a DOMINATOR message. The marking process then continues according to the following rules:

- If the first message that a node receives is a DOMINATOR message, it marks itself gray and broadcasts a DOMINATEE message.

- If a node received DOMINATEE messages from all its lower rank neighbors, it marks itself black and sends a dominator message.

The marking process finishes when it reaches the leaf nodes. At that time, the set of black nodes form an MIS, incorporating alternate levels of the spanning tree. Since the nodes are on alternating levels of the spanning tree, the distance between any subset of the MIS and its complement is exactly two hops away.

The final phase connects the nodes in the MIS to form a CDS, using INVITE and JOIN messages. Initially, the root joins the CDS and broadcasts an INVITE message. The INVITE message is relayed to all two-hop neighbors out of the current CDS. When a black node receives the INVITE message for the first time, it joins the dominating tree together with the gray node, which relayed the message. It then initiates an INVITE message. The process terminates when all the black nodes join the CDS.

This algorithm has time complexity of $O(n)$, and message complexity of $O(n \log(n))$. The resulting CDS has a size of at most $8opt + 1$.

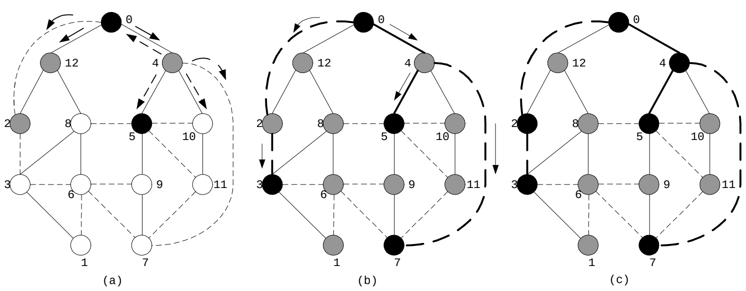

Figure 5: An example of Alzoubi and Wan's single leader algorithms for CDS construction.

Figure 5 illustrates the action of these algorithms. In this graph, node 0 is the root of the spanning tree that is constructed by using the leader election algorithm. The solid lines represent the edges of the rooted spanning tree, and the dashed lines represent other edges in the UDG. Node 0 is marked black first and broadcasts a DOMINATOR message (solid arrows in Figure 5(a)). After receiving this message, nodes 2, 4, and 12 are marked gray and broadcast DOMINATEE messages. (For simplicity, only the DOMINATEE messages from node 4 are shown as the dashed arrows in Figure 5(a)). Then node 5 is selected to be a DOMINATOR as it has received DOMINATEE messages from all its lower rank neighbors (node 4 only). Figure 5(b) shows the colors of the nodes when the labelling process finishes. The final process builds the dominating tree from the root. The INVITE message (solid arrow in Figure 5(b)) is sent from node 0, and it is relayed to its two-hop black neighbors 3, 5 and 7. These black nodes join the dominating tree, as well as their relaying gray nodes 2 and 4. The thick links in Figure 5(b) illustrate the edges in the final dominating tree. All the nodes in the tree form a connected dominating set. The improved approach in [5] merges the MIS construction with the dominating tree building processes. As shown in Figure 5(c), node 2 is colored black when it first receives a DOMINATOR message from its child 3, and node 4 is marked black for the same reason. Finally, all the black nodes form a connected dominating set.

4.3.2 Cheng's Single Leader Algorithm

Cheng *et al.* [13, 16, 25, 23] present two algorithms for growing a connected dominating set from a leader node. Compared with the work of Alzoubi *et al.* [4, 5, 69], they introduce a new ***active*** state for vertices to describe the current labelling set of vertex nodes. With the help of this new concept, either cost-aware or degree-aware optimization can be achieved.

Their first algorithm is cost-aware. Each host has a local cost, which serves as the selection criterion together with its ID. At the beginning, all vertices are in initial state with white color. The leader starts the algorithm by marking itself black and becoming a dominator. A white node goes to be a dominatee (gray) if one of its neighbors becomes a dominator. A non-active white vertex changes to status ***active*** if one of its neighbors becomes a dominatee. Its color still keeps white. Then, an ***active*** node with the smallest cost among all its active neighbors will compete to be a dominator. Its minimum cost gray parent also changes to serve as its dominator (black), ensuring the connectivity of the dominating tree. Finally, all black leaf nodes can change back to be dominatees (gray). This process terminates when all nodes are colored gray or black, and all the black nodes form a connected dominating set.

This algorithm has the time complexity of $O(n)$, and the message complexity of $O(n \; log \; n)$, which is dominated by leader election. The performance ratio is $8opt + 1$, the same as that of the algorithm in [4].

An example that illustrates the application of Cheng's single leader algorithm is shown in Figure 6. There are 9 hosts and 12 links. We assume host IDs are the costs. Host 0 is the leader. In the beginning, node 0 is colored black, serving as a dominator. Nodes 1 and 5 are then colored gray. Nodes 2 and 6 become active. Their competition results in the winner of node 2. Node 2 colors itself black and invites node 1 to be its dominator.

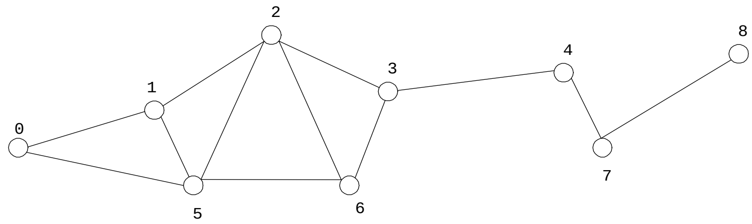

Figure 6: An example of unit-disk graph G containing 9 hosts and 12 links.

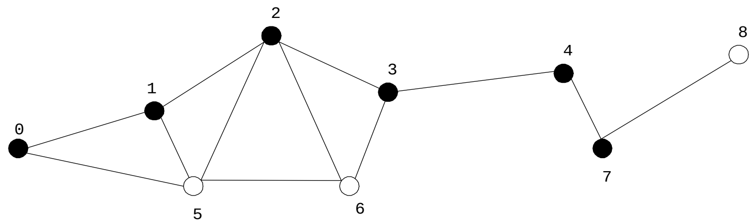

Figure 7: The computed connected dominating set from Cheng's single leader algorithm contains hosts $\{0, 1, 2, 3, 4, 7\}$. The optimal solution contains $\{1, 2, 3, 4, 7\}$

Therefore node 3 is colored gray and node 4 becomes active. This process continues until no white nodes left. The result is demonstrated in Figure 7. All black nodes form a CDS.

Note that in this algorithm, a dominating tree is grown from the leader. The resultant CDS contains two subsets: one changes color from white to black directly, and the other colors themselves from white to gray then to black. If the determination of the second subset is delayed until the first subset is constructed, and the criterion for changing color from gray to black is based on the number of black neighbors a gray node has, the degree-based algorithm described in [23] is obtained. This algorithm has performance ratio 8. However, since effective degree information (number of white neighbors) needs to be updated during the algorithm execution, the degree-aware algorithm takes higher number of messages compared to the cost-aware algorithm, even though their time and message complexities are the same. Min and Du [57] extends the second algorithm to consider reliable virtual backbone construction in MANET.

4.3.3 Alzoubi's Multiple Leader algorithm

Single leader based CDS construction has message complexity $\Omega(n\ log\ n)$, which is dominated by leader election. In single leader CDS construction, the dependence on a rooted tree renders the maintenance of the CDS extremely difficult. To decrease the protocol overhead caused by the large volume of message exchange, and to improve the structure of the computed CDS, multiple leader based algorithms [6, 25] are proposed. Compared with single leader based CDS construction algorithms, the maintenance of the CDS constructed based on multiple leaders may be faster, as it does not rely on a spanning tree. In this subsection, we will study the algorithm introduced by Alzoubi *et al.* [6]. In next subsection, the algorithm provided by Cheng *et al.* [25] will be elaborated.

The algorithm proposed by Alzoubi *et al.* [6] constructs a CDS in a UDG with size at most $192 \cdot |OPT| + 48$. The message complexity is $O(n)$. This algorithm does not use a rooted spanning tree. Initially all the nodes are candidates. Whenever the ID of a node becomes the smallest among all of its one-hop neighbors, it will change its status to dominator. Consequently, its candidate neighbors become dominatees. After all nodes change status, each dominator node identifies a path of at most three hops to another dominator with larger ID. The candidate nodes on this path become connectors. All dominators and connectors compose a connected dominating set.

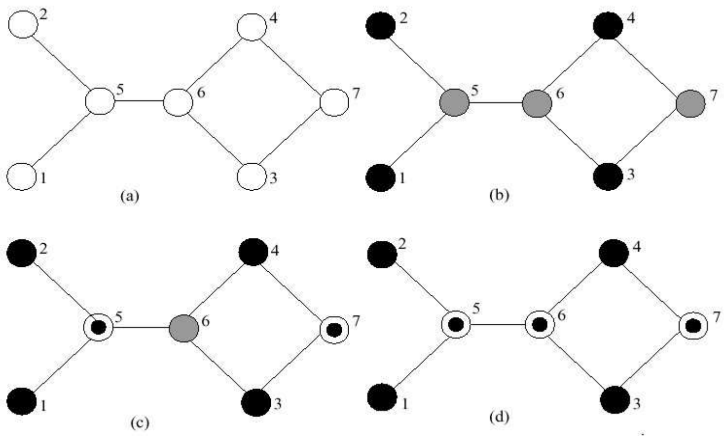

Figure 8: An example of Message-Optimal CDS Construction [6].

The example in [6] can illustrate the execution of this algorithm. As shown in Figure 8(b), nodes 1, 2, 3 and 4 declare themselves to be dominators in the beginning. Then node 5 declares itself as a connector on the paths from 1/2 to 3/4, so do nodes 6 and 7. Finally all the connectors and dominators form a connected dominating set.

4.3.4 Cheng's Multiple Leader Algorithm

Cheng's multiple leader algorithm [25] is motivated by [6]. Initially each node with smallest ID among its one-hop neighbors becomes a leader. A forest with each tree rooted at a leader is constructed first. Then a chain of one or two nodes are computed to connect neighboring trees. This algorithm has linear time and message complexities, and generates a CDS with size at most $147 \cdot |OPT| + 33$. The performance analysis explores the neighboring property of nodes in a tree, thus the algorithm achieves better performance ratio compared to the multiple leader algorithm in [6]. This algorithm achieves linear time and message complexities.

4.3.5 Single Leader vs. Multiple Leader

Although single leader based CDS construction algorithms provide better performance bounds, their non-localized construction may render them unusable. Since each algorithm has a time complexity of $O(n)$, nodes may move during CDS construction such that the result of the algorithm is not a CDS. Thus, the ultimate goal of the CDS construction is to develop truly localized algorithms that have constant time complexity. Since the worst-case time complexity of the two multiple leader based algorithms [6, 25] is $O(n)$ due to the MIS or forest construction, these algorithms are not purely localized in a strict sense.

Nonetheless, multiple leader based algorithms in [6, 25] do represent progress in message complexity - they achieve the optimal message complexity of $O(n)$. Message complexity of a CDS construction algorithm contributes to the protocol overhead. It also plays an important role in the effectiveness and efficiency analysis of the protocol. Single leader based CDS construction algorithms [4, 5, 69, 23, 16] have message complexity $O(n \log n)$. The direct distributed implementation of Guha and Khuller's algorithms have message complexities $O(n|C|)$ and $O(n|C| + m + nlog(n))$ [35, 33, 66]. These algorithms have better performance ratio. Therefore as a trade-off the performance ratios of [6] and [25] are much higher.

The two truly localized algorithms are presented by Wu and Li in [76], and by Adjih, Jacquet, and Viennot in [1], which will be discussed in Subsections 4.4 and 4.5. Both algorithms need two-hop neighborhood information.

Wu and Li's algorithm first creates a larger CDS by selecting more nodes than needed, then prunes the set of selected nodes to get a CDS with smaller size. The algorithm proposed by Adjih, Jacquet, and Viennot relies on a multipoint relaying set.

4.4 Pruning Based CDS Construction

There are two pruning-based CDS construction algorithms [14, 76] proposed in the context of ad hoc and sensor networks. We will study them in the following subsections.

4.4.1 Wu and Li's Algorithm

Wu *et al.*'s work [76, 31] proposes a completely localized algorithm to construct CDS in general graphs. Initially all vertices are unmarked. They exchange their open neighborhood information with their one-hop neighbors. Therefore each node knows all of its two-hop neighbors. The marking process uses the following simple rule: any vertex having two unconnected neighbors is marked as a dominator. The set of marked vertices form a connected dominating set, with a lot of redundant nodes. Two pruning principles are provided to post-process the dominating set, based on the neighborhood subset coverage. A node u can be taken out from S, the CDS, if there exists a node v with higher ID such that the closed neighbor set of u is a subset of the closed neighbor set of v. For the same reason, a node u will be deleted from S when two of its connected neighbors in S with higher IDs can cover all of u's neighbors. This pruning idea is generated to the following general rule [32]: a node u can be removed from S if there exist k connected neighbors with higher IDs in S that can cover all u's neighbors. Wu *et al.* extend their work to calculate power-aware connected dominating sets [72, 77], by considering the power property for all the nodes as a criterion for the post pruning.

This idea is also extended to directed graphs. Due to differences in transmission ranges, or the hidden terminal problem [68] in wireless networks, some links in an ad hoc network may be unidirectional. In order to apply this algorithm to a directed graph model, neighboring vertices of a certain node are classified into a dominating neighbor set and an absorbent neighbor set in terms of the directions of the connected edges [70]. Figure 10 illustrates the dominating and absorbent neighbor sets of vertex u. In this case, the objective is to find a small set that is both dominating and absorbent for a given directed graph. The original marking process is adapted

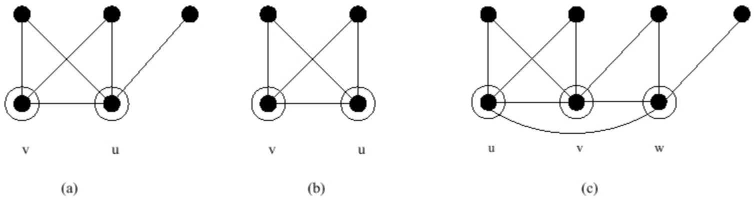

Figure 9: Examples of two pruning principles to eliminate redundant nodes [72].

as follows: a node u is added into the dominating and absorbent set, when there exists a node w in its dominating set and another node v in its absorbent set which can only be connected via u. Then similar post-process principles are used to delete the redundant nodes in the resulted dominating and absorbent set.

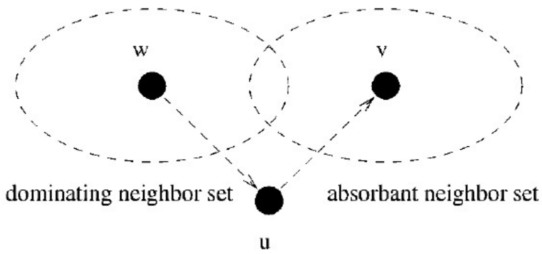

Figure 10: Dominating (absorbent) neighbor set of vertex u [70].

The performance ratio for Wu's algorithm is proved to be $O(n)$[69]. For the post-processing, the nodes need to know their two hop neighbors. Thus the algorithm has time complexity of $O(\Delta^3)$ and message complexity of $\Theta(m)$, which indicate that the maintenance of the CDS constructed by Wu's algorithm is relatively easier. In fact one major advantage of Wu and Li's algorithm is its locality of maintenance. Mobile hosts may switch off or on at any time for power efficiency. In addition to this switching, CDS maintenance may be required due to mobile hosts' movements. Compared to other CDS algorithms, Wu's algorithm only requires neighbors of a node to update their status when it changes switching status or location.

4.4.2 Butenko's algorithm

The centralized version of this algorithm [14, 15] is discussed in Subsection 3.5. Its distributed implementation has higher message complexity because global connectivity needs to be checked when examining each node. This can be done through any distributed spanning tree algorithm [9].

The algorithm starts from the node with minimum degree, which can be found by modified leader-election algorithms in [28]. Let u be the node under consideration at the current step. If removing u causes the CDS disconnected, then we color u black. u then selects its non-black neighbor with minimum effective degree (the number of non-black neighbors within the set) for consideration in next step. If it is OK to remove u, u will select a neighbor with minimum effective degree, if u does not have a black neighbor, for next step. If u does have a black neighbor v, then v will choose its neighbor with minimum effective degree for next step. This procedure will be continued until all nodes have been examined.

4.5 Multipoint Relaying Based CDS Construction

Multipoint relaying is a technique to allow each node u select a minimum forwarding set [61, 51, 17] from $N(u)$ to cover $N(N(u))$. Finding a *multipoint relay set (MRS)* with minimum size is NP-Complete [61]. Refs. [61] and [51] independently design different $log\, n$ factor greedy heuristics where n is the total number of hosts. Ref. [17] provides a sophisticated approximation algorithm with constant performance ratio 6. Multipoint relaying is mainly used for flooding control to decrease protocol overhead [46]. Recently Adjih, Jacquet, and Viennot [1] propose a localized heuristic to generate a CDS based on multipoint relaying. Their idea is sketched below.

Each node first compute a multipoint relay set, a subset of one-hop neighbors that can cover all the two-hop neighbors. The a CDS is constructed by two rules. The first rule puts all nodes with smallest ID among their neighbors into the CDS; The second rule puts a node into the CDS if the node is a member of the MRS of its smallest ID neighbor. Wu [71] enhanced the first rule by selecting the node that has at least two disconnected neighbors and has the smallest ID among all its neighbors. Chen and Shen [22] claims that by considering node degree instead of ID, the constructed CDS may have smaller size. The correctness of this heuristic is proved in [1] but no performance analysis is available.

4.6 Steiner Tree Based CDS Construction

The second algorithm proposed in Guha and Khuller [44] is Steiner tree based. Its distributed implementation [35, 33, 34] applies a spanning tree to approximate the computation of Steiner points. The single leader based algorithms proposed in [2, 3, 4, 5, 16, 23, 69] also compute Steiner points to connect the MIS. These algorithms have been discussed in Subsections 4.2 and 4.3, thus they will not be repeated here. In this subsection, we will study how the centralized algorithm proposed in [58] is implemented in a distributed fashion.

As mentioned by Subsection 3.4, Min's CDS construction algorithm applies Steiner tree with minimum number of Steiner points to connect an MIS with the following property: any subset of the MIS has hop distance two with its compliment. We need to consider two phases in the distributed implementation: the construction of the MIS and the computation of the Steiner points.

Cheng *et al.* [25] and Wan *et al.* [69] both imply heuristics to compute an MIS in a distributed fashion. Here we rephrase the simple idea based on [25]. Assume all nodes are white initially. In the first step, color the node with smallest id black. Color all its one-hop neighbors gray; color its two-hop neighbors yellow. At each step, choose a yellow node with smallest id among all its yellow neighbors and color it black. Color its one-hop white/yellow neighbors gray; color its two-hop white neighbors yellow. Repeat this procedure until all nodes are colored either black or gray. All black nodes form an MIS.

The distributed computation of the Steiner tree with minimum number of Steiner points is non-trivial, as demonstrated by [58]. The idea is stated below: each gray node u (a node not in the MIS) keeps track of the neighboring connected black components. Its *competitor* set includes all its gray neighbors and all the gray nodes adjacent to its neighboring connected black components. u becomes a Steiner point if and only if it is adjacent to the maximum number of black components among all its competitors. It is clear that this distributed implementation has very high message complexity.

4.7 Proactively Considering Nodal Mobility in CDS Construction

Most of the previous algorithms have attempted to address nodal movements by incorporating a maintenance phase in which the CDS can be reconstructed when nodes move. However, in the case of high nodal mobility, this reactive approach may not yield a stable infrastructure. A number of DS algorithms have been developed that attempt to proactively utilize mobility information in an attempt to create stable clusters.

The Mobility Adaptive Clustering Algorithm, for example, selects slow moving nodes for inclusion in a WCDS [11]. When nodes move randomly, a fast moving clusterhead is likely to encounter another clusterhead sooner than a slow moving one. Furthermore, the open neighbor sets of fast moving nodes will exhibit more change than those of slow moving nodes. Therefore, this algorithm selects slow moving nodes, which are more likely to have stable links.

Mobility-Based Clustering also considers nodal movement in the creation of a WCDS [8]. This WCDS approach has three notable features:

- It uses relative mobility information to create clusters.
- It allows cluster members to be L hops from the clusterhead, where $L \geq 1$, in order to increase the stability of clusters.
- Rather than seeking a minimal WCDS, it allows for clusterheads to come into contact with each other if the clusterheads are going in different directions.

In this WCDS protocol, neighbors periodically exchange their position and velocity information with their neighbors. A node calculates the relative velocity and the relative mobility between itself and all of its neighbors. Relative mobility is an average of the magnitude of the relative velocity vector in the recent past. Once relative mobility has been measured, there is an initial cluster creation where, a node elects as its clusterhead its neighbor with the lowest id, whose relative mobility falls below a user-defined threshold. As this step is performed, a clusterhead which has been elected, may join another cluster, with the two clusters then merging into one, subject to the L-hop rule. Cluster maintenance is done sparingly, only when a node loses contact with its clusterhead, and encounters a node whose relative mobility is less than the threshold.

(α,t)-Clusters takes a similar approach in the creation of a WCDS. It requires that nodes in a cluster be mutually reachable after time t with a probability of α [56]. In their simulations, nodes move in a random walk-based mobility. They derive the probability that two nodes are mutually reachable based on this mobility pattern, and current mobility information.

The Clustering for Open Inter-vehicle communication Networks (COIN) algorithm arises because clustering protocols designed random mobility patterns perform poorly for inter-vehicle communication [12]. This algorithm was tested with nodal mobility generated from microscopic vehicular traffic simulators. Vehicular mobility is highly constrained by the layout of road network, by traffic control devices, and by surrounding vehicles. Vehicle movement is characterized by high rates of speed, producing very high relative velocities. Driver behavior has a significant impact of mobility patterns both in the near term and in the long term. In the near term, vehicle movements vary dramatically based on individual lane changing, braking, and passing behaviors. In the long term, mobility is affected by the variations in the intended destination of a driver. The COIN algorithm enhances the prediction of future mobility by incorporating driver intentions into the prediction algorithm. The destination of a driver could be gleaned either from an on-board route guidance system, or through a statistical analysis of previous trips on the current roadway. The paper also identifies a source of instability in clustering - oscillatory inter-vehicle distances between vehicles with low relative mobility. Examples of this phenomenon can be found in stop and go traffic, as vehicles pass through four-way stop signs, or as vehicles slow to navigate curves in the roadway. In order to accommodate this phenomenon the algorithm relaxes the condition that no two clusterheads with low relative mobility be within radio contact. Instead, a design parameter, the inter-cluster minimum distance, specifies the minimum distance between two clusterheads.

5 Analysis of Distributed CDS Algorithms

The algorithms presented in the previous section represent an evolution of distributed CDS algorithms designed for use in a variety of wireless applications. This evolution has been driven by both improvements in efficiency and stability and by the design requirements of the consumer applications. The following analysis presents the evolution at these two levels. First, we describe the development of a toolbox of techniques for CDS construction. Then, at a higher level, we elicit the manner in which the needs of the consumer applications dictate the selection of specific techniques from this toolbox.

Since non-localized algorithms rely on global information, their time complexity is at least $O(n)$ as in [33]. Purely localized CDS construction algorithms operate much quicker, with a complexity related to the maximal degree, typically $O(\Delta^3)$ [1, 76]. This quicker execution time comes at a cost of a larger CDS.

5.1 A Toolbox for CDS Construction

5.1.1 WCDS Construction Techniques

WCDS construction plays an important role not only in clustering algorithms, but also in a number of CDS construction algorithms, which begin with the selection of a WCDS. The first WCDS construction algorithm selected the lowest ID neighbor as clusterhead [40]. In newer algorithms, techniques have been developed to reduce the size of the clusterhead set, improve the stability of the clusterhead set, and promote fairness in clusterhead selection.

The size of the WCDS can be reduced by choosing nodes with the highest degree as clusterheads [43]. Although this will reduce the size of the WCDS, if a network with mobile nodes, this may increase WCDS instability since nodal degree varies while ID does not.

Since stability of the WCDS is a key factor in many applications, clusterhead stability has been improved through preservation of clusterhead elections and multi-hop clusters. Clusterhead selection need not change whenever the topology of the network changes; rather, reelections might only occur when clusterheads move into range of each other or when a node moves out of range of all clusterheads [27]. Multi-hop clusters can improve cluster stability by increasing the area covered by a single clusterhead [7].

Fairness may also be a problem with Lowest ID clustering, since the burdens of being a clusterhead are primarily borne by those nodes with lower ID's. However, mitigating this shortcoming should not be done in a way that decreases stability, as in highest connectivity clustering without clusterhead election preservation. For example, Min-Max D-Clusters attempted to address this issue by selecting both low and high id nodes as clusterheads [7].

5.1.2 CDS Construction Techniques

We have discussed example algorithms exploiting the following distributed CDS construction techniques: greedy, MIS based, Steiner tree based, pruning based, and multipoint relaying. We briefly discuss them in the following.

Das *et al.*'s greedy algorithms [35, 33, 66] are the distributed implementations of Guha and Khuller's algorithms in [44]. These heuristics have high message complexity due to the global selection of the greedy choice. They are the first distributed ones proposed for MCDS computation in the context of wireless ad hoc networks. One feature of their scheme is to store the global topology information only in dominating nodes. This reduces access and update overheads for routing. However, their CDS construction requires two hop neighborhood knowledge. The generated CDS has high approximation ratio and high implementation complexities (in message and time). In addition, it is not clear in their algorithm description how each individual node is informed on when to start the second stage. The CDS maintenance is expensive too, as their approaches need to maintain a spanning tree.

The MIS based algorithms [6, 25, 23, 69] compute and connect an MIS. Their performance analysis in unit-disk graphs take the advantage of the relationship between an MIS and OPT. Algorithms in this category usually have good performance bound and time/message complexities. They only need one-hop neighborhood information. However, the single leader based algorithms [13, 16, 25, 23] require leader election. This drawback makes them difficult to support localized CDS maintenance. Multiple leader based algorithms [6, 25] are optimal in message complexity. Compared to single leader algorithms, they are relatively more practical in local maintenance since they obviates the rooted spanning tree construction.

Pruning based algorithms [14, 76] prune a large CDS. Wu and Li's algorithm [76] is the first purely localized CDS construction heuristic. Butenko *et al.*'s algorithm [14] has high message complexity due to the global connectivity checking.

Wu and Li's algorithm [70, 76, 72] is very simple. The localized property makes the CDS maintenance easier. However, there is no performance analysis in the original paper [76, 72], which incorrectly analyzes the algorithm's time complexity. Ref. [4] corrects the mistake in [76], and proves that Wu and Li's algorithm has a linear performance ratio. This algorithm needs at least two-hop neighborhood information. It is presented based on the general graph model, and is extended to directed graph [70]. This is important for wireless network, as either the disparity of transmission range or the hidden terminal problem in physical wireless networks can cause unidirectional links.

The Steiner tree based CDS construction heuristic in [58] uses a Steiner tree with minimum number of Steiner points to connect a dominating set, usually an MIS. Computing Steiner points requires large number of message exchange.

The multipoint relaying based heuristic [1] is pure localized. This algorithm selects CDS from a multipoint relay set. No complexity analysis for this algorithm in literature.

We summarize the above analysis for major algorithms in the following table.

	Graph modal	Approx. ratio	Time complexity	Msg. complexity	Ngh. info.	Maintenance
[33]-I	general	$2H(\Delta)+1$	$O((n+\|C\|)\Delta)$	$O(n\|C\|+m+n\log(n))$	2-hop	non-local
[33]-II	general	$2H(\Delta)$	$O(\|C\|(\Delta+\|C\|))$	$O(n\|C\|)$	2-hop	non-local
[4]-I	UDG	$8opt+1$	$O(n)$	$O(n\log(n))$	1-hop	non-local
[69]	UDG	$8opt$	$O(n)$	$O(n\log(n))$	1-hop	non-local
[6]	UDG	$192opt+48$	$O(n)$	$O(n)$	1-hop	non-local
[16]	UDG	$8opt$	$O(n)$	$O(n\log(n))$	1-hop	non-local
[25]	UDG	$147opt+33$	$O(n)$	$O(n)$	1-hop	non-local
[76]	general	$O(n)$	$O(\Delta^3)$	$\Theta(m)$	2-hop	local
[58]	UDG	$6.8opt$	-	-	2-hop	non-local
[1]	general	-	$O(\Delta^3)$	$\Theta(m)$	2-hop	local

Table 1: Performance comparison for distributed CDS construction algorithms in [33, 4, 69, 6, 16, 25, 76, 58, 1]. Here n and m are the number of vertices and edges respectively; *opt* is the size of any optimal MCDS for the given instance; Δ is the maximum degree; $|C|$ is the size of the computed CDS.

5.2 Selection of CDS Construction Methods

The design goals of the application dictate the selection from the different techniques for creating a virtual network infrastructure. Depending on the needs of the application, either a WCDS or CDS will be appropriate. The application requirements will also dictate the balance between bounded performance and fast operation. Finally, requirements for stability will affect the size and characteristics of the dominating set, as well as the node selection methodology.

The selection of a WCDS or a CDS depends on the communication requirements between nodes within the dominating set. Routing applications tend to rely on CDS, since messages must be forwarded along a backbone. However, if the network topology is highly unstable, a WCDS may be a better choice since its smaller size makes it easier to maintain. Likewise, while intra-cluster coordination functions might be managed within a WCDS, inter-cluster coordination is probably more easily handled by a CDS. So if adjacent clusters are using orthogonal codes or different frequency bands, a WCDS could manage media access within each cluster. However, for a system-wide media access coordination, a CDS may be more efficient since it includes nodes needed for clusterhead communication.

In addition to selecting the underlying virtual network infrastructure, the application developer must select the appropriate balance between performance bounds and fast operation. As seen in the previous examples, significantly better performance bounds are achievable if the dominating set is constructed serially as opposed to in parallel. However, the application may not be able to accommodate the additional time required to construct the dominating set. Moreover, with mobile nodes, then by the time the dominating set is constructed, the network topology may have changed such that the result of the algorithm is not a dominating set.

In addition to performance considerations, the stability requirements of the dominating set are also crucial for applications that require long-lasting dominating sets for mobile nodes. The stability of dominating set can be measured either from the perspective of a dominating set or from the perspective of a node outside of the dominating set. One may seek to reduce the rate of CDS change. Alternatively, one may seek to increase the duration that a node is associated with a given node in the dominating set. An example of the latter would be to increase the time that a node is in the cluster of a given clusterhead.

Depending on the definition of stability, different techniques can be employed. If one seeks to promote the stability of the CDS, then the selection of low mobility nodes as members of the CDS is suitable approach.

However, if one also seeks to promote the association stability between nodes and members of the CDS, then one must create these associations based on the predictions of relative mobility between nodes. Furthermore, these predictions are domain-specific - predictions that reflect random mobility are likely to be unsuitable for domains where mobility is constrained.

A common technique in many of these algorithms is enlarging the size of the dominating set. This achieves greater stability by relaxing the constraints on the number of elements in a dominating set that are within radio range of each other. For the vehicular mobility case, for example, one ends up with at least two dominating sets - one for each direction of vehicle traffic. The overlap in these sets allows for both stability of the CDS, and stability of node association with clusterheads.

6 Conclusions and Discussions

Dominating sets have proven to be an effective construct within which to solve a variety of problems that arise in wireless networks. Applications that use dominating sets include media access coordination, unicast and multicast routing, and energy efficiency.

The needs of these consumer applications drive the design goals of the dominating set construction algorithms. In our review of these algorithms, we have examined a number of these considerations. The most obvious is the choice of target set - a WCDS, CDS, or a d-hop dominating set. Furthermore, the large differences in time and message complexity are the result of tradeoffs between fast operation and bandwidth efficiency versus dominating set size. In addition, these algorithms may need to address nodal mobility, if the dominating set is not transitory. Some algorithms have attempted to address this problem through the creation of localized maintenance routines that seek to avoid CDS recreation through a rerunning of the construction algorithm. Others have also included proactive consideration of nodal mobility in order to promote CDS stability.

As ad hoc wireless networking moves from the research labs to the real world, additional requirements will certainly arise. Issues that will likely arise include the security and survivability of CDS-based applications. In infrastructure-based wireless networks, the infrastructure plays a key role for security purposes. A perimeter is established around the infrastructure, regulating access to the network. A CDS could perform similar actions through forming a virtual network infrastructure. However, significant issues must first be addressed including the real-time assurance of the trustworthiness of mobile hosts. Similarly, performance of the DS algorithms in the face of Denial of Service and other security attacks must be evaluated in order to ensure robust, survivable network functionality.

References

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

[2] K. M. Alzoubi, P.-J. Wan, O. Frieder, Distributed Construction of Connected Dominating Set in Wireless Ad Hoc Networks, to appear in *ACM Mobile Networks and Applications*.

[3] K. M. Alzoubi, P.-J. Wan, O. Frieder: Maximal Independent Set, Weakly Connected Dominating Set, and Induced Spanners for Mobile Ad Hoc Networks, *International Journal of Foundations of Computer Science*, Vol. 14, No. 2, pp. 287-303, 2003.

[4] K.M. Alzoubi, P.-J. Wan and O. Frieder, New Distributed Algorithm for Connected Dominating Set in Wireleess Ad Hoc Networks, *Proceedings of the 35th Hawaii International Conference on System Sciences*, Big Island, Hawaii, 2002.

[5] 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*, Vol. 4, No. 1, Mar. 2002.

[6] K.M. Alzoubi, P.-J. Wan and O. Frieder, Message-Optimal Connected Dominating Sets in Mobile Ad Hoc Networks, *MOBIHOC*, EPFL Lausanne, Switzerland, 2002.

[7] A. D. Amis, R. Prakash, T. H. P. Vuong, and D. T. Huynh, Max-Min D-Cluster Formation in Wireless Ad Hoc Networks, Proc. IEEE INFOCOM, Tel Aviv, Israel, 2000.

[8] B. An and S. Papavassiliou, A mobility-based clustering approach to support mobility management and multicast routing in mobile ad-hoc wireless networks, *International Journal of Network Management*, 11, pp. 387-395, 2001.

[9] H. Attiya and J. Welch, Distributed computing: fundamentals, simulations and advanced topics, McGraw-Hill Publishing Company, 1998.

[10] B. Awerbuch, Optimal Distributed Algorithm for Minimum Weight Spanning Tree, Counting, Leader Election and Related Problems, *Proceedings of the 19th ACM Symposium on Theory of Computing, ACM*, pp. 230-240, 1987.

[11] S. Basagni, Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks, Proc. IEEE 50th Vehicular Technology Conference, Piscataway, NJ, 1999.

[12] J. Blum, A. Eskandarian, and L. Hoffman, Mobility Management of IVC Networks, Proc. IEEE Intelligent Vehicles Symposium, Columbus, OH, 2003.

[13] S. Butenko, X. Cheng, D.-Z. Du, and P.M. Pardalos, On the construction of virtual backbone for ad hoc wireless networks, In S. Butenko, R. Murphey, and P.M. Pardalos, editors, *Cooperative Control: Models, Applications and Algorithms*, pp. 43-54, Kluwer Academic Publishers, 2003.

[14] S. Butenko, X. Cheng, C. Oliveira, and P.M. Pardalos, A new heuristic for the minimum connected dominating set problem on ad hoc wireless networks, In S. Butenko, R. Murphey, and P.M. Pardalos, editors, *Recent Developments in Cooperative Control and Optimization*, pp. 61-73, Kluwer Academic Publishers, 2004.

[15] S. Butenko, C. Oliveira, and P.M. Pardalos, A new algorithm for the minimum connected dominating set problem on ad hoc wireless networks, *Proc. of CCCT'03*, Vol. V, pp.39-44, International Institute of Informatics and Systematics (IIIS), 2003.

[16] M. Cadei, X. Cheng and D.-Z. Du, Connected Domination in Ad Hoc Wireless Networks, *Proc. 6th International Conference on Computer Science and Informatics*, 2002.

[17] G. Calinescu, I. Mandoiu, P.-J. Wan and A. Zelikovsky, Selecting forwarding neighbors in wireless ad hoc networks, *manuscript*, 2001.

[18] Y.-L. Chang and C.-C. Hsu, Routing in wireless/mobile ad-hoc networks via dynamic group construction, *Mobile Networks and Applications*, 5, pp. 27-37, 2000.

[19] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, Span: An energy-efficient coordination algorithm for topology maintenance in Ad Hoc wireless networks, *Wireless Networks*, 8, pp. 481-494, 2002.

[20] D. Chen, D.-Z. Du, X. Hu, G.-H. Lin, L. Wang, and G. Xue, Approximations for Steiner trees with minimum number of Steiner points, *Journal of Global Optimization*, Vol. 18, pp. 17-33, 2000.

[21] Y. Chen, A. Liestman, Approximating minimum size weakly-connected dominating sets for clustering mobile ad hoc networks, *ACM MobiHoc*, June 2002.

[22] X. Chen and J. Shen, Reducing connected dominating set size with multipoint relays in ad hoc wireless networks *Proceedings of the 7th International Symposium on Parallel Architectures, Algorithms and Networks*, pp. 539 - 543, May 2004.

[23] X. Cheng, Routing Issues in Ad Hoc Wireless Networks, *PhD Thesis*, Department of Computer Science, University of Minnesota, 2002.

[24] X. Cheng, M. Ding, and D. Chen, An approximation algorithm for connected dominating set in ad hoc networks, *Proc. of International Workshop on Theoretical Aspects of Wireless Ad Hoc, Sensor, and Peer-to-Peer Networks (TAWN)*, 2004.

[25] X. Cheng, M. Ding, D.H. Du, and X. Jia, On The Construction of Connected Dominating Set in Ad Hoc Wireless Networks, to appear in Special Issue on *Ad Hoc Networks* of *Wireless Communications and Mobile Computing*, 2004.

[26] X. Cheng, X. Huang, D. Li, W. Wu, and D.-Z.Du, Polynomial-Time Approximation Scheme for Minimum Connected Dominating Set in Ad Hoc Wireless Networks, *Networks*, Vol. 42, No. 4, pp. 202-208, 2003.

[27] C.-C. Chiang and M. Gerla, Routing and Multicast in Multihop, Mobile Wireless Networks, Proc. IEEE ICUPC'97, San Diego, CA, 1997.

[28] I. Cidon and O. Mokryn, Propagation and Leader Election in Multihop Broadcast Environment, *Proc. 12th Int. Symp. Distr. Computing*, pp. 104-119, Greece, Spt. 1998.

[29] B. N. Clark, C. J. Colbourn and D. S. Johnson, Unit disk graphs, *Discrete Mathematics*, Vol. 86, 1990, pp. 165-177.

[30] T. Clausen, P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, and L. Viennot, Optimized link state routing protocol, IETF Internet Draft, draft-ietf-manet-olsr-05.txt, October 2001.

[31] F. Dai and J. Wu, An Extended Localized Algorithm for Connected Dominating Set Formation in Ad Hoc Wireless Networks, to appear in *IEEE Transactions on Parallel and Distributed Systems*, 2004.

[32] F. Dai and J. Wu, Distributed dominate pruning in ad hoc wireless networks, *Proc. ICC*, 2003.

[33] Bevan Das and Vaduvur Bharghavan, Routing in Ad-Hoc Networks Using Minimum Connected Dominating Sets, *International Conference on Communications*, Montreal, Canada, June 1997.

[34] Bevan Das, Raghupathy Sivakumar and Vaduvur Bharghavan, Routing in Ad Hoc Networks Using a virtual backbone, *IC3N'97*, pp.1-20, 1997.

[35] Bevan Das, Raghupathy Sivakumar and Vaduvur Bharghavan, Routing in Ad Hoc Networks Using a Spine, *International Conference on Computers and Communications Networks*, Las Vegas, NV. Sept. 1997.

[36] S. Datta and I. Stojmenovic, Internal Node and Shortcut Based Routing with Guaranteed Delivery in Wireless Networks, *Cluster Computing*, 5(2), pp. 169-178, 2002.

[37] Budhaditya Deb, Sudeept Bhatnagar, and Badri Nath, Multi-resolution State Retrieval in Sensor Networks, *First IEEE Workshop on Sensor Network Protocols And Applications (SNPA)* , Anchorage 2003

[38] M. Ding, X. Cheng, and G. Xue, Aggregation tree construction in sensor networks, *Proc. of IEEE VTC*, 2003.

[39] D.-Z. Du, L. Wang, and B. Xu, The Euclidean bottleneck Steiner tree and Steiner tree with minimum number of Steiner points, *COCOON'01*, pp. 509-518, 2001.

[40] A. Ephremides, J. Wieselthier, and D. Baker, A design concept for reliable mobile radio networks with frequency hopping signaling, *Proceedings of the IEEE*, 75, pp. 56-73, 1987.

[41] M. R. Garey and D. S. Johnson, Computers and Intractability: A guide to the theory of NP-completeness, Freeman, San Frncisco, 1978.

[42] R. De Gaudenzi, T. Garde, F. Giannetti, and M. Luise, DS-CDMA techniques for mobile and personal satellite communications: An overview *IEEE Second Symposium on Communications and Vehicular Technology in the Benelux*, pp.113 - 127, 2-3 Nov. 1994.

[43] M. Gerla and J. T. Tsai, Multicluster, mobile, multimedia, radio network, *ACM/Baltzer Journal on Wireless Networks*, 1, pp. 255-265, 1995.

[44] S. Guha and S. Khuller, Approximation algorithms for connected dominating sets, *Algorithmica*, 20(4), pp. 374-387, Apr. 1998.

[45] A. Helmy, Efficient resource discovery in wireless adhoc networks: contacts do help, Manuscript, 2004.

[46] P. Jacquet, P. Muhlethaler, A. Qayyum, A. Laouiti, L. Viennot and T. Clausen, Optimized link state routing protocol, IETF Internet Draft, draft-ietf-manet-olsr-04.txt, March 2001.

[47] D.B. Johnson and D.A. Maltz, Dynamic source routing in ad hoc wireless networks, *Proc. Mobile Compuing* edited by T. Imielinski and H. Korth, Kluwer Academic Publisher, 1996.

[48] U. Kozat, L. Tassiulas, Service discovery in mobile ad hoc networks: an overall perspective on architectureal choices and network layer support issues, *Ad Hoc Networks Journal*, In press, 2003.

[49] B. Liang and Z.J. Haas, Virtual backbone generation and maintenance in ad hoc network mobility management, *IEEE INFOCOM* 2000, pp.1293-1302.

[50] J. Li, J. Jannotti, D. S. J. D. Couto, D. R. Karger, and R. Morris, A scalable location service for geographic ad hoc routing, Proc. of the sixth annual international conference on Mobile computing and networking, Boston, MA, 2000.

[51] H. Lim and C. Kim, Multicast Tree Construction and Flooding in Wireless Ad Hoc Networks, Proc. 3rd ACM international workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Boston, MA, pp. 61-68, 2000.

[52] H. Lim and C. Kim, Flooding in wireless ad hoc networks, *Computer Communications Journal*, Vol. 24 (3-4), pp. 353-363, 2001.

[53] G.-H. Lin and G.L. Xue, Steiner tree problem with minimum number of Steiner points and bounded edge-length, *Information Processing Letters*, Vol. 69, pp. 53-57, 1999.

[54] C. Lund and M. Yannakakis, On the hardness of approximating minimization problems, *Journal of the ACM*, Vol. 41(5), pp. 960-981, 1994.

[55] M. V. Marathe *et al.*, Simple heuristics for unit disk graphs, *Networks*, vol. 25, pp. 59-68, 1995.

[56] A. B. McDonald and T. F. Znati, Mobility-based framework for adaptive clustering in wireless ad hoc networks, *IEEE Journal on Selected Areas in Communications*, 17, pp. 1466-1487, 1999.

[57] M. Min and D.-Z. Du, A reliable virtual backbone scheme in mobile ad-hoc networks, *manuscript*, 2004.

[58] M. Min, C.X. Huang, S. C.-H. Huang, W. Wu, H. Du, and X. Jia, Improving construction of connected dominating set with Steiner tree in wireless sensor networks, submitted.

[59] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen and J.-P. Sheu, The broadcast storm problem in a mobile ad hoc network, *Proc. MOBICOM*, Seattle, Aug. 1999, pp. 151-162.

[60] C. Perkins and E. Royer, Ad-hoc on-demand distance vector routing, *Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90-100, 1999.

[61] A. Qayyum, L. Viennot, and A. Laouiti, Multipoint relaying for flooding broadcast messages in mobile wireless networks, *HICSS02*, pp. 3866 - 3875, January 2002.

[62] L. Ruan, D.H. Du, X. Jia, W. Wu, Y. Li, and K.-I Ko A greedy approximation for minimum connected dominating sets, *manuscripts*, 2004.

[63] J. Shaikh, J. Solano, I. Stojmenovic, and J. Wu, New Metrics for Dominating Set Based Energy Efficient Activity Scheduling in Ad Hoc Networks, *Proc. of WLN Workshop (in conjunction to IEEE Conference on Local Computer Networks)*, pp. 726-735, Oct. 2003.

[64] R. Sivakumar, P. Sinha and V. Bharghavan, CEDAR: a core-extraction distributed ad hoc routing algorithm, *Selected Areas in Communications, IEEE Journal on*, Vol. 17(8), Aug. 1999, pp. 1454 -1465.

[65] P. Sinha, R. Sivakumar and V. Bharghavan, Enhancing ad hoc routing with dynamic virtual infrastructures, *INFOCOM 2001*, Vol. 3, pp. 1763-1772.

[66] R. Sivakumar, B. Das, and V. Bharghavan, An Improved Spine-based Infrastructure for Routing in Ad Hoc Networks, *IEEE Symposium on Computers and Communications*, Athens, Greece, June 1998.

[67] I. Stojmenovic, M. Seddigh, J. Zunic, Dominating Sets and Neighbor Elimination Based on Broadcasting Algorithms in wireless networks, *Proc. IEEE Hawaii Int. Conf. on System Sciences*, 1988.

[68] A. Tanenbaum, *Computer Networks*, Prentice Hall, 1996.

[69] P.-J. Wan, K.M. Alzoubi and O. Frieder, Distributed Construction of Connected Dominating Sets in Wireless Ad Hoc Networks, *IEEE INFOCOM*, 2002.

[70] J. Wu, Extended Dominating-Set-Based Rounting in Ad Hoc Wireless Networks with Undirectional Links, *IEEE Trans. on Parallel and Distributed Systems*, pp. 866-881, Sept. 2002.

[71] J. Wu, An enhanced approach to determine a small forward node set based on multipoint relays *IEEE 58th Vehicular Technology Conference*, Vol. 4, pp. 2774 - 2777, October 2003.

[72] J. Wu, F. Dai, M. Gao, and I. Stojmenovic, On Calculating Power-Aware Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks, *Journal of Communications and Networks*, Vol. 5, No. 2, pp. 169-178, March 2002.

[73] J. Wu and F. Dai, Broadcasting in ad hoc networks based on self-pruning, *IEEE INFOCOM*, 2003.

[74] J. Wu and F. Dai, On Locality of Dominating Set in Ad Hoc Networks with Switch-On/Off Operations, *Journal of Interconnection Networks*, a special issue on *Algorithms for Networks*, Vol. 3, No. 3 & 4, pp. 129-147, Sept. & Dec. 2002.

[75] J. Wu and F. Dai, On Locality of Dominating Set in Ad Hoc Networks with Switch On/Off Operations, *Proc. of the 6th International Conference on Parallel Architectures, Algorithms, and Networks (I-SPAN02)*, 2002.

[76] J. Wu and H. Li, On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks, *Proc. of the Third International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pp. 7-14 , Aug. 1999.

[77] J. Wu, M. Gao, and I. Stojmenovic, On Calculating Power-Aware Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Net-

works, *Proc. of International Conference on Parallel Processing (ICPP)*, 346-356, 2001.

[78] J. Wu and B. Wu, A Transmission Range Reduction Scheme for Power-Aware Broadcasting in Ad Hoc Networks Using Connected Dominating Sets, *Proc. of 2003 IEEE Semiannual Vehicular Technology Conference (VTC2003-fall)*, Oct. 2003.

[79] J. Wu, B. Wu, and I. Stojmenovic, Power-Aware Broadcasting and Activity Scheduling in Ad Hoc Wireless Networks Using Connected Dominating Sets, *Proc. of IASTED International Conference on Wireless and Optical Communication (WOC 2002)*, 2002.

[80] J. Wu, B. Wu, and I. Stojmenovic, Power-Aware Broadcasting and Activity Scheduling in Ad Hoc Wireless Networks Using Connected Dominating Sets, *Wireless Communications and Mobile Computing*, a special issue on *Research in Ad Hoc Networking, Smart Sensing, and Pervasive Computing*, Vol. 3, No. 4, pp. 425-438, June 2003.

[81] W. Wu, H. Du, X. Jia, M. Min, C.H. Huang, and X. Cheng, Maximal independent set and minimum connected dominating set in unit disk graphs, *manuscripts*, 2004.

[82] Y. Xu, J. Heidemann, and D. Estrin, Geography-informed energy conservation for Ad Hoc routing, MobiCom 2001, pp.70-84, 2001.