Chapter 2

A Survey on Graph Domination Based Algorithms for Routing in Wireless Ad-Hoc Network

2. Introduction

Mobile Ad-hoc networks have been widely researched for many years research on wireless ad hoc networks has been ongoing for decades. The idea for wireless ad hoc networks was triggered during the project called Deferece Advanced Research Project Agency (DARPA) packet radio networks (PRNet). Which evolved into the survivable adaptive radio networks. Ad hoc networks have play an important role in military and other related areas [1]. Recent years have witnessed other areas of applications such as industrial and commercial communication equipment. Ad hoc networks are a new technology of wireless communication for mobile nodes. In ad hoc networks, there is no fixed infrastructure such as a base station; mobile nodes that are within each other’s radio range can communicate directly via wireless links. While these are for part of rely on other nodes to relay messages as routers. Node mobility in an ad hoc network causes frequent changes of the network topology. Therefore, to achieve high survivability, ad hoc networks should have a distributed architecture within no central entities. Introducing any centralized entity is compromised then the entire network will destroy, due to frequent changes in both its topology and its nodes (i.e. new nodes will join and existing nodes will cut off), communication among nodes also changes.
The advantages [2] of ad hoc networks include the following

- Independence from central networks administration
- Self configuring nodes
- Nodes are also act as routers
- Continuous re-configuration
- Scalability

Some limitations of ad hoc networks

- All the nodes should perform well
- The load of the networks will affect the throughput
- The reliability of the networks depends on availability of nodes
- When network comprises large number of nodes it affects latency.

If we compare these limitations with wired and centralized networks wireless ad-hoc networks are inherently different from the wired networks because it is new architecture. Some challenges are: Self-organization and transporting the information through wireless. The nodes in the wireless networks moving randomly at any time. So the networks topology changes quickly this affect severely on routing because nodes cannot be assumed to have persistent data storage, bandwidth is big challenge in ad hoc networks wireless links have significantly lower capacity than their wired
networks, also due to some other interference conditions etc. The wireless links have low throughput.

Energy constrained operation: All nodes in a MANET may rely on batteries the most important criteria for optimization may be energy conservation, limited physical security: Mobile networks are generally more prone to physical security threats than are fixed cable networks.

As mobile ad hoc networks are characterized by a multi hop network. Topology can change often due to node movement reliable routing protocols are required to establish link between nodes. Consisting less traffic overhead or computing time on energy constrained nodes. A huge number of techniques have been proposed, many of them being subject to standardization within the LETF. The proposed techniques should have always updates route to remaining nodes all the time. These protocols

Fig 1 Multihop Ad Hoc Networks
periodically exchange control information when topology changes. Such type of protocols is called proactive routing protocols. Most of the time it is not necessary to have an up-to-date route to all other nodes. Therefore reactive routing protocols have been proposed keeping the routes alive when they needed. Combining both the routing technique fall in category of hybrid routing protocols. A completely different approach has been proposed called location based routing protocol. Where packet forwarding is based on the location of a node’s communication partner. Location information services provide nodes with the location of the others.

2.1 Classifying Routing Techniques

2.1.1 Multicast Techniques for Mobile Ad Hoc Networks

To adopt multicast routing technique on mobile ad hoc networks. The main challenge is to efficiently controlling the dynamic topology due to node movement. To solve this problem many technique have been proposed based on their operations. There exist different taxonomy schemes to classify these ad hoc multicast routing protocols including connectivity among group members.

- Tree-based vs mesh based
- Proactive vs reactive
- Sender-initiated vs receiver- initiated
- Source-based vs group-shared
In this section we classify ad hoc multicast protocols based on the methodologies used to maintain connectivity among group members.

2.1.1.1 Tree-Based Protocols

Tree based multicast protocols are reliable for bandwidth constraints hence tree based multicast protocols are most preferred. The bandwidth constraints are more in ad hoc network due to its dynamic nature of its topology. Some tree based protocols have been designed to give stability of nodes. Mobility and failure adjust according to its requirement, for example “adaptive shared tree multicast “.

2.1.1.2 Mesh –Based Protocols

Tree based protocols are very sensitive due to dynamic nature of its topology, to retain the path all the time many ad hoc multicast protocols [3] based on a mesh structure have been proposed. Unlike a tree data packets are allowed to be forwarded to the same destination through more than one path, which increases the perfection of delivery.

2.1.2 Hierarchical Hybrid and Adaptive Protocols

This section covers ad hoc multicast protocols that view the networks as hierarchical [3] rather than flat. Adaptive protocol means they are not designed for only one scheme (ie Tree or Mesh) they adapt all behaviors of the network, for example MZR (Multicast Zone Routing Protocol) adopt hybrid approach using the same technique given by Zone Routing Protocol.
Another approach is to reduce the number of nodes when flooding on entire networks is required, this approach based on the notion of connected dominating sets (CDS). This technique is exploited by CGM (Clustered Group Multicast), and MCEDAR (Multicast Core-Extraction Distributed Ad hoc Routing).

2.1.3 Other Classifications for Multicast Protocols.

The classification described in the previous section is based on how multicast connectivity is setup (i.e. a tree vs a mesh). In fact there are other many different aspects we can consider for protocol classification. *Source-Based vs Group-Shared connectivity*

*Proactive vs Reactive*

The protocol continuously maintaining connectivity among all the members of the network whether the data traffic is available or not. Compare to proactive protocol reactive protocols are maintaining the route on demand. In other words when receiving or sending the data.

*Sender - Initiated vs Receiver –Initiated*

To establish multicast link among nodes it should be initiated by sources, receivers or both. When sender initiated each source is responsible for announcing its existence to other nodes in the network. A receiver - initiated protocol in the context requires each receiver initiated a request to the group by searching for a point of attachment to the current tree/ mesh.
**Unicast- Dependent vs Unicast – Independent**

Many unicast protocols have been design to operate independently, There are many disadvantages of using unicast protocol, for example waste of band width designing multicast protocol to be independent of any unicast protocol allows the protocol to have complete knowledge about and better utilizes the control overhead

A multicast Tree [3] or Mesh may be created and used to forward the data packets generated by a particular source (Source Based Connectivity) or by any source within the group (Group Shared Connectivity)

### 2.2 Network Model of Wireless Ad Hoc Networks

An ad hoc wireless network is established without the aid of any infrastructure or centralized network [30]. A wireless ad hoc network can be represented as a simple graph $G (V, E)$, where $V$ represents a set of mobile nodes and $E$ represents a set of edges. An edge $(u, v)$ in $E$ indicates that nodes $u$ and $v$ are neighbors, and that $u$ is within $v$’s range of transmission, while $v$ is within $u$’s range [31] although wireless ad hoc networks are modeled using a special class of graphs, unit disk graphs. Given an undirected graph $G = (V, E)$ the distance between two nodes, $u$ and $v$, denoted by $d(u, v)$, is the number of edges that together form the shortest path between those nodes. The diameter of $G$ is defined by $\text{diam}(G) = \max\{d(u, v) : u, v \in V \}$. For any non-empty subset $C$ of $V$, $G(C)$ denotes the subgraph of $G$ induced by $C$. A tree, denoted by $T$, is an acyclic connected graph. The tree $T(u)$ is the subtree of $T$ rooted at $u$. The height of $T$ denoted by $h(T)$, is the length of the longest path from the root
to a leaf. For a leaf node $x$, of a tree $T$, we have $h(T(x)) = diam(T(x)) = 0$. In [32] Johnson, Clark and Colbourn describe three canonical definitions for modeling a unit disk graph, Unit Disk Graph (UDG): is a graph $UDG (V, E)$ with set of nodes $V$ and a link $(i, j)$ between nodes $i$ and $j$ belongs to link set $E$ if and only if their Euclidean distance is less than 1. UDG can be directed in which link is also directed. Most of the ad-hoc network modeling uses UDG. A more generalized UDG is one, in which the link between two nodes is possible if their Euclidean distance is less than $r$, where $r$ is the radius of the circular transmission range of the antenna of the mobile node. This model is well suited for ad-hoc network, as it is almost realistic to ad-hoc network, where the geodesic of the transmission range of the radio signals coverage is almost circular. One of the issue related to this models is to find out the minimum threshold range of the transmitters of the nodes so that network is connected. One of which is called the intersection model. In the intersection model a set of $n$ unit disks in the plane is represented by an $n$ node graph, where every node corresponds to a unit disk and there is an edge between two nodes if the corresponding unit disks intersect or tangent. Our system model includes two general assumptions regarding the state of the network’s communication links and topology. The first assumption is that the network topology remains unchanged throughout the execution of the k-clustering algorithm. The second assumption is that the network may be modeled using a unit disk graph, even though communication between network nodes is, in actuality, a dynamic function involving a number of factors such as interference and signal propagation conditions. An ad hoc network can be viewed as a unit disk graph by
viewing every transmitter/receiver in the broadcast network as a point in the graph and by representing the effective broadcast range of each point as a unit disk.

### 2.2.1 Challenges of Routing Protocols

Many unique characteristics of self-organized network have posed new challenges on routing protocol design: dynamic network topology, energy constraints, lack of network scalability and a centralized entity, and the different characteristics between wireless links and wired links such as limited bandwidth, unidirectional links, and poor security. Address migration, locality migration and other critical properties related to computing mobility are also key challenges. Volatility of the network i.e.; host mobility causes network topological changes, multihop communication and limited resources (lower bandwidth, low battery power, limited CPU) are some properties that pose credible challenges to mobile networks. All these induce more failures in mobile networks. Several routing protocols have been proposed to address these problems. Routing problem is to find a route for sending a packet from a source to a given destination. There are two main classes of routing protocols [33, 34].

1) **Topology based**

2) **Position based**

Topology based routing protocols are based on information about the links. Position based routing protocols use additional information about physical positions. Shortest path algorithm does not work well in MANETS because some nodes may become temporarily inactive or nodes might move. Wireless networks require localized algorithms; traditional routing protocols that use link state or distance vector in wired networks are not suitable in ad hoc wireless networks [35] greedy routing. Face
algorithm, combination of Greedy Face Greedy (GFG) algorithm are other position based algorithms. Dynamic source protocol (DSR) has also been proposed [36] DSR allows the network to be completely self-organizing and self-configuring. The protocol includes two parts: Route Discovery and Route Maintenance. Some researches proposed a new approach where a sub-graph of the ad hoc wireless network is selected and then the sub-graph is searched for routing. This reduces the running time. Dominating set based routing is one such kind of sub graph routing. Wu and Li proposed an efficient algorithm to calculate connected dominating set [37].

2.2.2 Connected Dominating Set for Wireless Ad Hoc Network

Wireless ad hoc networks have unique features that need the all protocols specific to them compared to wired and centralized networks due to its dynamic topology, that motivate the development of protocols specific to them. For stability, reliability and efficiency reasons, at the time of designing these protocols researchers thought organizing the ad hoc network using dominating set specially with connected dominating set because connected dominating set is the one of the best approach but unfortunately finding the minimum connected dominating set is NP complete. The protocols of ad hoc networks designed to address the media access, routing, power management, and topology control. At the Link layer of ad hoc network, clustering the wireless nodes with dominating set can increase spatial reuse of the spectrum, minimize collisions, and provide Quality of Service (QoS) [38,39,40,41] correspondingly, the nodes in the dominating set can coordinate with one another and use orthogonal spreading codes in their neighbor hoods to improve
spatial reuse with code division spread spectrum techniques [42] still, there are many more advantages of having clustered nodes these nodes can coordinate access to the wireless media by their neighbors for \((Q_oS)\) or collision avoidance purposes. A CDS can create a virtual backbone for routing and control [43]. 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 [44, 45] or Backbone based routing [46] or spine based routing [47, 48]. Constructing the network using connected dominating set results in a significant reduction in message complexity associated with routing updates [49]. Furthermore, the dominating set can be organized into a hierarchy to further reduce control message overhead. 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 [50]. We can also improve the multicast/broadcast routing with the help of connected dominating set. If the message is routed along a CDS, most of the redundant broadcasts can be eliminated [51]. We know that nodes in a wireless network the main constraint is energy supply is on batteries. To save energy to
maximum extent connected dominating set play an important role in energy management. This is the one of the main application of the connected dominating sets is to balance the network management requirements to conserve energy among nodes [52]. In high-scale dense wireless nodes, network topology information extraction can be handled by CDS construction [53,54]. 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 [55] or to serve as database servers etc.

2.2.3 Minimum Connected Dominating Set

The algorithmic complexity of MCDS is known to be \(NP\)-hard for unit disk graphs according to [56]. Therefore, several approximation algorithms have been proposed for MCDS. For example, in [57], Breu et al. present some simple heuristics for a number of classical NP-hard optimization problems on unit disk graphs, including the MCDS problem. The heuristics do not require a geometric representation of a unit disk graph as part of the input. Geometric representations are used only for establishing certain properties of unit disk graphs. These properties, in turn, are used in order to derive the performance guarantees provided by the heuristics. The sequential approximation algorithm for MCDS presented in has a constant worst case ratio of 10. Given a unit disk graph \(G = (V, E)\), the algorithm first constructs a BFS spanning tree \(T\), rooted at a given arbitrary node \(r\). The tree \(T\) partitions the nodes of \(G\) into disjoint sets \(S_i, 0 \leq i \leq h(T)\), where \(S_i\) is the set of nodes in the tree at height \(i\). The connected dominating set produced by the algorithm is the union of the
following two sets of nodes. The first set is a maximal independent set for $G$, obtained by appropriately selecting an independent set $IS_i$ from each graph $G[S_i], 1 \leq i \leq h(T)$. More precisely, after choosing $IS_i - 1$, all dominated nodes in $S_i$ are removed, and $IS_i$ is chosen from the remaining nodes. In this way, $i = 0 IS_i$ forms a maximal independent set $IS$. The second set of nodes is used to ensure connectivity of $IS$, by connecting members of $IS_i$ with members of $S_i - 1, 1 \leq i \leq \ell$. Note that in this way, each edge is incident to one member of $IS$. Implementation of MCDS, provided in [58] which has a time complexity of $O(n)$ and a message complexity of $O(n \log n)$.

![Figure 2: Simple unconnected wireless network](image)

There are four nodes in the graph shown in Figure 2. Nodes $A$ and $B$ are within a certain transmission range, say 100 units. Nodes $B$ and $C$ are also within the same transmission range. But node $A$ and node $C$ are too far away that their distance is greater than the transmission range. Node $D$ is too far away from all other nodes in
the network hence there is no connection to node \(D\) from any other node in the network. If the wireless network needs to relay packets from node \(A\) to node \(C\), node \(B\) should be used as an intermediate node to forward packets between them. Other alternative paths could be used if there are other nodes between node \(A\) and node \(C\). Most mobile nodes usually function as end nodes and connecting gateways simultaneously.

### 2.2.4 Centralized CDS Algorithms and Decentralized CDS Algorithms

The dominating set problem first studied in the year 1850's, before the invention of wireless networks [59]."The objective of the seven queens’ problem is to find the minimum number of queens that can be placed on a chess board 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 [60] in which a reduction from the Set Cover Problem [61] "to the MCDS problem was shown. This result implies that for any fixed \(0 < \varepsilon < 1\), no polynomial time algorithm with performance ratio \(\leq (1-\varepsilon)H(\Delta)\) exists unless \(NP \subseteq DTIME[n^{O(\log \log n)}]\) [62]where \(\Delta\) is the maximum degree of the input graph and \(H\) is the harmonic function. The MCDS remains \(NP\)-hard [63]. MCDS in unit-disk graphs has constant performance ratio, as proved by [64, 65, 66, 67]. A
polynomial time approximation scheme for computing a MCDS in unit-disk graphs has been developed by Cheng et al. in [68]. 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. Connected dominating algorithms can be categorized into two types:

i) Centralized

ii) Decentralized

Centralized: The centralized algorithm gives minimum connected dominating set (MCDS) and good performance ratio.

Decentralized: We can further divide the decentralized algorithm into two categories: distributed algorithm and localized algorithms. In distributed algorithm, the decision process is decentralized. In the localized algorithm, the decision process is distributed. Following we present a literature survey of published graph domination based routing algorithms for wireless ad hoc networks. The task of constructing a stable and efficient routing algorithm for ad hoc network represents a greater challenge compared to routing in networks based on a fixed and wired infrastructure. Routing based on dominating set i.e. connected dominating set is a right approach, where the searching technique for a route is reduced to node in the dominating set. We survey different dominating set based routing algorithm for (MANETs) highlighting their objective, features, and complexity.
2.3. Graph Domination Based Routing for Wireless Ad Hoc Networks

2.3.1 A Dominating Set Based Routing Scheme

In [4] Wu, Li focused some work on the formulation of a dominating set from an undirected graph and on routing scheme within the induced graph from the connected dominating set. Once the construction of the dominating set is over in the wireless ad hoc network the next step is to decide the routing process in the network, here authors focused the routing exclusively on dominating set based. Therefore they have defined entire routing process is based on dominating set, pointed in three steps:

1. If the source is not a gateway host, forwards the packets to a source gateway, which is one of the adjacent gateway hosts.
2. This source gateway acts as a new source to route the packets in the induced graph generated from the connected dominating set.
3. Eventually, the packets reach a destination gateway, which is either the destination host itself or a gateway of the destination host.

Actually as pointed out in the introduction chapter we have two methods in routing process within the induced graph:

**Proactive Routing:** A protocol is considered proactive if it continuously maintains multicast connectivity among group members, regardless of the availability of data traffic. This scheme is advantageous is that the multicast connectivity is readily available for data transfer. However, it may often end up using a large portion of valuable network bandwidth to keep the connectivity updated, especially when the
network topology is dynamic. Therefore, a proactive scheme is usually suitable for networks with low mobility.

**Reactive routing:** Reactive protocols attempt to establish connectivity among members on demand, i.e., only when a source has data to send. Therefore, no bandwidth is wasted even though the network topology keeps changing, given that nobody has data to send. A drawback of a reactive scheme is the longer multicast route acquisition time and frequent use of network-wide flooding.

Authors used the destination-sequenced distance vector routing protocol (DSDV) [69] applying the proactive routing to illustrate the technique. Routing within the induced graph is not limited to proactive routing, which is usually based on routing tables; reactive routing can also be applied. DSDV is based on the distributed Bellman–Ford (DBF) routing mechanism to construct routing tables. According to DBF, a node knows the length of the shortest path from each neighbor node to every network destination and this information is used to compute the shortest path and next node in the path to each destination. An update message contains a vector of one or more entries, each of which specifies as a minimum, the distance to a given destination. A major performance problem with DBF is that it takes a very long time to update the routing tables of network nodes after network partitions, node failures, or increases in network congestion. This performance problem of DBF stems from the fact that it has no inherent mechanism to determine when a network node should stop incrementing its distance to a given destination. For example, given an ad hoc wireless network as shown in Figure 3 (a) [4], the corresponding routing information at host 8 is shown in
Figure 3 (b), which shows that host 8 has three members—3, 10, and 11—in its gateway domain member list. Figure 3 (c) shows the gateway routing table at host 8, which consists of a set of entries for each gateway and its member list. The third column of this table shows the next-hop information of a shortest path (here defined as a path with a minimum hop count) and the fourth column the distance (in hop count) to each destination.

![Diagram of network topology and routing table](image)

<table>
<thead>
<tr>
<th>Destination</th>
<th>member list</th>
<th>next hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>9 (1, 2, 3, 11)</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>4 (5, 6)</td>
<td>7</td>
</tr>
<tr>
<td>11</td>
<td>7 (6)</td>
<td>7</td>
</tr>
</tbody>
</table>

Figure 3. A routing example.
CDS Construction

Author’s objective is to design a CDS on decentralized form. Some essential features for such method are listed below:

- Construction of the CDS should be distributed and simple. Ideally, it requires only local information and a constant number of iterative rounds of message exchanges among neighboring hosts.
- The outcome of the dominating set should be connected and close to minimum.
- The resultant dominating set should include all intermediate nodes of any shortest path. In this case, an all-pair shortest paths algorithm only needs to be applied to the sub network generated from the dominating set.

Marking process

A marking process marks every vertex in a constant number of rounds in a given connected and simple graph $G = (V, E)$ that represent an ad-hoc wireless network, $m(v)$ is a marker for vertex $u$ to $v$, which is either $T$ (marked) or $F$ (unmarked). The set of vertices that are marked $T$ forms a connected dominating set. Assume that $N(u)$ represents the neighbor set of vertex $u$ and $v$ has $N(v)$ initially. The marking process consists of three steps
1) Initially assign marker $F$ to every $v$ in $V$

2) Every $v$ exchanges its neighbor set $N(v)$ with all its neighbors

3) Every $v$ assigns its marker $m(v)$ to $T$ if there exist two unconnected neighbors

In the example of Figure 4, $N(u) = \{v, y\}, N(v) = \{u, w, y\}, N(w) = \{v, x\}, N(y) = \{u, v\}$, and $N(x) = \{w\}$. After Step 2 of the marking process, vertex $u$ has $N(v)$ and $N(y)$, $B$ has $N(u)$, $N(w)$, and $N(y)$, $w$ has $N(v)$ and $N(x)$, $y$ has $N(u)$ and $N(v)$, and $x$ has $N(w)$. Based on Step 3, only vertices $v$ and $w$ are marked $T$. Clearly, each vertex knows distance-2 neighborhood information after Step 2 of the marking process, i.e., neighbor information of its neighbors. The cost of checking the connectivity of two neighbors is upper bounded by $\Delta^2(G)$ or simply $\Delta^2$, where $\Delta$ is the degree of graph $G$, i.e., $\Delta(G) = \max\{|N(v)| \mid v \in V\}$. There are $|N(v)|(|N(v)| - 1)/2$ possible pairs of neighbors of vertex $v$, which is
upper bounded by $\Delta^2$. Therefore, the cost of the marking process at each vertex is $O(\Delta^2)$. The amount of message exchange at each vertex is also $O(\Delta)$, which corresponds to the number of neighbors.

The Wu-Li characterized algorithm as follows: For each node $Z$ the following question is asked Does $Z$ have neighbors $x$ and $y$ such that $x$ and $y$ are not adjacent? The vertex $Z$ is then admitted to $G$ set which we will call $WuLi_o(G)$ if and only if the answer to the question is “yes”. It is then possible to show that $WuLi_o(G)$is a connected dominating set, unless $G$ is complete Wu-Li then consider refining the above technique by assuming that each vertex has a unique integer identifier. There “Rule1” amounts to asking a further question for each vertex $Z$ in $WuLi_o(G)$as follows: Does $Z$ have a neighbor $Z'$ in $WuLi_o(G)$whose ID is higher than if $Z$ and which is such that all if the neighbors if $Z$ are also neighbors of $Z”$? If so, $Z$ is deemed to be Superfluous. The set $WuLi_1(G)$consider of all the vertices from $WuLi_o(G)$for which the answer to the question is “no”. It too connected dominating set.

To further reduce the size of the set, Wu and Li also introduce “Rule2”. For each vertex $Z$ in $WuLi_1(G)$. The following question is asked: Does $Z$ have two neighbors from $WuLi_1(G)$. Which are themselves adjacent, and which have ID’s larger than that of $Z$ and which are such that their combined neighbors include all if the neighbors of $z$? The set $WuLi_2(G)$consists if all $y = $ the vertices from $WuLi_1(G)$for which the answer is “no”. This too can be a connected dominating set.
**Example: [4]**

Figure 5 [4] shows an example of using the proposed marking process and its extensions to identify a set of connected dominating nodes. Each node keeps a list of its neighbors and sends the list to all its neighbors. By doing so, each node has distance-2 neighborhood information, i.e., information about its neighbors and the neighbors of all its neighbors. Node 1 does not mark itself as a gateway node because its only neighbors, 2 and 3, are connected. Node 3 marks itself as a gateway node because there is no connection between neighbors 1 and 4 (2 and 4). After node 3 marks itself, it sends its status to its neighbors 1, 2, and 4.
Figure: 5 (a) Marked gateways without applying rules. (b) Marked gateways by applying Rules.
This gateway status is used to apply Rule 2 to unmark some gateway nodes to non
gateway nodes. Figure: 5 (a) shows the gateway nodes (nodes with double cycles)
derived by the marking process without applying two rules. After applying Rule 1,
node 17 is unmarked to the non gateway status. The closed neighbor set of node 17 is
\(N[17] = \{17, 18, 19, 20\}\), and the closed neighbor set of node 18 is \(N[18] = \{16, 17, 18, 19, 20\}\). Apparently, \(N[17] \subseteq N[18]\). Also the ID of node 17 is
less than the ID of node 18, thus node 17 can unmark itself by applying Rule 1.After
applying Rule 2, node 8 is unmarked to non gateway status, as shown in Figure :5 (b).

Node 8 knows that its two neighbors 14 and 16 are all marked. This invokes node 8
to apply Rule 2 to check whether condition \(N(8) \subseteq N(14) \cup N(16)\) holds or not.

The neighbor set of node 14 is \(N(14) = \{7, 8, 9, 10, 11, 12, 13, 16\}\), the neighbor
set of node 8 is \(N(8) = \{12, 13, 14, 15, 16\}\), the neighbor set of node 16 is \(N(16) = \{8, 14, 15, 18\}\), and therefore, \(14 \cup N(16) = \{7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18\}\). Apparently, \(N(8) \cup N(14) \cup N(16)\).The ID of node 8 is the smallest
among nodes 8, 14, and 16. Thus node 8 can unmark itself by applying Rule 2.

Wu-Li proposed algorithm calculates connected dominating set in \(O(\Delta^2)\) time
with distance-2 neighborhood information where \(\Delta\) is the maximum node degree in
the graph. In addition, the proposed algorithm uses constant (1 or 2) rounds of
message exchange, compared with \(O(\gamma)\) rounds of message exchanges in Das
algorithm.
2.3.2 Approximation Algorithm for Connected Dominating Set

In [6] Guha and Khuller originally addressed the connected dominating set problem. Authors proposed two approximation algorithms. Two algorithms are based on growing a tree. They defined the MCDS algorithm for general graphs which proposed a reduction from the set cover problem. This implies that for any fixed $0 < \varepsilon < 1$, no polynomial time algorithm with performance $\text{ratio} \leq (1 - \varepsilon) H(\Delta)$ exists unless $\mathbf{NP} \subseteq \mathbf{DTIME}[n^{O(\log \log n)}]$ where $\Delta$ is the maximum degree and $H$ is the harmonic function.

**Algorithm I [70]** introduce an algorithm that finds a connected dominating set, by “growing a tree”. The CDS is grown from one node outward. The algorithm works as follows:

**Step 1:** marks all vertices white. Starting, the algorithm selects the node with the maximal number of white neighbors.

**Step 2:** The selected vertex is marked black and its neighbors are marked gray.

**Step 3:** 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.

**Step 4:** The selected node or the selected pair of nodes are marked black, with their white neighbors marked gray.

**Step 5:** 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 $2(1 + H(\Delta)) |OPT|$, 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 6. Initially, either node $c$, or $e$, 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 f, d, e, g forms a CDS.

Algorithm II[70]

The second algorithm too starts by coloring all nodes white. A piece is defined to be either a connected black component, or a white node. The algorithm has two phases.

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.

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(Δ)) · |OPT|. Figure 7 shows an example of the second algorithm.
Step 1: Node g is marked as it is one of the nodes with the maximum number of white neighbors.

Step 2: Node c is marked because it can reduce the maximum number of pieces compared with any other node.

Step 3: First phase ends as there is no white node left.

Step 4: In the second phase, a Steiner Tree is constructed by adding nodes d and e to connect nodes c and g.
2.3.3 Dominating Sets and Neighbor Elimination Based Broadcasting Algorithms

In [7] [23] the context of clustering and broadcasting, Stojmenovic, Seddigh, and Zunic presented three synchronized distributed constructions of CDS. In each of the three constructions the CDS consists of two types of nodes: the cluster-heads and the border-nodes. The cluster-heads form a maximal independent set (MIS) i.e. A domination set in which any pair of nodes are non-adjacent. A node is a border-node if it is not a cluster-head and there at least two cluster-heads within its 2-hop neighborhood. The set of cluster-heads is induced by a ranking of nodes which give rise to a total ordering of all nodes. Three ranking are used: The ID only [8] [9]. An ordered pair of degree and ID [10] and an order pair of degree and location. The selection of the cluster-heads is given by a synchronized distributed algorithm which can be generalized to the following framework. Initially all nodes are colored white. In each stage of the synchronized distributed algorithm, all white nodes which have the lowest rank among all white neighbors are colored black. Then all white nodes adjacent to these blanks nodes are colored gray. Finally the ranks of the remaining white nodes are updated. The algorithm ends when all nodes are colored either black or gray. All blank nodes then from the cluster-heads. Regardless of the choice of the ranking, the algorithms have a $\Delta(n)$ approximation factor. Such inefficiency stem from the non-selective inclusion of all border-nodes. In fact if the rank is ID only.
A family of instances which would simply the approximation factor to be exactly $n[11]$. The worst possible in these instances the nodes with the largest $ID$ is located at the center of a unit-disk and all other nodes are evenly distributed in the boundary of the unit disk. After the cluster heads are selected all other nodes become border-nodes. Thus the CDS would consist of all nodes [12]. On the other hand the node at the center dominates all other nodes. If the rank is an ordered pair of degree and location of the time complexity and message complexity is $O(n^2)$ and $\Omega(n)$ respectively.
**Internal nodes and dominating Set**

The nodes in the dominating set are called internal nodes. So far routing approach based on dominating set best approach which we can minimize traffic overhead. We can cluster the wireless network the cluster head belongs to dominating for wireless ad hoc networks maintenance of its topology due to it dynamic in nature.

**Reducing the ratio of internal node**

Authors maintained the record i.e. \( \text{key} = (\text{degree}, x, y) \) where degree is the number of neighbors of a node and \( x \) and \( y \) are its two coordinates in the plane. According to Wu-Li [4] rule nodes shall compare first their degree and node with higher degree has greater chance of remaining an internal nodes. In case of ties \( x \)- coordinate is used. If \( x \)- coordinates is also same use \( y \) coordinate for final decision. Such a comparison rule will result in fewer remaining nodes in the graph.

**Improved broadcasting by neighbor elimination**

The existing techniques of broadcasting algorithm which described in [18]. Authors proposed improvement of each of these broadcasting algorithms. The improved algorithm is as follows.

![Figure 9 Node A eliminates neighbors E and F from its broadcast list](image-url)
The algorithm based on Fig 9[7].

Step 1: A node will broadcast the message. If it neighbor need the message.

Step 2: Some neighbors are eliminated for rebroadcasting

Step 3: each node that is not supposed to rebroadcasts will assign retransmitting neighbors.

Step 4: If network is clustered neighbor is the corresponding cluster head.

Step 5: Each non internal node A will assign itself to neighboring internal node B which has the largest degree.

Step 6: In case of tie use lowest id among candidate neighbors.

Step 7: Neighbors that received one of message copies that reached at node A.

Step 8: Neighbors of A eliminated from the list if it might need the message.

Example[7]: Node A in fig 9[7], which received twice the message which is being broadcast from neighbors B and C. Neighbors E and F are eliminated from the broadcast list since they received the same broadcast message from neighbors B and C respectively. However, node A will, in this example still rebroadcast the message because of neighbor G which is not covered by B or C. Nodes E, F and G are either internal nodes or non internal nodes which are assigned to A. This scheme will further reduce the number of retransmission in a broadcasting task.
2.3.4 A Polynomial Time Approximation Scheme for MCDS

In [13] cheng, Hunng Li-Wu and Du proposed the \((1 + 1/s)\) – approximation for the minimum connected dominating set in unit-disk graphs running in \(n^0 ((s \log s)^2 )\). Here the authors focused on Unit-disk graphs: Considering the application of wireless ad hoc networks they summaries the need of the virtual backbone as follows: (i) the number of nodes in the dominating set is minimized (ii) all nodes in the dominating set are connected; (iii) each of the node not in the dominating set has at least one neighbor in the dominating set clearly the idea of a minimum connected dominating set. A connected dominating set on a graph is a subset of vertices such that (a) every vertex is either in the subset or adjacent to a vertex in the subset and (b) the subgraph induced by the subset is connected. The problem of minimum connected dominating set (MCDS) is to compute a connected dominating set of minimum cardinality. Assume an ad hoc wireless network contains only homogeneous mobile hosts. Each host is supplied with an equal power omani directional antenna. We can organize the ad hoc wireless network in the form of unit-disk graph. In a unit-disk graph, the vertex set consists of finite number of points on the Euclidean plane and an edge exists between two vertices (points) if and only if the distance between them is at most one. As per the above analysis, authors formulate the problem of constructing a virtual backbone to the problem of minimum connected dominating set (MCDS) in unit-disk graphs. MCDS in general graphs has been studied by [6]. Here authors proposed a Polynomial Time Approximation Scheme (PTAS) for MCDS in unit-disk graphs. An algorithm \(A\) is a Polynomial Time Approximation Scheme (PTAS) for a minimization problem with
optimal cost $OPT$ if the following is true: Given an instance $I$ of the problem and a small positive error parameter $e$, (i) the algorithm outputs a solution which is at most $(1 + e)OPT$; (ii) when $e$ is fixed, the running time is bounded by a polynomial in the size of the instance $I$. If there exists a PTAS for an optimization problem, the problem’s instance can be approximated to any required degree.

Authors formalized the algorithm as follows

First consider the below figure 10 and 11

![Figure 10 Squares Q and Q’](image1)

![Figure 11 Central area and boundary area](image2)

For input connected unit-disk graph $G = (V, E)$ we initially find a minimal square $Q$ to contain all vertices in $V$. Without loss of generality, assume $Q = \{(x, y) \mid 0 \leq x \leq q, 0 \leq y \leq q\}$. Let $m$ be a large integer that we will determine later. Let $p = \lfloor q/m \rfloor = +1$. Consider the square $Q' = \{(x, y) \mid -m \leq x \leq mp, -m \leq y \leq mp\}$. Partition $Q'$ into $(p + 1) \times (p + 1)$ grid so that each cell is a $mxm$ square. Excluding the top and right boundaries and hence, no two cells are overlapping each other. This partition of $Q'$ is denoted by $P (((0, 0)))$ (Fig. 10). In general, the partition
$P(a,b)$ s obtained from $P(0,0))$ by shifting the bottom left corner of $Q'$ from $(-m,-m)$ to $(-m+a,-m+b)$ Figure 10: Squares $Q$ and $Q'$. For each cell $e$ as an $m \times m$ square, we denote by $Ce(d)$ the set of points in $e$ away from the boundary by distance at least $d$, for example $Ce(0)$ for the cell $e$ itself. Fix a positive integer $h$ whose value will determine later. We will call $Ce(h)$ the central area of $e$ and $Ce(0) - Ce(h+1)$ boundary area of $e$ (Fig.11). For simplicity of notation, we denote $B_e(d) = Ce(0) - Ce(d)$. Note that for each cell its boundary area and central area are overlapping with width one. For each partition $P(a,a)$, denote by $C^a(d)(B^a(d))$ the union of $Ce(d)(B_e(d))$ for $e$ over all cells in $P(a,a)$. $C^a(h)$ and $B^a(h+1)$ are called the central area and the boundary area of $P(a,a)$. For a graph $G$, denote by $Ge(d)(G\sim e(d)$ the subgraph of $G$ induced by all vertices lying in $Ce(d)(B_e(d))$ and by $G^a(d)(Ge(d))$ the subgraph of $G$ induced by all vertices lying in $Ce(d)(B_e(d))$. Let $G = (V,E)$ be an input connected unit-disk graph. Consider a subgraph $Ge(h)$. This subgraph may consist of several connected components. For each connected component. Let $K_e$ be a dominating set in $Ge(0)$ for each $Ge(0)$ for each $Ge(h)$ with each minimum cardinality such that for each connected component $H$ of $Ge(h), K_e$ contains a connected component dominating $H$. In other words $K_e$ is minimum union of connected dominating set in $Ge(0)$ for connected components of $Ge(h)$. Now we denote by $K^a$ the union of $K_e$ for $e$ over all cells in portions $P(a,a)$. 

82
2.3.5 Distributed Routing Using d-hop connected d-hop Dominating Set

In [17]. The authors generalized the Wu.Li algorithm so as to produce a $d$-hop connected $d$-dominating set that work as routers one of the important aspect of their routing scheme was that it also guaranteed shortest path routing through the network along a path that was guaranteed at any point along the way to encounter another router node within every $k$-steps. Later the authors modified this algorithm and proposed a number of variations on it. The authors said that “There is a trivial way to apply the Wu-Li algorithm” in order to produce a $d$-hop connected $d$-hop dominating set for $G$. To do so simply apply the Wu-Li algorithm to $Gd$ instead of $G$. Then from the standpoint of $G$ the resulting set is a $d$-hop connected $d$-hop dominating set. However because the graph $Gd$ obscures the sense of the distance in $G$. They felt that this is not a desirable approach. Authors concluded that working directly with the graph $G$ rather than $Gd$ and results in a set with this desirable “shortest path property”. Moreover they showed that this algorithm has a more efficient implementation.

Altering the Wu and Li algorithm

Authors replaced the Wu and Li algorithm of Rule 1" with efficient condition, and refer to the resulting algorithm as altered Wu-Li. The outcome of set of vertices will be denoted by $D(G)$. Specially, the algorithm we wish to consider proceeds as follows:
Step 1. Consider each pair of vertices $x$ and $y$ which are separated by a distance 2 in $G$.

Step 2. For such a pair, consider all of the common neighbors of $x$ and $y$. Let $E(x, y)$ denote the vertex among these common neighbors whose ID is largest.

Step 3. Admit a vertex to the set $D(G)$ if and only if it is $E(x, y)$ for some suitable pair $x$ and $y$.

We will say that $E(x, y)$ was “elected” by the pair $x$ and $y$ to join the set. Notice that vertices elected by altered Wu-Li are also in the set $WuLi0(G)$. Moreover, any vertex eliminated by Rule 1 from $WuLi1(G)$ would not be elected to $D(G)$. Thus $D(G)$ and $WuLi1(G)$. An advantage of this approach over that of Wu and Li is the “shortest path property” described in the following theorem. This is a special case $(d = 1)$ of Theorem 2.

**Theorem 1:** Assume that the connected graph $G$ has radius at least two. Then the set $D(G)$ constructed by the altered Wu-Li algorithm is a connected dominating set. Moreover, any two vertices in $G$ can be connected by a shortest path consisting solely of vertices from $D(G)$ (a part from the endpoints).

**d-hop connected and d-hop dominated**

Here authors applied the Wu-Li algorithm directly to $G_d$ not on $G$ the outcome of the set is a $d$-hop connected $d$-hop dominating set. But $G_d$ will not give the correct sense of distance in $G$. Therefore authors proposed new algorithm called $d$-hop Connected
d-hop Dominating Set algorithm \( (d\text{-}CDS) \) works directly on \( G \) and results in a set with this desirable "shortest path property". The algorithm is as follows:

**Algorithm**

1. For each pair of vertices \( x \) and \( y \) satisfying \( \delta(x,y) = d + 1 \) consider all of the shortest paths from \( x \) to \( y \).

2. Consider the set of vertices that lie strictly between \( x \) and \( y \) along such a path. Let \( E(x,y) \) be the vertex in this set with the highest ID. Call this vertex \( E(x,y) \).

3. Construct the set \( D_d(G) \) by including all such \( E(x,y) \) and only these vertices.

This algorithm also has a “Shortest path property” as described in the following theorem.

**Theorem 2:** Assume that the connected graph \( G \) has radius at least \( d + 1 \). Then the set \( D_d(G) \) is a \( d \)-hop connected \( d \)-hop dominating set. Moreover any two vertices \( u \) and \( v \) from \( G \) can be connected by a shortest path (in \( G \)) with the property that the set of vertices which are on this path and also in \( D_d(G) \) together with the vertices \( u \) and \( v \), from a connected path between \( u \) and \( v \) in the \( d \)-closure \( G_d \). Overall, the generalized \( d \)-CDS algorithm performed very well compared to others in terms of the message costs and cumulative routing path lengths. The dominating set size for generalized \( d \)-CDS was little larger than that of for Wu-li with Rules 1 & 2 turned on. This is
expected since the generalized $d$-CDS may add more nodes into the set to ensure the “shortest path property”.

**Example:** consider the example in Figure 12[17]. The vertices here together with the solid edges constitute the graph $G$. By adding to this the dashed edges, the graph $G2$ is obtained. In this example, when the Wu-Li algorithm is applied to the graph $G2$ (not $G$), the set WuLi0($G2$) is found to consist of all the vertices except 2, 7, and 8. Each of these three vertices has the property that it forms a clique with its neighbors, and so does not have a pair of non-adjacent neighbors. Rule 1 eliminates vertex 5 because all of its neighbors are also neighbors of vertex 13.
Rule 2 eliminates several nodes. Vertex 1 is “covered by" vertices 6 and 13 in that the combined neighbors of 6 and 13 include all the neighbors of 1. Moreover, vertices 1, 6 and 13 are pair wise adjacent, and of course 6 and 13 are both larger than 1. Therefore Rule 2 eliminates vertex 1. Likewise vertex 3 is covered by 6 and 12, vertex 4 is covered by 9 and 10, and vertex 11 is covered by 12 and 13. As a result, $WuLi2(G2) = \{ 6, 9, 10, 12, 13, 14, 15 \}$. So $|WuLi2(G2)| = 7$.

Next consider applying altered Wu-Li to $G2$. Pairs of vertices are a distance two apart in $G2$ if and only if they are a distance three or four apart in $G$. Based on this, it is not difficult to check that $D(G2) = \{ 1, 5, 6, 9, 10, 12, 13, 15 \}$. So $|D(G2)| = 8$. Lastly, consider $2 - CDS$ algorithm applied to the graph $G$. This set consists of the nodes elected by pairs at a distance three in $G$, and one checks that $D2(G) = \{ 5, 6, 9, 10, 12, 13, 15 \}$. So $|D2(G)| = 7$. Notice that this set contains the vertex 5, while $WuLi2(G2)$ does not. Also notice that the unique shortest path from 7 to 6 does not contain an intermediary node from $WuLi2(G2)$. Thus this set does not have the “shortest-path property”. By Theorem 2, the set $D2(G)$ must contain such a node.

### 2.3.6 Extended Dominating Set Based Routing

In[18] Jie Wu extended the dominating - Set - Based routing to Ad-Hoc wireless networks with unidirectional links Wu and Li [19] conducted some preliminary work on the formation of a dominating set for an undirected graph on a preliminary routing scheme within the induced graph from the connected dominating set. Specifically an
efficient localized algorithm for determining dominating and absorbent set of the vertices is given and this set can be easily updated when the network topology changes dynamically. In [18] author proposed new algorithm called extended dominating-set-based routing to ad hoc wireless networks with unidirectional links. In an ad hoc wireless network, some links may be unidirectional due to either the disparity of transmitter ranges of hosts or the hidden terminal problem [71]. In a network with unidirectional links, the domination concept has to be redefined. Specifically, an ad hoc wireless network is represented as a directed graph, $D = (V, A)$ consisting of a finite set $V$ of vertices and a set $A$ of directed edges. A host $v$ in $V$ is called a dominating neighbor (absorbent neighbor) of another host $u$ in $V$ if there is a directed link $((v, u))$. A subset of vertices (mobile hosts) is dominating and absorbent if every vertex not in the subset has one dominating neighbor and one absorbent neighbor in the subset. A special localized algorithm called the extended marking process is proposed. This algorithm needs only two or three rounds of information exchanges to determine a connected dominating and absorbent set. Author applied the extended marking process iteratively to form a hierarchy of dominating and absorbent sets. They show that the dominating and absorbent set can be easily updated when the network topology changes dynamically. It is shown that they derived dominating and absorbent set exhibits good locality properties.

**Domination in directed graph**

A directed graph $D = (V, A)$ consists of a finite set $V$ of vertices and a set $A$ of directed edges, where $A \subseteq V \times V$. $D$ is a simple graph without self-loop or multiple
edges. A directed edge from \( u \) to \( v \) is denoted by an ordered pair \((u, v)\). If \((u, v)\) is an edge in \( D \), we say that \( u \) dominates \( v \) and \( v \) is an absorbent of \( u \). A set \( V' \subseteq V \) is a dominating set of \( D \) if every vertex \( v \in V - V' \) is dominated by at least one vertex \( u \in V' \). Also, a set \( V' \subseteq V \) is called an absorbent set if for every vertex \( u \in V - V' \), there exists a vertex \( v \in V' \) which is an absorbent of \( u \). The dominating neighbor set of vertex \( u \) is defined as \( \{w : (w, u) \in A\} \). The absorbent neighbor set of vertex \( u \) is defined as \( \{v : (w, u) \in A\} \). Fig. 13 illustrates the dominating (absorbent) neighbor set of vertex \( u \). These two neighbor sets may overlap with each other. A directed graph \( D \) is strongly connected if for any two vertices \( u \) and \( v \), there exists a \((u, v)\) -path (i.e., a path connecting \( u \) to \( v \)). If it is not strongly connected, the network management subsystem will partition the network into a set of independent sub networks, each of which is strongly connected. The objective here is to quickly find a small set that is both dominating and absorbent in a given directed graph. The absorbent subset may overlap with the dominating subset. In an undirected graph, these two concepts are the same and, hence, a dominating set is also an absorbent.

---

Figure 13
**Extended Marking Process:** To determine a set that is both dominating and absorbent, we propose an extended marking process basically, a node is marked whenever it is on the shortest path from one neighbor to another. We will show later that the marked set is both dominating and absorbent. Fig. 14a shows four gateway hosts (4, 7, 8, and 9) derived from the extended marking process. Arrow dashed lines correspond to unidirectional links and solid lines represent bidirectional links. A bidirectional link \( vu \) can be considered as two unidirectional links \((vu)\) and \((v,u)\) and\((u,v)\). Fig. 5 shows three assignments of \( u \), with one dominating neighbor \( w \) and one absorbent neighbor \( v \). The only case in Fig. 15 with \( m(u) = F \) is when \((w,v) \in A\), for every dominating neighbor \( w \) and every absorbent neighbor \( v \) of \( u \). The fourth case, where \( v \) and \( w \) are bidirectional connected (a combination of Figs.1 5a and1 5b), is not shown. Assume that \( V' \) is the set of vertices that are marked \( T \) in \( V \), that is, \( V' = \{ u: u \in V, m(u) = T \} \). The induced graph \( D' \) is the subgraph of \( D \) induced by \( V' \) (i.e., \( D' = D[V'] \)).

1. Initially assign \( F \) to each \( u \in V \)

2. \( u \) changes its marker \( m(u) \) to \( T \) if there exist vertices \( v \) and \( w \) such that \( (w,u) \in A \) and \( (u,v) \in A \) but \( (w,v) \notin A \)
Non-gateway mode
Level -1 gateway mode
Level -2 gateway mode

Gateway dominating member list

(b)

Gateway routing table

(c)

Figure 14
Extensions: Author proposed two rules to reduce the size of connected dominating and absorbent set generated from the extended marking process. They first randomly assign a distinct label \(id(u)\) to each vertex \(u\) in \(V\). \(Nd(n(Na(u)))\) represents the dominating (absorbent) neighbors sets; that is \(N(u) = Na(u) \cup Nd(u)\). Vertex \(u\) is called neighbor of vertex \(V\) if \(u\) is a dominating absorbent or dominating and absorbent neighbor of \(v\). Again \(V'\) is the marked set after applying the extended marking process and \(D'\) is the representing the topology of the ad hoc wireless network.

2.3.7 On Calculating Power-Aware Connected Dominating set

In [20] authors extended the Wu and Li’s distributed algorithm for calculating CDS. The connected dominating set in given ad hoc network. Obviously the CDS is selected based on node degree and energy level. Authors provided the scheme that...
balanced the energy consumption in the network par alley small connected dominating set is generated. While the energy level-based approach tries to prolong the average life span of each node.

**Node-Degree-Based Rules:** Authors proposed two rules based on node degree(ND)[20] to reduce the size of $G$ connected dominating set generated from the marking process. First of all, a distinct $ID$, $id(\nu)$ is assigned to each vertex $\nu$ in $G$. In addition $nd(\nu)$ represent the node of $u$ in $G$. i.e the cardinality of $u$'s open neighbor set $N(u)$.

**Rule 1:** The rule indicates that when the closed neighbor set of $\nu$ is covered by that of $u$, node $\nu$ can be removed from $G'$ if the ND to $\nu$ is smaller that of $u$. Node $ID$'s are used to break a tie when the node degrees of two nodes are the same. Note that $nd(\nu) < nd(u)$ implies that $N[u] \not\subseteq N[\nu]$, and if $\nu$ is marked and it is closed neighbor set is covered by that of $u$, it implies that node $u$ is also marked. It is easy to prove that $G' - \{\nu\}$ is still a connected dominating set of $G$. The condition $N[\nu] \subseteq N[u]$, implies $\nu$ and $u$ are connected in $G'$.

**Rule 2:** The rule1 indicates that when the open neighbor set of $\nu$ is covered by the open neighbor sets of two of its marked neighbors $u$ and $w$

1. If neither $u$ nor $w$ is covered by the other two among $u, \nu$ and $w$ node $\nu$ can be removed from $G'$.

2. If nodes $\nu u$ are covered by $u$ and $w$. $\nu$ and $w$ respectively but $w$ is not covered by $u$ and $\nu$ node $u$ can be removed from $G'$ if the ND of $\nu$ is small
than that of \( u \) or the ID of \( v \) is smaller than that of \( U \). When their ND's are the same

3. When each of \( u, v \) and \( W \) is covered by the other two among \( u, v \) and \( w \) node \( u \) can be removed from \( G' \) if one of the following conditions hold: \( U \) has the minimum NW among \( u, v \) and \( w \) the ND of \( v \) is the same as the ND of \( u \) but it is smaller than that of \( W \) and the ID of \( v \) is smaller than that of \( u \) or the ND's of \( u, v \) and \( w \) are the same and \( v \) has the minimum LD among \( u, v \) and \( w \). The condition \( N(u) \subseteq N(v) \cup N(w) \) in Rule 2 implies that \( u \) and \( w \) are connected. Again it is easy to prove that \( G' - \{u\} \) is still a connected dominating set. Both \( u \) and \( w \) are marked because the fact that \( u \) is marked and \( N(v) \subseteq N(u) \cup N(w) \) in \( G \) does not imply that \( u \) and \( w \) are marked.

Therefore if one of \( u \) and \( w \) is not marked \( u \) cannot be unmarked

**Energy -Level:** Author proposed two rules based on energy level (EL) to prolong average life span of a host and at the same time to reduce the size of a connected dominating set generated from the marking process. We first assign a distinct ID, \( Id(V) \) and an initial EL, Let e1 (u) to each vertex \( v \) in \( G' \)

![Figure 16](image-url)
Since $N[v](N[u])$, node $v$ is removed from $G'$ if $e_1(v) < e_1(u)$ and node $u$ is the only dominating node in the graph. Therefore $N[v] = N[u]$. Either $v$ or $u$ can be removed as shown in the above figure. We pick that with a small $EL$. In a dynamic system such as ad hoc wireless network topology changes overtime. Therefore the connected dominating set also needs to change assume that $d'$ and $d$ are energy consumption in a given interval for a gateway host and a non-gateway host, respectively. That is each time applying both Rule 1 and Rule 2 $EL$ of each gateway host will be decreased by $d$. When the energy level of $u$, $d(v)$, reaches zero it is assumed that host $u$ ceases to function. In general, $d' > d$ and $d'$ and $d$ are variables dependent on the length of update interval and by pass traffic given an initial energy level of each host and values for $d'$ and $d$, the energy level associated with each host has multiple discrete levels.

2.3.8 Routing Using MCDS

In [21] Das and Bharghavan proposed the concept of virtual backbone for unicast, multicast/broadcast in ad hoc wireless network. The virtual backbone is mainly used to collect topology information for route direction. It also works as a backup when route is unavailable temporarily. Das and Bharghavan provide the distributed implementation of the two centralized algorithm given by Guha and Khuller[6], both implementations suffer from high message complexity. The author use the connected dominating set on a graph to do shortest path based routing. The domination set induces a virtual backbone of connected vertices in the graph. Since it is 1-hop connected and 1-hop dominating. The centralized version of the distributed
algorithm proposed by Das and Bharghavan consists of three stages. The first stage
finds an approximation to minimum dominating set which is essentially the well
studied set covered problem [5] Let \( u \) denotes the Dominating set output in this stage.
The second stage constructs a spanning Forrest \( F \). each tree component in \( F \) is a union
of stars centered at the nodes in \( u \). The star \( s \) are generated by letting each dominate
node pickup an arbitrarily neighbor in \( u \). The third stage expands the spanning forest
\( F \) to a spanning tree \( T \). all internal nodes in \( T \) form a CDS. The distributed
implementation of the above greedy algorithm has very high time complexity and
message complexity, indeed, both time complexity and massage complexity can be as
high as \( \Delta(n^2) \).

**Minimum Connected Dominating Set Approach**

Authors designed an algorithm it finds as well as updates and allow to acts as a
MCDS function as a virtual backbone or Spine and finds and updates shortest path
router. MCDS provide supporting node when MCDS edge or node Spoils and MCDS
can be used for multicast and broadcast routing. The algorithm works as follows.

*Step 1: Compute the MCDS \( C \).*

*Step 2: Gather topology information from non- MCDS nodes to MCDS. Each node \( u \)
not in \( C \) transmit its adjacency list ( in one message) to \( \text{dom}(u) \)*

*Step 3: Broadcast topology to all MCDS nodes. Let \( \nu \in C \). After receiving adjacency
list from all nodes in \( \nu \)'s \( \nu \) broadcasts this information to the rest of \( C \)
Step 4: Determine routes. Each node \( v \) in \( \mathcal{C} \) runs an all pairs shortest paths algorithm on its local copy of \( G \).

Step 5: Propagate information out to non MCDS nodes. Each node \( v \) in \( \mathcal{C} \) transmits a next-hop routing task to each node \( u \) in \( v \)'s domain.

Step 6: Send periodic maintenance update. Every \( T \) seconds for some large value of \( T \), the MCDS nodes repeat the topology broadcast step. This periodic broadcast ensure that the MCDS nodes recognize drastic topology changes.

*Example:* [21] The minimum connected dominating set comprise \{3, 5, 6\}. Node dominates 1,2, 4 and 5. And node 6 dominates 7, 8, and 9. The edges plus the MCDS edges. After step 2 and 3 each MCDS node can construct its local copy of \( G \) from these local copies in step 4, 5, 6 compute the shortest path. Finally in step 5, 3 and 6 transmit routing task to the non-MCDS nodes.
**Approximation Approach**

Das and Bharghavan proposed three polynomial time approximation algorithms two for general graphs and one for bounded degree graphs. The algorithm 1 and 2 are distributed in nature which have performance ratio of $2H(\Delta) + 1$ and $2H(\Delta)$ respectively. The third algorithm has performance ratio of $\mathcal{C} + 1$

**Algorithm 1:** The algorithm has two stages, it works as follows

**Stage 1:**

*Step 1: Rounds of adding to S*

i) A node $u$ in $S$ is marked; otherwise $u$ remain unmarked

ii) The number of unmarked neighbors of a node is its effective degree $\delta*(u)$.

iii) In each round, $u$ gathers $\delta*(v)$ for all $v$ in $N_2(v)$.

iv) Node $u$ is added to $S$ if $\delta*(u) > \delta*(v)$ for all $v \neq u$ in $N_2(u)$.

*Step 2: We use minimum node ID to break tie*

**Stage 2:** Labels each components formed by the $(u, \text{dom}(u))$ edges with a common label and then connects these components.

*Step 1: before connecting discard edges that connects two nodes in the same component.*
**Step 2:** Weight the remaining edges to favor those edges that do not increase the size of \( C \) too much.

**Step 3:** An edge between a node in \( S \) and a node not in \( S \) adds only one node to \( C \).

**Step 4:** While an edge between two nodes not in \( S \) adds two nodes to \( C \).

**Step 5:** Use the number of end points not in \( S \) when assigning weights to edges.

**CDS Approach:**

Algorithm 2 gives minimum spanning tree and it grows one fragment in to \( C \). Extension to the fragment be either a one or two nodes respectively not in fragment.

The effective combined degree of an extension is the number of unmarked nodes adjacent to the non-fragment node in the extension. In each round algorithm adds the best extension to the current fragment.

**Bounded degree Approach**

When \( \Delta \leq C \) for some small constant \( c \), we can modify algorithms to find MCDS faster than for graphs of unbounded degree. Here each node \( v \) selects as its dominator the node \( w \) with the maximum degree in \( n(v) \). This one step selection takes one local broadcast of \( \delta(v) \) for each node \( v \) hence \( O(\Delta) = O(1) \) time and \( n \) messages.
2.3.9 Distributed Construction of Connected Dominating Set

In [23] Wan, Alzoubi, and Frieder considered the existing distributed approximation algorithm to find minimum connected dominating set. Here authors reinvestigated the their performance and come to the conclusion that none of these algorithms have constant approximation factors and said these approximation algorithms will not guarantee to generate a CDS of small size. Authors proposed distributed algorithm which outperform existing algorithm has an approximation factor of at most the time complexity is $O(n)$ and message complexity is $O(n \log n)$ and established $\Omega(n \log n)$ lower bound on message.

**Wan Alzoubi and Frider distributed algorithm**

The algorithm consist of two phases MIS and a dominating tree.

**MIS**: Any pair of nodes in an MIS are separated by at least two hops however a subset of nodes in an MIS may be three hops away from the subset of the rest nodes in these MIS, MIS construction guarantees that the distance between any pair of its complementary subsets is exactly two hops. Authors chosen rank definition. The ranking is induced by an arbitrary rooted spanning tree $T$. Which can be constructed by the distributed leader- election algorithm in [72] with $O(n)$ time complexity and $O(n \log n)$ message complexity. Given a rooted spanning tree $T$. the (tree) level of a node is the number of hops in $T$ between itself and root of $T$. The level of the root is 0. The rank of a node is then given by ordered pair of its level and its ID. Such ranking gives rise to a total ordering of the nodes in the lexicographic order. The MIS
construction guarantees that the distance between any pair of its complementary subsets is exactly two hops. Construction uses rank definition. The rank is indeed by an arbitrary rooted spanning tree $T$. Which can be constructed by the distributed leader election algorithm in [73].

Given a rooted spanning tree $T$, the (tree) level of node is the number of hops in $T$ between itself and root of a $T$. The rank such a ranking gives rise to a total ordering of the nodes in the lexicographic order. Algorithm works as follows.

**Step 1:** Mark all nodes initially with white color and will be marked with either gray or black eventually.

**Step 2:** Each node maintains a blacklist records IDs of its black neighbors.

**Step 3:** The root first marks itself black and broadcasts a Black message

**Step 4:** If Message is black then add sender ID to a black list else it color is white marks itself gray and broadcast a gray

**Step 5:** when gray message is received, if the rank of the sender is lower than its own then white node decrements $y$ by 1.

**Step 6:** If $Y = 0$ after update it marks itself black and broadcast black message.

**Step 7:** When a leaf node is marked with either a gray or black it transmits a Mark complete to parent node.

**Step 8:** Upon receiving a Mark Complete message towards itself a node decrements $n_2$ by 1

**Step 9:** If $n_2 = 0$ after the update and it is not the root a node transmit a mark complete message to its parent.
Finally when the local variable $n_2 = 0$ at the root all nodes have been marked with either gray or black and root will move on the construction of the CDS.
An Example of the MIS construction (a) – (g) and dominating tree construction (h) –(k)

Figure 18
Example [23]. In the graph the IDs of the nodes are labeled beside the nodes and node 0 is the leader elected in the first phase. The solid lines represents the edges in the rooted spanning tree $T$ and the dashed lines represents other edges in the unit disk graph. The ordering of the nodes by rank is given by $0, 4, 12, 2, 5, 8, 10, 3, 6, 9, 11, 1, 7$. The nodes $0, 5, 3$ and $7$ from the CDS

**Dominating Tree Construction:** The second stage constructs a dominating tree $T$ whose internal nodes would become a CDS. Each node maintains a local Boolean variable $Z$ which is initialized to 0 and set to 1 after the node joins the tree. Each node also maintains a local variable parent which stores the ID of its parent in $T$ and is initially empty and a children list which records the ID’s of its children in $T$ and is initially empty. The root of $T$ is gray neighbor of the root of $T$ which has the largest number of blank neighbor. To select the root for $T$, the root of $T$ also maintains a variable root and a variable degree which is initialized 0. Both MIS construction and dominating tree construction are linear time, Wan Alzoubi and Frieder algorithm overall takes $O(n \log n)$ message complexity and $O(n)$ time complexity, But the algorithm in [17] used for the construction of $T$ has $O(n \log n)$ message complexity and $O(n)$ time complexity

- Upon receiving an INVITE2 message, a black node with $x = 0$ sets $z = 1$ and parent to the ID of the sender, transmits a JOIN message towards the sender, and then broadcasts an INVITE1 message.
Upon receiving an INVITE1 message, a gray node with $z = 0$ sets $x = 1$ and parent to the ID of the sender, transmits a JOIN message towards the sender, and then broadcasts an INVITE2 message.

Upon receiving a JOIN message towards itself, a node adds the ID of the sender to children list.

In the above fig the thick lines are edges in the dominating tree. The internal nodes $12,0,5,7,2,3$ form a CDS.

### 2.3.10 Spine Routing

In[25] Shiva kumar Das and Bhargyan introduced a *self organizing* network structure called a *spine*. The spine structure functions as a virtual backbone for the ad hoc network. However, unlike a traditional backbone, its primary task is not to carry data packets, but to participate in route computation and maintenance. Authors designed and constructed a spine by using an approximation to the minimum connected dominating set, i.e., the interior nodes of the maximum leaf spanning tree. Authors proposed two spine routing algorithms: (1) *Optimal Spine Routing* (OSR), which uses full and up-to-date knowledge of the network topology, and (2) *Partial-knowledge Spine Routing* (PSR), which uses partial knowledge of the network topology. They analyze the two algorithms considering the most of the recent work on routing in ad hoc networks can be classified into one of the above two mechanisms, authors perceive a need for a more efficient algorithm that would balance the optimality of the shortest path algorithms with the low overhead of the routing on-demand algorithms. They introduce a *self organizing* network structure called a *spine*.
that helps combine optimality with low overhead. Briefly, the spine is chosen to be a small and relatively stable sub network of the ad hoc network, and is used to aggregate the network state distributively among the nodes of the spine in order to generate routes. Using this spine structure, they present a new spine-based routing infrastructure for fault-tolerant unicast and multicast routing in ad hoc networks. The key issues focused in this at the time of designing the algorithm are: (a) how to build and maintain the spine, (b) what network topology information to collect in the spine, and (c) how to compute routes once the information is aggregated in the spine nodes. Authors main goals are 1. Support efficient unicast routing by trading-off between shortest path routing and routing on-demand algorithms. 2. Support multicast routing by using the spine structure as the multicast backbone. 3. Compute alternate routes for long-lived connections, and switch routes dynamically upon failure of the primary route in order to provide fault-tolerant routing. Because spine is basically a routing infrastructure. Authors present two spine routing algorithms: (a) Optimal Spine Routing (OSR), which uses full and up-to-date knowledge of the network topology, and (b) Partial knowledge Spine Routing (PSR), which uses partial knowledge of the network topology. For the construction of the spine authors used the MCDS approximation algorithm

**Optimal Spine Routing**

In Optimal Spine Routing, the spine is used to provide optimal, up-to-date routes to sources in reply to route queries or update requests. To determine routes with the spine, we gather global knowledge of $G$ into all the spine nodes and compute shortest
paths based on local copies of $G$. OSR comprises the following steps to determine the
routes in spine.

**Step 1:** Compute the spine $C$.

**Step 2.** Gather information from non-spine nodes to spine nodes. Node $u$ not in $C$
transmits its edge adjacency list to $\text{dom}(u)$.

**Step 3.** Broadcast topology to all spine nodes. After receiving adjacency lists from all
nodes in $\nu$'s domain, $v$ broadcasts this information to the rest of $C$. Hence in OSR, the
state maintained is the global topology of $G$.

**Step 4.** Determine routes, due to the availability of global topology information, route
discovery is completely localized in OSR. Each node $v$ in $C$ runs an all-pairs shortest
paths algorithm on its local copy of $G$.

**Step 5.** Source gets routes. When a source $s$ wants a route not in its route cache, $s$
sends out a route query. The dominator $\text{dom}(s)$ replies with the shortest-path route
from $s$ to its targeted destination.

**Step 6.** Event-based maintenance updates. Each edge/node insertion/deletion is
broadcast along the spine.

**Step 7.** Periodic maintenance updates. Every seconds, for some large value of $t$ the
spine nodes repeat the topology broadcast step.

This step ensures that spine nodes recognize drastic topology changes, such as the
partitioning of $G$ [74].
We illustrate the above steps on the example in figure 19. Step (1), 3, 5 and 6 are the spine nodes, with \( 3 = \text{dom}(1) = \text{dom}(2) = \text{dom}(4) \) and \( 6 = \text{dom}(7) = \text{dom}(8) = \text{dom}(9) \). Step (b), 1, 2 and 4 send their adjacency lists to 3, and 7, 8 and 9 send their adjacency lists to 6. For finding routes between all pairs of nodes, OSR has the following time and message complexities.

<table>
<thead>
<tr>
<th>Move</th>
<th>Effects</th>
<th>Route updates</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>2 in same domains, (2,4) broken</td>
<td>&lt;2,3,4&gt; new route</td>
</tr>
<tr>
<td>b</td>
<td>2 now in 5’s domain, (2,1) broken</td>
<td>&lt;2,3,5,1&gt;</td>
</tr>
<tr>
<td>c</td>
<td>4 is new spine node, 2 in 4’s domain, (2,1)(2,3) broken</td>
<td>&lt;2,4,3,1&gt; &lt;2,4,3&gt;</td>
</tr>
<tr>
<td>d</td>
<td>2 isolated</td>
<td>all routes to 2 deleted</td>
</tr>
</tbody>
</table>

Figure 19

Non spine moves: a, b, c, d
spine leaf move: e
spine interior node move: f
<table>
<thead>
<tr>
<th>Step</th>
<th>Time</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute spine</td>
<td>$O((n +</td>
<td>C</td>
</tr>
<tr>
<td>Gather info</td>
<td>$O(\Delta')$</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>Spine broadcast</td>
<td>$O((n + \text{diam}(C))\Delta(C))$</td>
<td>$O(n</td>
</tr>
<tr>
<td>Local shortest paths</td>
<td>$O(1)$</td>
<td>none</td>
</tr>
<tr>
<td>Total</td>
<td>$O((n +</td>
<td>C</td>
</tr>
</tbody>
</table>

Let $\text{diam}(C)$ be the diameter of $C$, and $\Delta(C)$ be the maximum degree of $C$.

**Partial – Knowledge Spine Routing (PSR)**

In OSR each spine node needs to maintain global state. Therefore authors propose PSR, a light weight spine routing algorithm that uses only local state for computing “good “routes rather than “optimal “routes. In order to improve the optimality of the routes that PSR compute. Authors introduce the novel mechanism by which information about stable edges gets propagated on the spine. The mechanism uses two types of waves namely add and delete waves to convey the stable edge information. Finally PSR uses a LMR like directed probing mechanism for route discovery.

**2.3.11 Routing in Ad Hoc Network Using Spine**

Wireless ad hoc network has no physical backbone infrastructure a virtual back bone can be formed by nodes in connected dominating set of the corresponding unit – disk graph. Such virtual back bone also referred to as spine. In [26] Das Shivakumar and Bharghavan presented a two –level hierarchical routing architecture for ad hoc networks. Within each cluster at the lower level, we use self- organizing, dynamic
structure called a spine to help reduce the overhead. Between clusters they maintain link – state knowledge of the cluster graph topology in which each cluster is represented by one node.

**Spine Routing**

To determine routes with the spine, authors gather global knowledge of $G$ in to the entire spine nodes and computes shortest path based on local copies of $G$. The algorithm to construct the spine and identity shortest path routes is largely the same as in [25][26]. They used an approximation $C$ to construct a minimum connected dominating set ($MCDS$) as their spine. Authors made one improvement compared to the algorithm in [18] for the route discovery step. The basic spine routing algorithm takes $O((n + |c|) \Delta) = O(n \Delta)$ time to initially determine routes using $O((n + |c| + m + n \log n)$ messages.

**Hierarchical Spine Routing Algorithm**

Authors characterize the clustering of $G$ using several parameters. Each cluster has $\delta c$ and $\Delta c$ nodes minimum and maximum degrees respectively. The roots in the cluster maintain upper-level topology and gives inter cluster routes. Some nodes considered as boundary nodes which are adjacent to other cluster nodes. The spine within each cluster is denoted by $S_c$ and has at most $|S_c|$ nodes.
Route discovery

The spine routing algorithm establishes a spine within each node. For finding a routes in inter cluster the cluster head maintains the topology of $G_c$, membership table and list of local boundary nodes. The following algorithm describes this process.

1) $s$ checks route cache for route to $t$

2) $s$ asks $\text{dom}(s)$ for route:

   If $t \in C_s$ then $\text{dom}(s)$ replies with route

3) ($t \notin C_s$)

   $\text{dom}(s)$ asks root of $C_s$ for route

   a) Root of $C_s$ looks up $C_s$

   b) Root replies to $\text{dom}(s)$ with

      i) Cluster route $C_s = C_1, C_2 \ldots \ldots , C_k = C_t$

      ii) Boundary node $b$ in $C_s$ adjacent to $C_2$

   c) $\text{dom}(s)$ Forwards $C_1, C_2 \ldots C_k$ and $b$ to $s$. 
2.3.12 Minimum CDS with Shortest Path Constraint

In [27] Ling, Gao, Wu Lee, Zhu, and Du studied how to construct a shortest path connected dominating set (SPCDS) in a network which cannot be modeled as a complete graph. Authors introduce constructions of CDS from two aspects centralized construction and distributed constructions. They have categorized centralized CDS algorithm into two types—one is 1-stage and other is 2-stage. In 2-stage algorithms the first step is to select a minimum CDS using the technique of Steiner Tree [28]. In contrast 1-stage algorithm aim to select a CDS directly skipping the step of finding a DS. Authors main objective is to find *Connected Dominating Set* (CDS) in wireless networks, which selects a minimum CDS with property that all intermediate nodes inside every pair wise shortest path should be included. Authors proved that finding such a minimum SPCDS can be achieved in polynomial time and design an exact algorithm with time complexity \(O(\delta^2 n)\), where \(\delta\) is the maximum node degree in communication graph.

![Diagram](image)

**Figure 20**

Node in CDS set

---

112
Let $(u, v) = \{u_1, w_2, \ldots, w_r, v\}$ be one shortest path between $u$ and $v$ in $V$, and all nodes on $p(u, v)$ except $u, v$ are called intermediate nodes. Every node pair may have more than one shortest paths and these shortest paths compose of a shortest path set $(u, v)$. For instance, in Fig. 20, the shortest path between $B$ and $D$ can be $p_1(B, D) = \{B, E, D\}$ or $p_2(B, D) = \{B, A, D\}$. Therefore, the shortest path set between node $B$ and $C$ should be $PB, = \{p_1(B, D), p_2(B, D)\}$.

Figure 21. An Example Network with 10 Nodes.
They proposed Alg. 1 based on 2-hop neighborhood information. After collecting information, nodes will be checked one by one to see whether it has a pair of dis-adjacent neighbors. If a node has only one 1-hop neighbor, then the node will not be selected as one member in $S$. In Fig. 21(b), $I$ has only one neighbor $H$, so it will not be chosen as a node in $S$. In Fig. 21(c), $C$ has two dis-adjacent neighbors $A$ and $B$, so $C$ should be selected into $S$. In Fig. 21(d), $E$ has two neighbor nodes $D$ and $F$ and, $D$ and $F$ are within each other’s transmission range, so $E$ will not be chosen. The detailed algorithm can be shown in Alg. 1.

2.3.13 Iterative Local Solutions for Connected Dominating Sets.

In [29] Wu Dai. and Yang present a general frame work of the iterative local solution (ILS) that relaxes the non propagation constraint of local solutions in order to improve efficiency. Each application of a selected local solution enhances the result obtain from the previous iteration but based on a different node priority scheme. However, ILS still keeps locality that is ILS can quickly provide a solution after a network topology change. Here authors focus on using ILS to calculate a $CDS$ with the objective of reducing the $CDS$ size over a number of iterations $ILS$ in a Static Environment: The following algorithm shows a $K$-round $ILS$. Where local topology information can be defined in different ways algorithm $K$-round $ILS$ (at each node U)
1: Each node collects local topology information and applies a local solution to determine its states (marked or unmarked)

2: The process completes if the number of iterations reaches K; otherwise each node selects a new priority and exchanges states,

3: Apply the local solution again based on the new node states and node priorities go to step 2 for the next iteration.

**SILS in a Dynamic Environment**

After pointing out several drawbacks of CILS. We give a novel extension of the ILS called the stream less iterative local solution (SILS). The basic idea is that the CDS formation process continues beyond \( K \) rounds of iteration Node states (marked/unmarked) is adjusted in reaction to topology changes as the process iterates.

1. **At each round. Rule \( K \) is applied at all nodes. Marked or unmarked previously to determine their new states.**

2. **Node states is no longer exchanged among neighbors.**

The main contributions of the authors are the stream less integration of the iterative process and handling of topology changes in ad hoc wireless networks. They considered two extensions to the ILS to extend its use beyond static environment one is the natural extension that fails to obtain many desirable properties.
Table 2: Performance comparison of the presented CDS algorithms

<table>
<thead>
<tr>
<th>CDS Approaches</th>
<th>Type</th>
<th>Time Complexity</th>
<th>Message Complexity</th>
<th>Performance Ratio</th>
<th>Neighbor Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alzoubi et al.</td>
<td>MIS based</td>
<td>$O(n)$</td>
<td>$O(n \log n)$</td>
<td>Constant</td>
<td>1-hop</td>
</tr>
<tr>
<td>Bhargyan &amp; Das</td>
<td>Single initiator</td>
<td>$O(n^2)$</td>
<td>$O(n^2)$</td>
<td>$\Theta(\log n)$</td>
<td>2-hop</td>
</tr>
<tr>
<td>Das &amp; Bhargyan</td>
<td>Single initiator</td>
<td>$O(n^2)$</td>
<td>$O(n^2)$</td>
<td>$\Theta(\log n)$</td>
<td>2-hop</td>
</tr>
<tr>
<td>Shivakumar et al.</td>
<td>Single initiator</td>
<td>$O(n^2)$</td>
<td>$O(n^2)$</td>
<td>$\Theta(\log n)$</td>
<td>2-hop</td>
</tr>
<tr>
<td>Wu &amp; Li</td>
<td>Prune based</td>
<td>$(O(D^3))$</td>
<td>$\Theta(m)$</td>
<td>$n$</td>
<td>2-hop</td>
</tr>
<tr>
<td>Stojmenovic</td>
<td>Prune based</td>
<td>$\Omega(n)$</td>
<td>$O(n^2)$</td>
<td>$n$</td>
<td>2-hop</td>
</tr>
<tr>
<td>Li et al.</td>
<td>Prune based</td>
<td>$O(\Delta)$</td>
<td>$(O(nD^2))$</td>
<td>$O(4.8 + \ln 5)$</td>
<td>2-hop</td>
</tr>
<tr>
<td>Guha &amp; Kullar</td>
<td>Centralized</td>
<td>$O(n^2)$</td>
<td>$O(n^2)$</td>
<td>$O(n \log n)$</td>
<td>----</td>
</tr>
</tbody>
</table>
2.3.14 Conclusions

Wireless Ad Hoc networks have attracted significant attention over the past few years. A growing list of civil and military applications can employ wireless Ad hoc networks for increased effectiveness. Significant attention has been paid to routing algorithms strategies and algorithms yielding a large number of publications. In this paper, we surveyed the state of the research and classified the different domination set based routing algorithms. We highlighted the effect of the topology on the existing approaches and summarized a number of schemes stating their strength and limitations.
References


[27] Ling Ding · Xiaofeng Gao · Weili Wu · Wonjun Lee · Xu Zhu · Ding-Zhu Du “An Exact Algorithm for Minimum CDS with Shortest Path Constraint in Wireless Networks “.


http://tonnant.itd.nrl.navy.mil/mmnet/ mmnetRFC.txt.”.