Clustering Algorithms for Hierarchical Routing in Wireless Networks

Amira Mohamed Kotb, Associate Lecturer — Prof. Dr. Mohamed Gamal El-Din Darwish, Vice Dean — Dr. Imane Aly Saroit, Associate Professor

Information Technology Dept., Faculty of computers and Information, Cairo University

Abstract

Routing in wireless networks faces many problems like the wired networks, in addition to the problems due to the mobility and the wireless nature of the wireless networks. By introducing hierarchical routing to the wireless networks, these problems can be overcome. Clustering provides a method to build and maintain hierarchical levels in wireless networks. This paper provides an comparative study of the hierarchical routing in wireless networks, and compare between the different proposed clustering algorithms to establish the hierarchical image of the network. It focuses on two main approaches, the 1-hop clustering and the D-hop clustering. The 1-hop clustering will be discussed considering two clustering algorithms, the Lowest ID clustering algorithm and the Connectivity Based clustering algorithm. The D-hop clustering will be illustrated considering two clustering algorithms, the Max-Min D-clustering algorithm and the Connectivity Based D-clustering algorithm. Evaluation of their performance is illustrated.

1. Introduction.

Routing in Ad Hoc networks faces extreme challenges from node mobility/dynamics, potentially very large number of nodes and limited communication resources [1,2]. The scalability is more challenging in the presence of both, the large number of nodes and node mobility. The routing protocols have to adapt quickly frequent and unpredictable topology changes.

Three routing strategies are considered, source routing, distributed routing and hierarchical routing. They are classified according to the way how the state information is maintained and how the search of feasible paths is carried out. In the source routing [3,4]; each node maintains the complete global state, including the network topology and the state information of every link. Based on the global state, a feasible path is locally computed at the source node. In the distributed routing [5]; the path is computed by a distributed computation. Control messages are exchanged among the nodes and the state information kept at each node is collectively used for the path search. In the hierarchical routing [6-8]; nodes are clustered into groups, which are further clustered into higher-level groups recursively, creating a multi-level hierarchy. Each physical node maintains an aggregated global state. Source routing is used to find feasible path on which some nodes are logical nodes representing groups.

Hierarchical techniques have long been known to afford scalability in wireless networks. By summarizing topology details via a hierarchical map of the network topology, network nodes are able to conserve memory and link resources [9,13]. There are several clustering algorithms that construct the hierarchical map of the network topology.

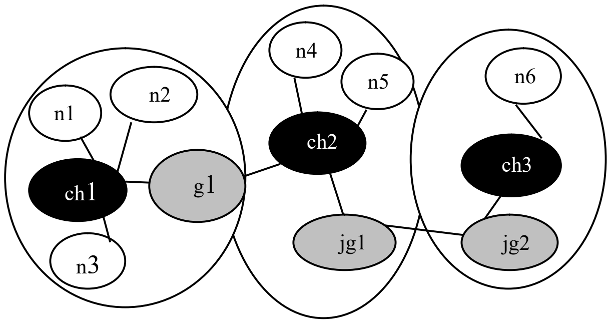
Clustering algorithms are classified based on graph domination [15], as clustering with independent dominating sets [13,14], clustering with dominating sets [16], clustering with connected dominating sets [16-18], clustering with weakly connected dominating sets [19] or clustering with other methods than graph domination [20,21].

This paper analyze the performance of four clustering algorithms considering two clustering approaches, the 1-hop clustering and the D-hop clustering.

The rest of this paper is organized as follows. In section 2: the state information, needed by the different algorithms, will be described. In section 3: the 1-hop clustering will be discussed considering two clustering algorithms, the Lowest ID clustering algorithm and the Connectivity Based clustering algorithm. In section 4: the D-hop clustering will be illustrated considering two clustering algorithms, the Max-Min D-clustering algorithm and the Connectivity Based D-clustering algorithm. In section 5: the performance of the clustering algorithms will be discussed and compared. Finally in section 6: a final conclusion and future works will be given.

2. State Information Description

Nodes are grouped into clusters based on different clustering criteria. Each cluster has a cluster leader called cluster head. The cluster head maintains information about all its cluster members, it is responsible about the routing of the messages coming from other cluster heads direct to one of its cluster members. A node that is only one hop away from more than one cluster head is called a *gateway* and can be used by the cluster heads to route between the clusters [9]. It is also possible for a pair of nodes to serve as *joint gateways* between two clusters. In such pairing, each node is a member of different clusters, and the two nodes are able to communicate directly as illustrated in figure1.



g: Gateway, jg: Joint Gateways, ch:Cluster Head

Figure 1. Cluster structure

A node maintains either one or two clustering tables, depending on its status level [10]. Nodes that are cluster heads maintain two kind of tables, *a node table* and *a neighbor table.*

The node table contains all nodes that are members of that cluster. It has following two fields:

- The node ID field, which lists the members of the cluster.
- The cluster heads field that lists all the cluster heads for which the node is a gateway. If the node is not a gateway, the cluster heads field is empty list.

The neighbor table stores all neighboring cluster heads and the gateways used to reach them. It has two fields:

- The cluster head ID field that stores the neighboring cluster heads.
- The gateway ID field that lists the gateway for the cluster head indicated in the cluster head ID field.

Table1 illustrates the cluster head tables for the clustered network in figure1.

Member Node ID	Cluster Head
n1	None
n2	None
n3	None
g1	ch2

a: Node table of Ch1

Cluster Head ID	Gateway ID
ch2	g1
ch3	None

b: Neighbor table of Ch1

Table 1. Cluster head tables

Nodes that are not cluster heads, maintain only one table, *the cluster head table.* This table has the following fields:

- Cluster head ID field, where there is an entry for each cluster head.
- Joint gateway ID field, which contains the neighboring nodes that are members of cluster head's cluster, and that can be used to reach the neighboring cluster head.
- Gateway ID field, a flag that will be set if the node is a gateway between different clusters.

Table 2 illustrates the cluster head table for member node Jg1 in figure 1. Different clustering algorithms with different clustering criteria can use this data structure. The next two sections illustrate different clustering algorithms considering the 1-hop clustering and the D-hop clustering.

Cluster head	Joint Gateway	Gateway
ch2	None	None
ch3	jg2	None

Table 2.Cluster head table for member node Jg1

3. The 1-Hop Clustering

In a single hierarchy, nodes are divided into clusters, each cluster has a cluster head. A multi-level hierarchy has nodes organized in a tree-like fashion with several levels of cluster heads [11]. A three level hierarchy employs ordinary nodes, cluster heads and super cluster heads, It is suitable for networks with few hundreds nodes. For simplicity, single level hierarchies is considered.

The 1-hop clustering approach puts constraints on the size of the cluster. The cluster head must be only 1-hop away from any member node in its cluster [10,11]. So there is no node that can be more than 1-hop away from its own cluster head. This limits the cluster size and the members in the same cluster are just two hops away from each other.

In this section, two clustering algorithms considering this approach will be illustrated, the Lowest ID algorithm and the Connectivity Based algorithm [10].

3.1 The Lowest ID Clustering Algorithm

In this subsection the Lowest ID clustering algorithm will be illustrated, considering the 1-hop clustering approach.

3.1.1 The Basic Idea

In the Lowest ID algorithm, a node checks the ID of all its one hop neighbors(that are not cluster heads or members). If this node has the lowest ID among all its neighbors, it declares itself as a cluster head (CH) and sends messages to all its one hop neighbors to declare themselves as cluster members. For example, in figure 2, node 3 is the lowest ID node among all its neighbors (node 4,node 11), so node 3 elects itself to be cluster head and nodes 4 and 11 specify themselves as cluster members. Node 2 is not considered because it is already member of another cluster. A node, which can hear two or more CHs, is a 'gateway'. Otherwise, a node is an ordinary node [10].

The Lowest ID algorithm has the properties that cluster heads are not directly linked, also each cluster head is directly linked to every other node in its cluster. Thus each node is either cluster head itself or is directly linked to one or more cluster heads. Such clustering is refereed to as 1-hop clustering.

We observe that in some cases the algorithm may fail to cluster the nodes. Suppose that the neighbors of the lowest ID node are classified as members in other clusters, this leads to a cluster that has a cluster head (lowest ID node) but has *no* members (empty cluster). This can be solved by considering the degree of connectivity of the nodes as will be discussed in the next subsection.

Every node can determine its cluster and only one cluster during the algorithm. For example, the clustering algorithm for the topology in figure 2 produces clusters as indicated in figure 3. The algorithm creates non-overlapping clusters by requesting nodes to select one of several neighboring CHs. Note that the node may still be linked to other CHs and thus the clustering organization is essentially overlapping. The maintenance of clusters is performed in the following way: within each cluster, nodes must be able to communicate with each other in at most two hops. Incoming nodes that preserve the property may join the cluster. When a link is disconnected, the highest connectivity node and its neighbors stay in the original cluster. Thus this node effectively takes over the cluster head role from the lowest ID node. Other nodes from the former cluster shall either join another cluster or form their own cluster. This may lead to single node clusters. Thus it seems that additional procedures for merging or rearranging clusters may be desirable.

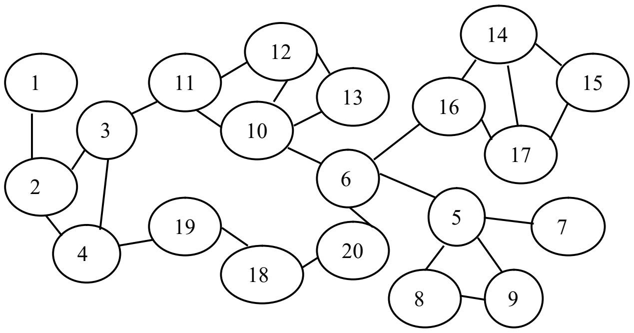


Figure 2. System Topology

The Lowest ID clustering algorithm creates a one level cluster topology across a network of nodes. Nodes are grouped into clusters based on their proximity to one another. Each cluster has a cluster leader called cluster head. Nodes within a cluster are one-hop neighbors of the cluster head. The following rules explain the logical steps of the algorithm to cluster the network.

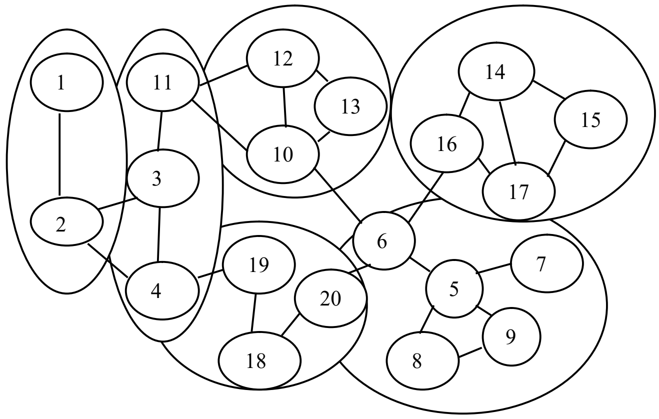


Figure 3. Lowest ID Clustering

Rule 1: every node check if it is the Lowest ID node among its 1-hop neighbors, then it declare itself a cluster head, all its 1-hop neighbors members and skip the rest of the rules, otherwise proceed to Rule 2.

Rule 2: every node check if its lowest ID neighbor is cluster head, then it declare itself as member. If the lowest ID neighbor is NOT a cluster head, then proceed to Rule 3.

Rule 3: eliminate the lowest ID neighbor from the nodes array, then proceed to Rule 1 again.

3.2 The Connectivity Based Clustering Algorithm

In this subsection the Connectivity Based clustering algorithm will be illustrated, considering the 1-hop clustering approach.

3.2.1 The Basic Idea

The Connectivity Based clustering algorithm does not consider the node ID in its cluster formation process. Otherwise, the highest degree node in a neighborhood becomes the cluster head. The node with large number of links to other nodes is expected to be cluster head [10].

The Connectivity Based algorithm has the following properties; first, Cluster heads are not directly linked, second, Each cluster head is directly linked to every other node in its cluster with the highest connectivity. Thus each node is either cluster head itself or is directly linked to one or more cluster heads. Such clustering is refereed to as 1-hop clustering.

Every node can determine its cluster and only one cluster during the algorithm. For example, the clustering algorithm for the topology in figure 2 produces clusters as indicated in figure 4. Like the Lowest ID algorithm, the algorithm creates non-overlapping clusters by requesting nodes to select one of several neighboring cluster heads. Note that the node may still be linked to other cluster heads, and thus the clustering organization is essentially overlapping.

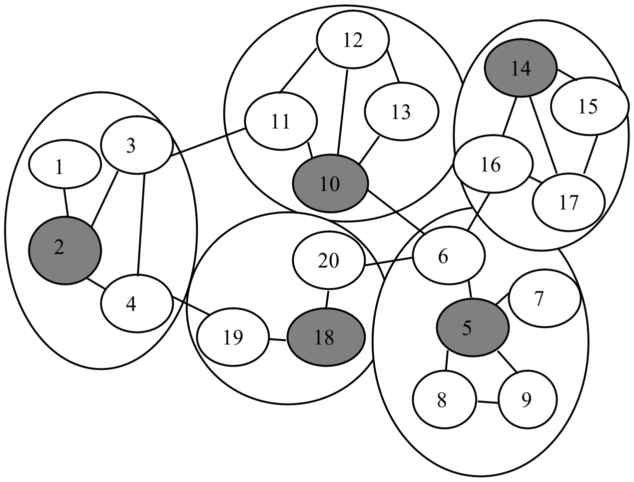


Figure 4. Connectivity Based Clustering

The connectivity based clustering algorithm creates a one level cluster topology across a network of nodes. Nodes are grouped into clusters based on their connectivity degree. The following rules explain the logical steps of the algorithm to create clusters.

Rule 1: every node check if it is the highest degree node, then it declare itself as cluster head and all its 1-hop neighbors are members of its cluster. if node is neither cluster head nor member, proceed to Rule 2.

Rule 2: eliminate nodes that are declared as both cluster heads and members. For uncovered nodes, proceed to Rule 1 again.

4. The D-Hop Clustering

The 1-hop clustering approach, as mentioned in the previous section, puts constraints on the size of the cluster. The cluster head must be only 1-hop away from any member node in its cluster. However, in the D-hop clustering, the cluster head can be D-hop away from its member or less than D-hop. "D" is a system parameter that depends on the network size [12,22]. The D-hop clustering loosens the constraints on the size of the cluster, leading to lager cluster size and less number of clusters than that of 1-hop clustering. The performance analysis of the clustering algorithms will be discussed in the next section.

In this section, two clustering algorithms will be illustrated considering this approach. The Max Min D-clustering algorithm and the Connectivity Based D-clustering algorithm.

4.1 The Max Min D-Clustering Algorithm

The MaxMin D-clustering algorithm generalize the cluster definition to a collection of nodes that are up to *d*-hops away from the cluster head.

4.1.1 Basic Terminology

The algorithm runs for 2d rounds of information exchange; each node maintains two arrays, WINNER and SENDER, each of size 2d node ids: one id per round of information exchange [12].

- The WINNER is the winning node id of a particular round. It is used to determine the cluster head for a node, as described below in the basic idea.
- The SENDER is the node that sent the winning node id for a particular round. It is used to determine the shortest path back to the cluster head, once the cluster head is selected.

Initially, each node sets its WINNER to be equal to its own node id. This is followed by the Floodmax phase.

The Floodmax: Each node locally broadcasts its WINNER value to all of its 1-hop neighbors. After all neighboring nodes have been heard from, for a single round, the node chooses the largest value among its own WINNER value and the values received in the round as its new WINNER. This process continues for d rounds.

The Floodmin: This follows Floodmax and also lasts d rounds. It is the same as Floodmax except that a node chooses the smallest rather than the largest value as its new WINNER.

Overtake: As flooding occurs in the network, WINNER values are propagated to neighboring nodes. At the end of each flooding round a node decides to maintain its current WINNER value or change to a value that was received in the previous flood round. Overtaking is the act of a new value, different from the node's own id, being selected based on the outcome of the information exchange.

Node Pairs: A node pair is any node id that occurs at least once as a WINNER in both the (Floodmax) and (Floodmin) d rounds of flooding for an individual node.

The algorithm has three logical stages; first the propagation of larger node ids via floodmax, second the propagation of smaller node ids via floodmin and third the determination of cluster heads.

The first stage uses 'd' rounds of floodmax to propagate the largest node id in each node's neighborhood. At the conclusion of the floodmax the surviving node ids are the elected cluster heads in the network. Nodes record their winning node for each round. Floodmax is a greedy algorithm and may result in an unbalanced loading for the cluster heads [12]. In fact, there may be cases where the cluster head is disjoint from its cluster as a result of being overtaken by another cluster head. Therefore, a node must realize not only if it is the largest in its d-neighborhood but also if it is the largest in any other node's d-neighborhood.

The second stage uses 'd' rounds of floodmin to propagate the smaller node ids that have not been overtaken. This allows the relatively smaller cluster heads the opportunity to (i) allow them to regain nodes within their d-neighborhood and, (ii) realize that they are the largest node in another node's d-neighborhood. Again each node records the winning node for each round.

At the conclusion of the floodmin, each node evaluates the round's WINNERs to best determine their cluster head. In order to accommodate cases where a node's id is overtaken by another node id, the smallest node id appearing in both of the flooding stages is chosen as the cluster head. The smaller cluster head is chosen to provide load balancing.

4.1.2 The Basic Idea

The mechanics of the algorithm are quite simple. At some common epoch each node initiates 2d rounds of flooding. Each node maintains a logged entry of the results of each flooding round. The rounds are segmented into the first d rounds and the second d rounds. The first d rounds are a floodmax to propagate the largest node ids. After completion of the first d rounds of flooding, the second d rounds of flooding begin. It use the values that exist at each node after the first d rounds. The second d rounds of flooding are a floodmin to allow the smaller node ids to reclaim some of their territory. After completion of the second d rounds each node looks at its logged entries for the 2d rounds of flooding. The following rules explain the logical steps of the algorithm that each node runs on the logged entries to creates a clustered network.

Rule 1: First, each node checks to see if it has received its own original node id in the second d rounds of flooding. If it has then it can declare itself a cluster head and skip the rest of this phase of the heuristic. Otherwise proceed to Rule 2.

Rule 2: Each node looks for node pairs. Once a node has identified all node pairs, it selects the minimum node pair to be the cluster head. If a node pair does not exist for a node then proceed to Rule 3.

Rule 3: Elect the maximum node id in the first d rounds of flooding as the cluster head for this node.

After a node has determined its cluster head based on Rules 1, 2, or 3, it communicates that it is a member of the cluster to the cluster head. In order to minimize messages this information is communicated from the fringes of the cluster, gateway nodes, inward to the cluster head. Furthermore, a node has no way to know if it is a gateway node. Therefore, after cluster head selection each node broadcasts its elected cluster head to all of its neighbors. Only after hearing from all neighbors can a node determine if it is a gateway node. If all neighbors of a node have the same cluster head selection then this node is not a gateway node. However, if there are neighboring nodes with cluster head selection that are different, then these nodes are gateway nodes.

The Max-Min D-clustering algorithm will be used to cluster the network in figure 5 leading to the clustered network shown in figure 6.

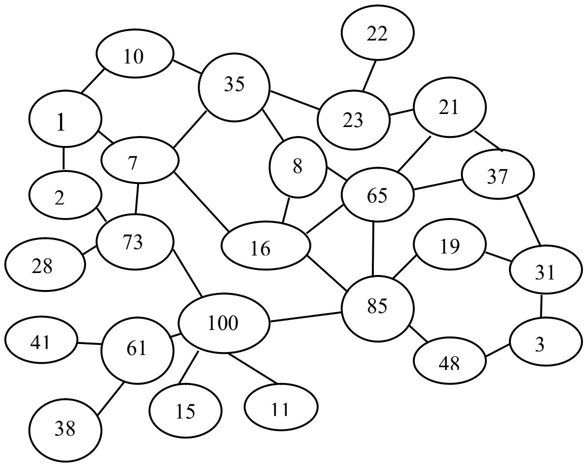


Figure 5. System Topology

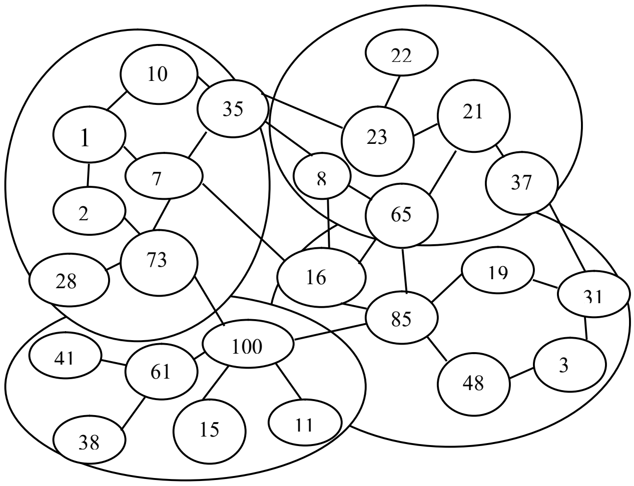


Figure 6. MaxMin D-clustering with d=3

4.2 Connectivity Based D-Clustering Algorithm

In this subsection the Connectivity Based D-clustering algorithm will be illustrated, considering the D-hop clustering approach.

4.2.1 Basic Idea

The Connectivity Based clustering algorithm, previously illustrated in section 3.2, does not consider the node ID in its cluster formation process. Otherwise, the highest degree node in a neighborhood becomes the cluster head. More precisely, such nodes are elected as cluster heads and their neighbors are then covered [11]. In the Connectivity Based D-Clustering algorithm, the same concept will be used with slight difference. Such that the cluster head is the node with the highest connectivity degree, but the cluster head can be 1-hop or d-hops away from any of its members. Figure 7 shows the clusters generated by the Connectivity based D-clustering algorithm for the network topology used in figure 5.

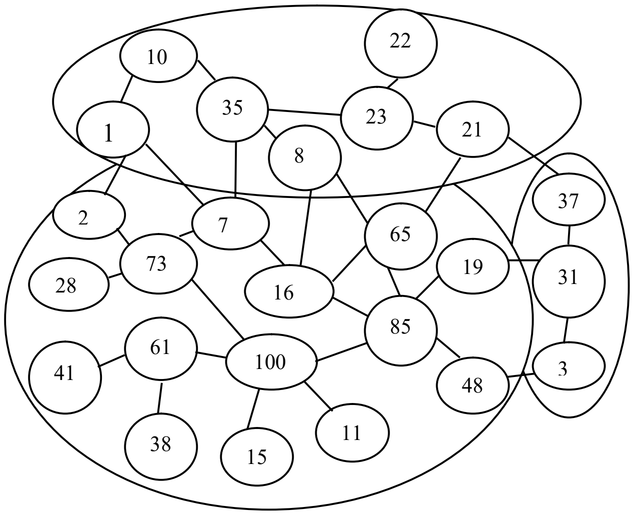


Figure 7. Connectivity Based D-clustering with d=2

The connectivity based D-clustering algorithm creates a clustered network that depends on the connectivity degree. The following rules explain the logical steps of the algorithm to create clusters.

Rule 1: every node check if it is the highest degree node among its "*d*" neighbors, then it declare itself as cluster head and all its *d*-hop neighbors are members of its cluster. if node is neither cluster head nor member, proceed to Rule 2.

Rule 2: eliminate nodes that are declared as both cluster heads and members. For uncovered nodes, proceed to Rule 1 again.

5. Analysis of Clustering Algorithms

Performance evaluation for the clustering algorithms illustrated in section 4; the Lowest ID algorithm, the Connectivity Based algorithm, the Max-Min D-clustering algorithm and the Connectivity Based D-clustering algorithm, is done in this section.

5.1 Performance Parameters

The following five performance parameters will be considered in order to evaluate the performance of the clustering algorithms.

1. Average Number of Created Clusters

The average number of created clusters can refer also to the average number of cluster heads. In both cases, it is the mean number of clusters in a network for a sample.

$$\text{Average number of clusters} = \frac{\text{Number of cluster heads}}{\text{Total number of nodes}}$$

2. Average Cluster Size:

The average cluster size considers the mean size of the cluster. This value is inversely proportional to the average number of created clusters.

$$\text{Average cluster size} = \frac{\text{Total number of member nodes}}{\text{Total number of clusters}}$$

3. Average Ratio of Gateways

The average ratio of gateways represents the number of gateways and how they are affected by the size of the network.

$$\text{Average ratio of Gateways} = \frac{\text{Total number of gateways}}{\text{Total number of nodes}}$$

4. Average Ratio of Joint Gateways

Like the average ratio of gateways, the average ratio of joint gateways represents the number of joint gateways and how they are affected by the size of the network.

$$\text{Average ratio of Jgateways} = \frac{\text{Total number of Jgateway}}{\text{Total number of nodes}}$$

5. Average Message Overhead:

The average message overhead considers the messaging and the load on the network during the clustering process.

$$\text{Average message overhead} = \frac{\text{Total number of messages}}{\text{Total number of nodes}}$$

5.2 Results

In order to obtain a clustered network that can make routing more efficient and simple, a simulation program was built using C++ programming language using its tools and libraries. The network topology used in the simulation is randomly generated ranging from 20 to 50 nodes placed randomly. This randomization makes the network topology periodically change to resemble the mobile nature of the wireless networks.

In this subsection, the performance issues will be calculated for the clustering algorithms considering both the 1-hop clustering (Lowest ID, ConBased algorithms) and the D-hop clustering (MaxMin-D clustering, ConBased-D clustering algorithms) with d=2.

1. Average Number of Created Clusters

Figure 8 shows the average number of created clusters for the four algorithms. It is obvious from the figure that the 1-hop clustering algorithms create more clusters than the d-hop clustering algorithms, because the d-hop clustering algorithms gather the nodes that are not only 1-hop distance from the cluster head but those who are d-hop distance from the cluster head, leading to small number of clusters.

For the 1-hop clustering, the lowest ID algorithm has the highest average number of created clusters as it does not consider the connectivity of the nodes. For the D-hop clustering, the ConBased-D algorithm has the lowest average number of created clusters as it creates small number of highly interconnected clusters.

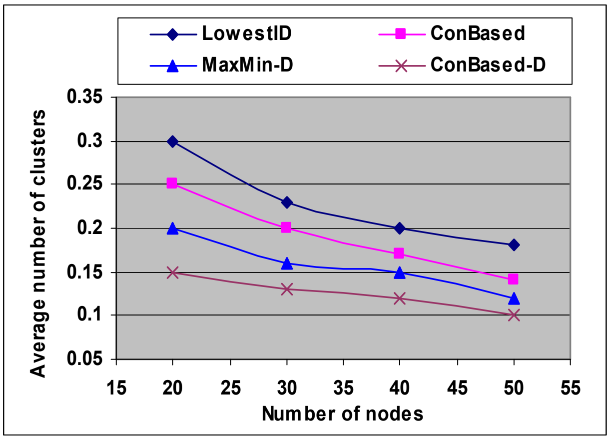


Figure 8. Average Number of Created Clusters

It is neither recommended to have small number of created clusters, as they will be overloaded with too many cluster members [11], nor it is good to have large number of created clusters, each managing very small number of members. For the MaxMin-D clustering algorithm, it provides a moderate average number of created clusters. The average number of created clusters for the MaxMin-D clustering algorithm is higher than that of the ConBased-D clustering algorithm because it considers the node ID as its clustering criteria. Also it has small average number of created clusters compared to the 1-hop clustering algorithms because the nodes are D-hops away from their cluster heads.

2. Average Cluster Size

Figure 9 illustrates the average cluster size for the four algorithms. The average cluster size is inversely proportional to the average number of created clusters. So it is clear from the figure that the D-hop clustering algorithms have larger average cluster size than that of the 1-hop clustering algorithms, due to the small average number of created clusters of the d-hop clustering algorithms.

For the 1-hop clustering, the LowestID algorithm has the smallest average cluster size because it creates the largest number of clusters compared to all other algorithms. For the D-hop clustering, the ConBased-D clustering algorithm has the largest average cluster size in which its clusters contain many members because there are few created clusters

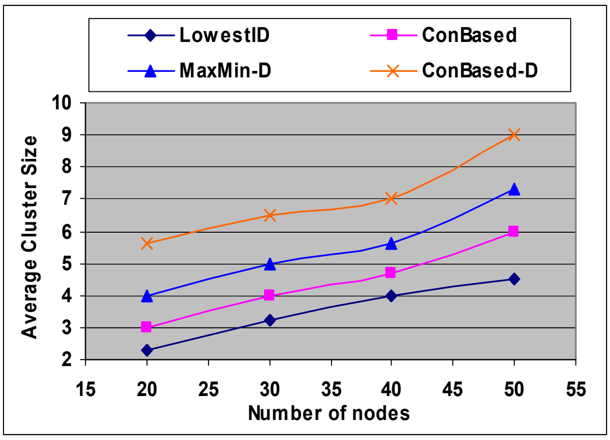


Figure 9. Average Cluster Size

As for the average number of created clusters, it is also recommended that the size of the cluster to be moderated. So the cluster must not contains many members resulting on loading the cluster head, nor small number of members [11]. The MaxMin-D clustering algorithm has a small average cluster size with respect to the D-hop clustering, as it has higher number of created clusters than that of the ConBased-D clustering algorithm. It has a moderate average cluster size compared to both the D-hop clustering and 1-hop clustering, tending to be close to that of the 1-hop clustering algorithms.

3. Average Ratio of Gateways

Figure 10 illustrates the average ratio of gateways for the four algorithms. It is obvious from the figure that the D-clustering algorithms have less average ratio of gateways than that of the 1-hop clustering algorithms. The member nodes in the D-hop clustering are not only one hop away from the cluster head but it may be D-hop away from its cluster head. This reduces the number of nodes that can be directly connected to two or more cluster heads (gateway).

For the 1-hop clustering, the lowest ID algorithm has the highest average ratio of gateways, due to its large number of created clusters. For the D-hop clustering, the ConBased-D clustering algorithm has the lowest average ratio of gateways because of its small number of created clusters.

The more number of gateways, the more flexibility in routing between clusters due to existence of multiple paths between the neighboring cluster heads [11]. The MaxMin-D clustering algorithm has the highest average ratio of gateways for the D-clustering (compared to the ConBased-D clustering algorithm), because it has higher average number of created clusters.

It also has a small average ratio of gateways compared to the ConBased algorithm, due to the extension of the cluster scope (members are D-hop away from the cluster head). The average ratio of gateways for the MaxMin-D clustering algorithm is much closer to that of the 1-hop clustering algorithms

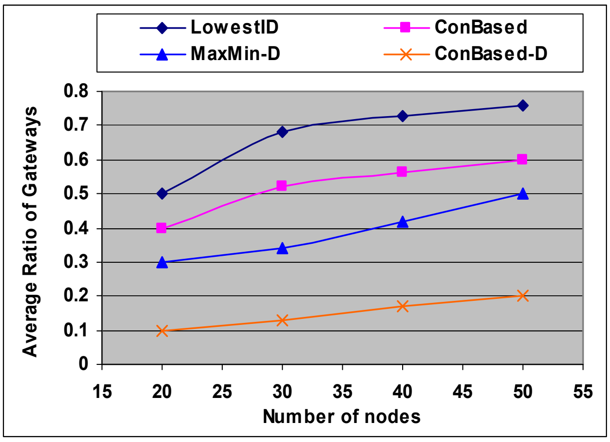


Figure 10. Average Ratio of Gateways

4. Average Ratio of Joint Gateways

Figure 11 shows the average ratio of joint gateways for the four algorithms. Like the average ratio of gateways, figure 11 shows that the 1-hop clustering algorithms achieve higher average ratio of joint gateways than that of the D-hop clustering algorithms.

For the 1-hop clustering, the lowest ID algorithm achieves the largest average ratio of joint gateways, because it does not consider the connectivity degree in its clustering criteria. For the D-hop clustering, the ConBased-D clustering algorithm has the lowest average ratio of joint gateways. It has small number of created clusters whose members are highly interconnected to each other leading to very small number of joint gateways.

The MaxMin-D clustering algorithm achieves the highest average ratio of joint gateways for the D-clustering (compared to the ConBased-D clustering algorithm), because it does not consider the connectivity of the nodes, but it primary depends on the node ID as clustering criteria. It also has a moderate average ratio of joint gateways between the ConBased algorithm and the ConBased-D clustering algorithm. Like the average ratio of gateways, the average ratio of joint gateways for the MaxMin-D clustering algorithm is much closer to that of the 1-hop clustering algorithms.

5. Average Message Overhead

Figure 12 shows the average Message Overhead for the four algorithms. It is obvious from the figure that the average message overhead for the D-clustering algorithms is much higher than that of the 1-hop clustering algorithms. More messages will be created in order to cover the D-hop distance between the cluster head and its members for the D-clustering algorithms, on the other hand, the 1-hop clustering algorithms create messages that move only 1-hop between the cluster head and members.

For the 1-hop clustering, the average message overhead of the ConBased algorithm is much higher than that of the Lowest ID algorithm because the ConBased algorithm has higher average cluster size than that of the Lowest ID algorithm. So the ConBased algorithm creates more messages that moves between the large number of members and their cluster head. For the D-hop clustering, the ConBased-D clustering algorithm suffers from the highest average message overhead because it has the highest average cluster size. The large number of nodes in the cluster leads to large number of messaging between the members and their cluster heads.

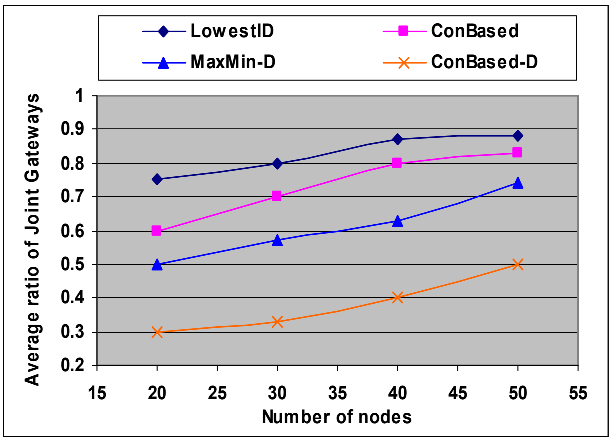


Figure 11. Average Ratio of Joint Gateways

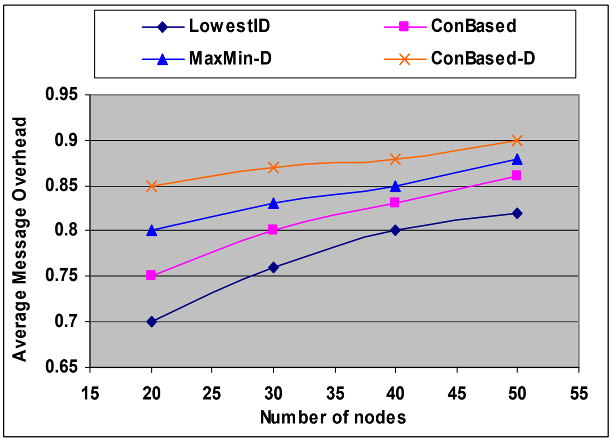


Figure 12. Average Message Overhead

It is obvious from figure 12 that the MaxMin D-clustering algorithm achieves the lowest average message overhead within the D-clustering category. Its small average cluster size leads to small number of messages that moves through the cluster. Compared to the 1-hop clustering algorithms, the MaxMin D-clustering algorithm has average message overhead that is much closer to that of the ConBased algorithm. So the MaxMin D-clustering algorithm achieves a small average message overhead, much less than that of the ConBased D-clustering algorithm and much close to that of the 1-hop clustering algorithms.

5.3 General Comparison

Table 3 summarizes the comparison between the four studied clustering algorithms, LowestID, ConBased, MaxMin-D clustering and ConBased-D clustering algorithms using the five performance parameters mentioned in section 5.1.

1-Hop Clustering / Performance Parameters	Lowest ID	ConBased
Average Number of Created Clusters	Highest	High
Average Cluster Size	Lowest	Low
Average Ratio of Gateways	Highest	High
Average Ratio of Joint Gateways	Highest	High
Average Message Overhead	Low	Moderate

a. 1-Hop Clustering

D-Hop Clustering / Performance Parameters	MaxMin-D	ConBased-D
Average Number of Created Clusters	Highest	High
Average Cluster Size	Lowest	Low
Average Ratio of Gateways	Highest	High
Average Ratio of Joint Gateways	Highest	High
Average Message Overhead	Low	Moderate

b. D-Hop Clustering

Table 3. Summary of Clustering Algorithms

For the 1-hop clustering, two clustering algorithms were considered, the lowest ID clustering algorithm and the connectivity based clustering algorithm. Table 3.a shows that the connectivity based clustering algorithm has a less average number of created clusters than that of the lowest ID clustering algorithm which suffers from the highest average number of created clusters. Also it is clear from the table that the connectivity based algorithm has smaller average cluster size than that of lowest ID algorithm that has the lowest average cluster size. For both, the average ratio of gateways and average ratio of joint gateways, the connectivity based algorithm has less ratios than that of the lowest ID algorithm. Finally, the lowest ID algorithm succeed to has lowest average message overhead as shown in table 5.

For the D-hop clustering two clustering algorithms were considered, the Max-Min D-clustering algorithm and the connectivity based D-clustering algorithm. It is clear from table 3.b that Max-Min D-clustering algorithm has higher average number of created clusters than that of the connectivity based D-clustering algorithm. The Max-Min D-clustering algorithm has smaller average cluster size than that of the connectivity based D-clustering algorithm. The table shows that the Max-Min D-clustering achieves higher average ratio of gateways than that of the connectivity based D-clustering. Also for the average ratio of joint gateways, it is clear from the table that the Max-Min D-clustering achieves higher average ratio of joint gateways than that of the connectivity based D-clustering. Finally, the Max-Min D-clustering achieves small average message overhead compared to that of the connectivity based D-clustering.

Since the network size is dramatically increasing and changeable, the D-hop clustering provides more realistic solution for the clustering problem. It is recommended to work towards the D-hop clustering where *D* is variable with respect to the network size.

It is clear from the table that the MaxMin D-clustering algorithm has better performance over the Lowest ID, Connectivity Based and Connectivity Based D-clustering algorithms. It achieves moderate average number of created clusters, less than that of the 1-hop clustering algorithms and much higher than that of Connectivity Based D-clustering algorithm. Also it has moderate average cluster size, higher than that of the 1-hop clustering algorithms and lower than that of the Connectivity Based D-clustering algorithm. For both, the average ratio of gateways and the average ratio of joint gateways, the MaxMin D-clustering algorithm has high values which is slightly less than that of the 1-hop clustering algorithms and much higher than that of the Connectivity Based D-clustering algorithm. Finally, it achieves moderate average message overhead that is much close to that of the 1-hop clustering (Connectivity Based) and much lower than that of the Connectivity Based D-clustering algorithm.

As a final comparison between the four clustering algorithms previously mentioned, it is clear that the MaxMin D-clustering algorithm has proved to have the best performance compared to the other algorithms.

6 Conclusion and Future Work

In this paper, two clustering approaches were illustrated, the 1-hop clustering and the D-hop clustering. For the 1-hop clustering, two clustering algorithms were discussed, the Lowest ID clustering algorithm and the Connectivity Based clustering algorithm. Also for the D-hop clustering, two clustering algorithms were studied, the MaxMin D-clustering algorithm and Connectivity Based D-clustering algorithm. five performance parameters were considered, the average number of created clusters, average cluster size, average ratio of gateways, average ratio of joint gateways and average message overhead. The MaxMin D-clustering algorithm is considered to have the best performance over the other algorithms. It has a moderate average number of created clusters, a moderate average cluster size, a high average number of gateways, a high average number of joint gateways and a moderate average message overhead.

Some of the future work related to the same subject of this paper can be considered as follows:

1. The selection of the cluster head can depend on other criteria rather than the connectivity degree or the node ID. It can depend on quality of service parameters such as delay, bandwidth, or cost.
2. Clustering can be done without having cluster head, so the cluster can be controlled by the border nodes, these nodes are nodes that have direct links with other nodes in different clusters.
3. The MaxMin-D clustering algorithm can be modified in order to consider the quality of service parameters.

References.

[1] X. Hong, K.Xu, M. Gerla, "Scalable Routing Protocols for Mobile Ad Hoc Networks", IEEE Network, vol.16, no.4, July/August 2002.

[2] Sang Ho Bae, Sung-Ju Lee, " The Design, Implementation, and Performance Evaluation of the On-Demand Multicast Routing Protocol in Multihop Wireless Networks", IEEE Network , vol.14, no.1, February2000

[3] X. Yuan, X. Liu, "Heuristic Algorithms for Multi-Constrained QoS Routing" , IEEE INFOCOM 2001, March 2001.

[4] Y. Chun Hu, D. Johnson, "Implicit Source Routes for On-demand Ad-Hoc Network Routing.", Proceedings of the Second Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc) 2001,

[5] Y. Yang, R. Kravets, " Distributed QoS Guarantees for Real time Traffic in Ad-Hoc Networks", The first IEEE International Conference on Sensor and Ad Hoc communications and Networks, SECON2004

[6] X. Hong, K. Xu, M. Gerla, " Scalable Routing Protocols for Mobile Ad-Hoc Networks", 2002 http://citeseer.nj.nec.com

[7] Y. Ge, L. Lamont, "Hierarchical OLSR: A Scalable Proactive Routing Protocol for Heterogeneous Ad-Hoc Networks", 2005, http://www.crc.ca

[8] S. Banerjee, S. Khuller, "A Cluster Scheme for Hierarchical Control in Wireless Networks.", IEEE INFOCOM'2001, March 2001.

[9] John Sucec, Ivan Marsic, "Hierarchical Routing Overhead in Mobile Ad Hoc Networks", IEEE transactions on mobile computing, vol. 3, no. 1, January-March 2004.

[10] Elizabeth Michelle Royer, "Routing in Ad Hoc Mobile Networks: On Demand and Hierarchical Strategies", PhD thesis in Electrical and Computer Engineering, University of California, Santa Barbara, December 2000. http:// www. Citeseer.nj.nec.com

[11] Ching-Chuan Chiang, Winston Liu, M. Gerla, "Routing in Clustered Mulithop Mobile Wireless Networks with Fading Channels", 2005 http//www.eecs.wsu.edu

[12] Alan D. Amis, Ravi Prakash, "Max-Min D-Cluster Formation in Wireless Ad Hoc Networks", 2000, http:// www. citeseer.nj.nec.com

[13] Y.P. Chen, A.L. Liestman, "Clustering algorithms for Ad-Hoc wireless networks", 2003 http:// www.citeseer.nj.nec.com.

[14] Thomas Moscibroda, Roger Wattenhofer, "Efficient Computation of Maximal Independent sets in Unstructured Multihop Radio Networks", IEEE International Conference on Mobile Ad-Hoc and Sensor Systems MASS 2004

[15] B. Liang and Z. J. Haas, Virtual backbone generation and maintenance in ad hoc network mobility management, in INFOCOM, 2000.

[16] I. Stojmenovic, M. Seddigh, and J. Zunic, Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks, IEEE Transactions on Parallel and Distributed Systems, 2002.

[17] Fei Dai, Jie Wu, "An Extended Localized Algorithm for Connected Dominating set Formation in Ad-Hoc Networks", IEEE Transactions on Parallel and Distributed Systems, vol.15, no.10, October 2004

[18] A dominating-set-based routing scheme in ad hoc wireless networks, Special Issue on Wireless Networks, Telecommunication Systems Journal, 2001.

[19] Y. P. Chen and A. L. Liestman, Approximating minimum size weakly-connected dominating sets for clustering mobile ad hoc networks, in The Third ACM International Symposium on Mobile Ad Hoc Networking and Computing MobiHoc'02), June 2002.

[20] Y.P. Chen, A.L. Liestman, "A zonal algorithm for clustering Ad-Hoc Networks", International Journal of Foundation of Computer Science IJFCS, 2003

[21] S.Banerjee, S.Khuller, " A Clustering scheme for hierarchical routing in wireless networks", Technical report, university of Maryland, college park, February 2000, http:// www.citeseer.nj.nec.com.

[22] C.C.Shen, R.Liu, "CLTC: A Cluster-Based Topology Control Framework for Ad Hoc Networks", IEEE Transaction on Mobile Computing, vol.3, no.1, January 2004.